

Dept. of Computer Science
University of Maryland
A.V. Williams Building
College Park, MD 20742

www.cs.umd.edu/~dvanhorn
dvanhorn@cs.umd.edu
(202) 460-4104

David Van Horn

Education

- 2009 **Brandeis University**
Ph.D. in Computer Science, *The Complexity of Flow Analysis in Higher-Order Languages*
- 2006 **University of Vermont**
M.S. in Computer Science, *Algorithmic Trace Effect Analysis*
- 2003 B.S. in Computer Science & Information Systems

Employment

- 2013 — **University of Maryland**, Assistant Professor
- 2012 — 2013 **Northeastern University**
Research Assistant Professor
- 2011 — 2012 Visiting Assistant Professor
- 2009 — 2011 CRA Computing Innovation Fellow
- 2007 — 2009 Visiting Lecturer

Research interests

I work on all aspects of program analysis and its applications to programming languages, software engineering, verification, and security.

Grants

- 2015–2017 **The Science and Applications of Crypto-Currency**, Co-PI. Awarded by the National Science Foundation Secure & Trustworthy Cyberspace program. Budget: \$1,935,000.
- 2015–2018 **Sound Over- & Under-Approximations of Complexity & Information Security**, Co-PI. Awarded by the DARPA Information Innovation Office (I20), Space/Time Analysis for Cybersecurity (STAC). Program Manager: Timothy Fraser. Budget: \$3,408,353.
- 2014–2017 **Trustworthy and Composable Software Systems with Contracts**. PI. Awarded by the National Security Agency, Science of Security. Budget: \$358,179.
- 2012–2015 **Behavioral Software Contract Verification**. Co-PI. Awarded by National Science Foundation, CISE, Software and Hardware Foundations. Program Director: John Reppy. NSF grant 1218390. Budget: \$400,000.

Grants (continued)

- 2012–2015 **Scalable and Precise Abstractions of Programs for Trustworthy Software.** PI. Awarded by DARPA Information Innovation Office (I2O), Automated Program Analysis for Cybersecurity (APAC). Program Manager: Timothy Fraser. Budget: \$577,000.
- 2009–2011 **Computing Innovation Fellow.** Awarded by the Computing Research Association and Computing Community Consortium. Funded: \$267,500. Subaward of NSF grant CNS-0937060.

Projects

- 2015— **The Science and Applications of Crypto-Currency,** Co-PI.
Crypto-currencies and smart contracts are a new wave of disruptive technology that will shape the future of money and financial transactions. Today, crypto-currencies are a billion-dollar market, and hundreds of companies are entering this space, promising exciting new markets and eco-systems. Unfortunately, usage of crypto-currencies outstrips our understanding. Currently most crypto currencies rely on heuristic designs without a solid appreciation of the necessary security properties, or any formal basis upon which strong assurance of such properties might be achieved.
This work aims to establish a rigorous scientific foundation for crypto-currencies. To achieve this, this work blends cryptography, game theory, programming languages, and systems security techniques. Expected outcomes include new crypto-currency designs with provable security properties, financially enforceable cryptographic protocols whose security properties are backed by enforceable payments in case of a breach, smart contract systems that are easy to program and formally verifiable, as well as high-assurance systems for storing and handling high-value crypto-currencies and transactions. The project will provide solutions to some of the most difficult and important technical questions surrounding the current digital-money revolution. The investigators will organize a crypto-currency speaker series that will bring together technologists, economists, social scientists, and policy-makers to foster collaborations that will shape the future of digital currencies. This collaboration with Elaine Shi, Michael Hicks, and Jonathan Katz is supported by the National Science Foundation Secure & Trustworthy Cyberspace program.
- 2015— **Sound Over- & Under-Approximations of Complexity & Information Security,** Co-PI.
Critical components of national cyberinfrastructure have been hardened to withstand traditional exploit-based attacks that target erroneous program behavior to compromise security. Looking forward such systems will be subject to a new class of attacks that do not rely on errors but rather exploit the inherent resource usage of the services the infrastructure provides. This project investigates techniques and tools for the systematic and scalable human-aided, automated resource analysis of large scale software systems for which source code is not available.
This collaboration with Michael Hicks, Elaine Shi, Eric Koskinen, and Dawn Song is supported by the DARPA Information Innovation Office, Space/Time Analysis for Cybersecurity program.

Projects (continued)

- 2014— **Trustworthy and Composable Software Systems with Contracts**, PI.
Over the past decade, language-based security mechanisms—such as type systems, model checkers, symbolic executors, and other program analyses—have been successfully used to uncover or prevent many important (exploitable) software vulnerabilities, such as buffer overruns, side channels, unchecked inputs (leading to code injection), and race conditions, among others. But despite significant advances, current work makes two unrealistic assumptions: (1) the analyzed code comprises a complete program (as opposed to a framework or set of components), and (2) the software is written in a single programming language. These assumptions ignore the reality of modern software, which is composed of large sets of interacting components constructed in several programming languages that provide varying degrees of assurance that the components are well-behaved. In this project, we aim to address these limitations by developing new static-analysis techniques based on software contracts, which provide a way to extend the analysis of components to reason about security of an entire heterogeneous system.
This collaboration with Sam Tobin-Hochstadt, Jeff Foster, and Michael Hicks, is supported by the National Security Agency, Science of Security Research Lablet program.
- 2012— **Behavioral Software Contract Verification**, Co-PI.
Behavioral software contracts express invariants and agreements between components of a program (procedures, modules, classes, even different languages) and assign blame to the appropriate party whenever these agreements are violated. Such contracts tend to be formulated in the full programming language, allowing arbitrary properties to be encoded a programs. While this is crucial for constructing reliable components, it thwarts static reasoning and incurs significant run-time monitoring costs. This work rectifies the situation with tools for modular and compositional automated reasoning about behavioral contracts. This collaboration with Sam Tobin-Hochstadt is supported by the National Science Foundation, Software and Hardware Foundations program.
- 2012— **Scalable and precise abstractions of programs for trustworthy software**, PI.
Applications deployed on mobile devices play a critical role in our daily life. They carry sensitive data and have capabilities with significant social and financial effect. Yet while it is paramount that such software is trustworthy, these applications pose challenges beyond the reach of current practice for low-cost, high-assurance verification and analysis. The primary goal of this project is to enable sound, secure, automatic program analysis for the elimination of security vulnerabilities in mobile applications in high-level programming languages.
This collaboration with Matthew Might is supported by the DARPA Information Innovation Office, Automated Program Analysis for Cybersecurity program.

Projects (continued)

- 2011–2013 **Raising the level of discourse with GnoSys**, Senior personnel.
The goal of this project is to design language mechanisms that capture design knowledge and to leverage this knowledge to qualitatively improve automated reasoning about programs. As part of the GnoSys project, I am investigating the interaction of analysis with language design, formal methods, and operating systems to enable mutually beneficial combinations for constructing robust systems. The focus of my work is to design program analysis tools for capturing domain knowledge and to design program abstractions that can be exploited by the components of the system such as the operating system and automated theorem prover.
This collaboration with Matthias Felleisen, Matthew Flatt, Pete Manolios, Matthew Might, Olin Shivers, and Mitchell Wand is supported by the DARPA Clean-slate design of Resilient, Adaptive, Secure Hosts (CRASH) program.

Publications: Journals and conferences

- 2016 **Pushdown Control-Flow Analysis for Free**. With Thomas Gilray, Steven Lyde, Michael D. Adams, and Matthew Might. *The 43rd ACM SIGPLAN-SIGACT Symposium on Principles in Programming Languages (POPL'16)*, St. Petersburg, Florida, January 2016.
- Higher-order symbolic execution for contract verification and refutation**. With Phuc C. Nguyen and Sam Tobin-Hochstadt. *Journal of Functional Programming*, Accepted, to appear 2016. Special Issue for “Best of ICFP 2014.”
- 2015 **Galois Transformers and Modular Abstract Interpreters**. With David Darais and Matthew Might. *Proceedings of the ACM SIGPLAN Conference on Object-Oriented Programming, Systems, Languages, and Applications (OOPSLA'15)*, Pittsburgh, Pennsylvania, October 2015.
- Incremental Computation with Names**. With Matthew A. Hammer, Joshua Dunfield, Kyle Headley, Nicholas Labich, Jeffrey S. Foster, and Michael Hicks. *Proceedings of the ACM SIGPLAN Conference on Object-Oriented Programming, Systems, Languages, and Applications (OOPSLA'15)*, Pittsburgh, Pennsylvania, October 2015.
- Relatively Complete Counterexamples for Higher-Order Programs**. With Phuc C. Nguyen. *Proceedings of the 36th ACM SIGPLAN Conference on Programming Language Design and Implementation (PLDI'15)*, Portland, Oregon, June 2015.
- Running Probabilistic Programs Backwards**. With Neil Toronto and Jay McCarthy. *Proceedings of the European Symposium on Programming (ESOP'15)*, London, United Kingdom, April, 2015.
- 2014 **Abstracting Abstract Control**. With J. Ian Johnson. *Proceedings of the 10th ACM Symposium on Dynamic Languages (DLS'14)*, Portland, Oregon, October 2014.
- Soft Contract Verification**. With Phuc C. Nguyen and Sam Tobin-Hochstadt. *Proceedings of the ACM SIGPLAN International Conference on Functional Programming (ICFP'14)*, Gothenburg, Sweden, September 2014.

Publications: Journals and conferences (continued)

- Pruning, Pushdown Exception-Flow Analysis.** With Shuying Liang, Weibin Sun, Matthew Might, and Andrew W. Keep. *Proceedings of the 14th IEEE International Conference on Software Code Analysis and Manipulation*, Victoria, British Columbia, September 2014.
- Pushdown flow analysis with abstract garbage collection.** With J. Ian Johnson, Ilya Sergey, Christopher Earl, and Matthew Might. *Journal of Functional Programming*, **24**(2-3), May 2014. Special Issue for “Best of ICFP 2012.”
- 2013 **Optimizing Abstract Abstract Machines.** With J. Ian Johnson, Nicholas Labich, and Matthew Might. *Proceedings of the 18th ACM SIGPLAN International Conference on Functional Programming (ICFP’13)*, Boston, Massachusetts, September, 2013.
- 2012 **Higher-Order Symbolic Execution via Contracts**, with Sam Tobin-Hochstadt. *The ACM SIGPLAN Conference on Object-Oriented Programming, Systems, Languages, and Applications (OOPSLA’12)*, Tuscon, Arizona, October, 2012.
- Introspective Pushdown Analysis of Higher-order Programs.** With Christopher Earl, Ilya Sergey, and Matthew Might. *Proceedings of the 17th ACM SIGPLAN International Conference on Functional Programming (ICFP’12)*, Copenhagen, Denmark, September, 2012.
- Systematic Abstraction of Abstract Machines**, with Matthew Might. *Journal of Functional Programming*, **22**(4–5), Special Issue for “Best of ICFP 2010.”
- Subcubic Control Flow Analysis Algorithms.** With Jan Midtgaard. To appear in *Higher-Order and Symbolic Computation*.
- 2011 **Abstracting Abstract Machines: A Systematic Approach to Higher-Order Program Analysis.** With Matthew Might. In *Communications of the ACM, Research Highlights* **54**(9), September, 2011.
- A Family of Abstract Interpretations for Static Analysis of Concurrent Higher-Order Programs**, In *The 18th International Static Analysis Symposium (SAS 2011)*, Venice, Italy, September, 2011. Lecture Notes in Computer Science, 6887.
- 2010 **Abstracting Abstract Machines.** With Matthew Might. In *Proceedings of the 15th ACM SIGPLAN International Conference on Functional Programming (ICFP’10)*, Baltimore, Maryland, September, 2010.
- Implementing Call-By-Need on the Control Stack.** With Stephen Chang and Matthias Felleisen. In *Symposium on Trends in Functional Programming (TFP’10)*, Norman, Oklahoma, May, 2010. Winner of the best student paper award.
- Resolving and Exploiting the k -CFA Paradox: Illuminating Functional vs. Object-Oriented Program Analysis.** With Matthew Might and Yannis Smaragdakis. In *Proceedings of the ACM SIGPLAN 2010 Conference on Programming Language Design and Implementation (PLDI’10)*, Toronto, Canada, June, 2010.

Publications: Journals and conferences (continued)

- 2008 **Deciding k CFA is complete for EXPTIME**, with Harry G. Mairson. In *Proceedings of the 13th ACM SIGPLAN International Conference on Functional Programming (ICFP'08)*, Victoria, British Columbia, Canada, September, 2008.
- Flow Analysis, Linearity, and PTIME**, with Harry G. Mairson. In *The 15th International Static Analysis Symposium (SAS 2008)*, Valencia, Spain, July, 2008. Lecture Notes in Computer Science, 5079.
- Types and Trace Effects of Higher Order Programs**, with Christian Skalka and Scott Smith. *Journal of Functional Programming* 18(2), March, 2008.
- 2007 **Relating Complexity and Precision in Control Flow Analysis**, with Harry G. Mairson. In *Proceedings of the Twelfth ACM SIGPLAN International Conference on Functional Programming (ICFP'07)*, Freiburg, Germany, October, 2007.

Publications: Workshops and others

- 2016 **Tutorial: An Introduction to Redex with Abstracting Abstract Machines**. In *Tutorials at The 43rd ACM SIGPLAN-SIGACT Symposium on Principles in Programming Languages (POPL'16)*, St. Petersburg, Florida, January 2016.
- Mechanically Verified Computational Abstract Interpretation**. With David Darais. <http://arxiv.org/abs/1507.03559>
- Constructive Galois Connections**. With David Darais. <http://arxiv.org/abs/1511.06965>
- 2013 **Sound and Precise Malware Analysis for Android via Pushdown Reachability and Entry-Point Saturation**, with Shuying Liang, Andrew Keep, Matthew Might, Steven Lyde, Thomas Gilray, and Petey Aldous. In *ACM CCS Workshop on Security and Privacy in Smartphones and Mobile Devices (SPSM)*, Berlin, Germany, November 2013.
- Static Contract Checking for Scripting Languages**, with Phuc C. Nguyen and Sam Tobin-Hochstadt. <http://arxiv.org/abs/1307.6239>
- From Principles to Practice with Class in the First Year**, with Sam Tobin-Hochstadt. In *International Workshop on Trends in Functional Programming in Education*, Provo, Utah, May 2013.
- Concrete Semantics for Pushdown Analysis: The Essence of Summarization**, with J. Ian Johnson. In *Workshop on Higher-Order Program Analysis*, New Orleans, Louisiana, June 2013.
- AnaDroid: Malware Analysis of Android with User-supplied Predicates**, with Shuying Liang and Matthew Might. In *Workshop on Tools for Automatic Program Analysis*, Seattle, Washington, June 2013.

Publications: Workshops and others (continued)

- 2011 **Semantic Solutions to Program Analysis Problems**, with Sam Tobin-Hochstadt. In *The ACM SIGPLAN 2011 Conference on Programming Language Design and Implementation (PLDI'11)*, FIT Session, San Jose, California, June 2011.
- 2010 **Pushdown Control-Flow Analysis of Higher-Order Programs**, with Christopher Earl and Matthew Might. In *The 2010 Workshop on Scheme and Functional Programming (SFP'10)*, Montréal, Québec, Canada, August, 2010.
- 2009 **The Complexity of Flow Analysis in Higher-Order Languages**. Ph.D. dissertation, Brandeis University, August, 2009.
- 2008 **A Few Principles of Macro Design**, with David Herman. In *The ACM Workshop on Scheme and Functional Programming*, Victoria, British Columbia, Canada, September, 2008.
- 2006 **Algorithmic Trace Effect Analysis**. Masters thesis, University of Vermont, May, 2006.
- 2005 **A Type and Effect System for Flexible Abstract Interpretation of Java**, with Christian Skalka and Scott Smith. In *Proceedings of the ACM Workshop on Abstract Interpretation of Object-Oriented Languages (AIOOL'05)*, Electronic Notes in Theoretical Computer Science. Volume 131, January, 2005.

Publications: Books

- 2013 **Realm of Racket**, with Matthias Felleisen, Conrad Barski, and Northeastern undergraduates: Forrest Bice, Rose DeMaio, Spencer Florence, Feng-Yun Mimi Lin, Scott Lindeman, Nicole Nussbaum, Eric Peterson, Ryan Plessner. Published by No Starch Press.

Teaching

- University of Maryland**
2014–2015 Program Analysis and Understanding, CMSC 631 (40)
2015–26 Compilers, CMSC 430 (90)
- Northeastern University**
2007–2010 Intro. to Programming and Computing I (247)
2009–2011 Intro. to Programming and Computing I, Honors (134)
2008–2009, Intro. to Programming and Computing II (312)
2012–2013
2011–2013 Intro. to Programming and Computing II, Honors (110)

Service: External

- Journal, editor**
2016 Journal of Functional Programming, Special Issue for “Best of ICFP 2015”
- Journal, referee**

Service: External (continued)

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

Steering committee

2013–2016 ACM SIGPLAN International Conference on Functional Programming (ICFP)
2014 Workshop on Higher-Order Program Analysis (HOPA)

Chair

2016 Symposium on Trends in Functional Programming (TFP)
2014 Workshop on Higher-Order Program Analysis (HOPA)
2011 NII Workshop on Automated Techniques for Higher-Order Program Verification
New England Programming Languages and Systems Symposium (NEPLS)

Panelist

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

Program committee

2017 ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages (POPL)
2016 European Conference on Object-Oriented Programming (ECOOP)
2015 ACM SIGPLAN International Conference on Functional Programming (ICFP)
Off the Beaten Track: New Frontiers for Programming Languages Research (OBT)
2014 European Symposium on Programming (ESOP)
International Symposium on Practical Aspects of Declarative Languages (PADL)
Workshop on Syntax and Semantics of Low-Level Languages (LOLA)
Symposium on Trends in Functional Programming (TFP)
International Workshop on Trends in Functional Programming in Education (TFPIE)
2013 Scala Workshop
Workshop on Higher-Order Program Analysis (HOPA)
2012 Symposium on Trends in Functional Programming (TFP)
2011 ACM SIGPLAN International Conference on Functional Programming (ICFP)
Scheme and Functional Programming Workshop
2009 Scheme and Functional Programming Workshop

External review committee

2016 ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages (POPL)
2013 ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages (POPL)

Reviewer

2016 International Conference on Verification, Model Checking, and Abstract Interpretation (VMCAI)
2015 ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages (POPL)
2014 ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages (POPL)
International Conference on Verification, Model Checking, and Abstract Interpretation (VMCAI)
ACM SIGPLAN International Conference on Functional Programming (ICFP)

Service: External (continued)

- 2012 ACM SIGPLAN Symposium on Dynamic Languages (DLS)
ACM SIGPLAN Conference on Object-Oriented Programming, Systems, Languages, and Applications (OOPSLA)
Dynamic Languages Symposium (DLS)
- 2011 European Symposium on Programming Languages (ESOP)
- 2010 ACM SIGPLAN International Conference on Functional Programming (ICFP)
- 2008 ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages (POPL)
- 2007 IEEE Symposium on Logic in Computer Science (LICS)
EACSL Conference on Computer Science and Logic (CSL)
- Other**
- 2013–2016 ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages (POPL), Workshop chair
- 2012 New England Programming Languages and Systems Symposium (NEPLS), Speaker selection committee.

Service: Internal

- 2014–2016 Graduate Admissions Committee.
- 2015–2016 Graduate Student Review Committee.
- 2015–2016 Iribe Center Space Committee.
- 2015 Grad School Panel for first-year PhD students.
- 2015 Organizer: “Is Grad School Right for Me?” event of undergraduates.
- 2012 (Northeastern) CS3500 Committee, with Javed Aslam and William D Clinger, charged with reviewing the CS3500 (*Object-Oriented Design*) curriculum, with the end goal of revising CS3500 in such a way that (a) it retains the best aspects of the current object-oriented design course, (b) it does not excessively and unnecessarily overlap with CS2510 (*Fundamentals 2*), and (c) it contains substantially more algorithmic content.

Talks

- 2016 **Tutorial: Introduction to Redex with Abstracting Abstract Machines**, Tutorials at the ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages (POPL), January 2016.
Verification and Refutation of Behavioral Contracts with Higher-Order Symbolic Execution, PLEAID Seminar, University of Chile, Santiago, Chile, January 2016.
Tutorial: Introduction to Redex with Abstracting Abstract Machines, PLEAID Seminar, University of Chile, Santiago, Chile, January 2016.
- 2015 **Verification and Refutation of Behavioral Contracts with Higher-Order Symbolic Execution**, CS Seminar, Johns Hopkins University, 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.

Talks (continued)

- 2014
- 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.
 - 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.
- 2013
- From Principles to Practice with Class Trends in Functional Programming in Education**, Provo, Utah, May 2013.
 - Abstracting Definitional Interpreters**, Mid-Atlantic Programming Languages Seminar, College Park, Maryland, April 2013.
 - Analysis for Trustworthy Software**, Computer Science Colloquium, University of Maryland, College Park, Maryland, March 2013.
- 2012
- Analyzing Software Contracts**, DARPA Clean-slate design of Resilient Adaptive Secure Hosts meeting, Boston, Massachusetts, December 2012.
 - Towards the Verification of Behavioral Software Contracts**, Microsoft Research, RiSE Group invited lecture, Redmond, Washington, November 2012.
 - Raising the Level of Discourse with GnoSys**, DARPA Clean-slate design of Resilient Adaptive Secure Hosts PI Meeting, San Diego, California, November 2012.
 - Program Verification via Abstract Reduction Semantics, Optimized Machines for Program Analysis, Abstract Machines for Program Analysis** Invited lectures, Harvard University, Advanced Functional Language Compilation, Cambridge, Massachusetts, November 2012.
 - Scalable Abstractions for Trustworthy Software**, DARPA Automated Program Analysis for Cybersecurity PI Meeting, Arlington, Virginia, October, 2012.
- 2011
- Low-level Analysis for High-level Assurance**, GnoSys project report for DARPA CRASH, Boston, Massachusetts, October, 2011.
 - 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.
 - 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.
- 2010
- Modular Analysis via Abstract Reduction Semantics**, New Jersey Programming Languages and Systems Symposium, Rutgers University, Piscataway, New Jersey, December 2010.
 - Pushdown Control-Flow Analysis of Higher-Order Programs**, IBM Programming Languages Day, Hawthorne, New York, July 2010.

Talks (continued)

- 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, University of Oregon CIS Colloquium, Eugene, Oregon, April 2010.
- 2009 **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 in Higher-Order Languages, Ph.D. defense, Brandeis University, Waltham, Massachusetts, July 2009.
- 2008 **The Complexity of Flow Analysis**, New England Programming Languages and Systems Symposium, Harvard University, Boston, Massachusetts, November 2008.
The Complexity of Flow Analysis, Northeastern University Graduate Programming Languages Seminar, Boston, Massachusetts, October 2008.
- 2007 **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.
- 2006 **Linearity and Program Analysis**, Northeastern University, Graduate Programming Languages Seminar, Boston, Massachusetts, October 2006.
Algorithmic Trace Effect Analysis, MS thesis defense, University of Vermont, March 2006.
- 2005 **Abstract Machines for the Multi-return λ -calculus**, Northeastern University, “Principles of Programming Languages” graduate seminar, December 2005.
Algorithmic Trace Effect Analysis, University of Vermont CS Research Day, August 2005.
Context Based Security in Programming Languages, Vermont EPSCoR conference, August 2005.

Honors & Awards

- 2013 **Northeastern University Excellence in Teaching Award Nominee.**
- 2011 **Communications of the ACM, Research Highlight.**
- 2009—2011 **Computing Innovation Fellow**, CRA/CCC with funding from the NSF.
- 2009 **ACM Doctoral Dissertation Award Nominee.**
- 2004—2005 **Upsilon Pi Epsilon International Computer Science Honor Society.**
- 2003—2004 **ACM Faculty Award**, College of Engineering & Mathematics, University of Vermont.