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

www.cs.umd.edu/~dvanhorn
dvanhorn@cs.umd.edu
(617) 285-7894

David Van Horn

Education

Brandeis University
2009 Ph.D. in Computer Science, *The Complexity of Flow Analysis in Higher-Order Languages*

University of Vermont
2006 M.S. in Computer Science, *Algorithmic Trace Effect Analysis*
2003 B.S. in Computer Science & Information Systems

Professional experience

2013— **University of Maryland**, Assistant Professor

Northeastern University
2012—2013 Research Assistant Professor
2011—2012 Visiting Assistant Professor
2009—2011 CRA Computing Innovation Fellow
2007—2009 Visiting Lecturer

2005—2008 **Brandeis University**, Graduate Research & Teaching Assistant

2003—2005 **University of Vermont**, Graduate Research Assistant

Research interests

I am interested in all aspects of program analysis and its applications to programming languages, software engineering, verification, and security.

Grants

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.

2012–2015 **Scalable and Precise Abstractions of Programs for Trustworthy Software**. PI. Awarded by DARPA Information Innovation Office (I20), 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.

Current projects

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.

2011— **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

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.

Publications: Journals and conferences (continued)

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. Invited to submit to special issue of *Journal of Functional Programming*.

Systematic Abstraction of Abstract Machines, with Matthew Might. *Journal of Functional Programming*, **22**(4–5), Special Issue for 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.

2008 **Deciding kCFA 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 Twelth ACM SIGPLAN International Conference on Functional Programming (ICFP'07)*, Freiburg, Germany, October, 2007.

Publications: Workshops and others

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.

Publications: Workshops and others (continued)

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.

2011 **Semantic Solutions to Program Analysis Problems**, with Sam Tobin-Hochstadt. In *The 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

Northeastern University

2007–2010	Intro. to Programming and Computing I (247)
2009–2011	Intro. to Programming and Computing I, Honors (134)
2008–2009, 2012–2013	Intro. to Programming and Computing II (312)

Teaching (continued)

2011–2013	Intro. to Programming and Computing II, Honors (110)
	Brandeis University (TA)
2007–2008	Semantics of Programming Languages
2006–2007	Structure and Interpretation of Computer Programs
	University of Vermont (TA)
2006	Types in Programming Languages
2005	Programming Languages

External service

Journal referee
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
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
2014	European Symposium on Programming (ESOP)
	International Symposium on Practical Aspects of Declarative Languages (PADL)
	Symposium on Trends in Functional Programming (TFP)
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
	Reviewer
2012	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)

External service (continued)

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
2013	ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages (POPL), ERC
2012	New England Programming Languages and Systems Symposium (NEPLS), Speaker selection committee.

Internal service

2012 CS3500 Committee, with Javed Aslam and William D Clinger, charged with reviewing the CS3500 (*Object-Oriented Design*) cirriculum, 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.

Selected talks

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.

Selected talks (continued)

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.
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, 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.
The Complexity of Flow Analysis, Northeastern University, Graduate Programming Languages Seminar, Boston, Massachusetts, October.

2007 **Relating Complexity and Precision in Control Flow Analysis**, Northeastern University, Programming Languages Seminar, Boston, Massachusetts, May.
Relating Complexity and Precision in Control Flow Analysis, IBM Programming Languages Day, Hawthorne, New York, May.

2006 **Linearity and Program Analysis**, Northeastern University, Graduate Programming Languages Seminar, Boston, Massachusetts, October.
Proofnets and Paths in Constructive Classical Logic: Too Old, Too New, Geometry of Interaction Workshop, Geocal'06, Marseille, February.
Algorithmic Trace Effect Analysis, MS thesis defense, University of Vermont, March.

Selected talks (continued)

2005 **Abstract Machines for the Multi-return λ-calculus**, Northeastern University, "Principles of Programming Languages" graduate seminar, December.
Algorithmic Trace Effect Analysis, University of Vermont CS Research Day, August.
Context Based Security in Programming Languages, Vermont EPSCoR conference, August.

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.