David Walker

Department of Computer Science
Princeton University
35 Olden St.
Princeton, NJ 08540

Phone: (609) 258-7654
Fax: (609) 258-1771
dpw@cs.princeton.edu
http://www.cs.princeton.edu/~dpw/

Positions: Princeton

Full Professor with tenure, Princeton University, February 2013-.

Associate Professor with tenure, Princeton University, July 2008-January 2013.

Assistant Professor, Princeton University, February 2002-June 2008.

Positions: Other

Associate Visiting Faculty, University of Pennsylvania, 2015-2016.

Visiting Researcher, Microsoft Research Cambridge, June-July 2010.

Visiting Researcher, Microsoft Research Redmond, September-December 2009.

Assistant Professor, Princeton University, February 2002-June 2008.

Post-doctoral Researcher, Carnegie Mellon University, October 2000-October 2001.

Education

Ph.D. Computer Science, Cornell University, Advisor: Greg Morrisett, 2001

Masters of Science, Computer Science, Cornell University, 1999

Bachelors of Science (Honors), Computer Science, Queen's University, 1995

Research Interests

Programming languages; type systems; domain-specific programming languages; network reliability, configuration synthesis and verification

Awards and Honors

PhD advisee Ryan Beckett wins the following awards for his thesis entitled *Network Control Plane Synthesis and Verification*:

- 2019 ACM SIGPLAN John C Reynolds Dissertation Award (Best thesis in the field of Programming Languages)
- 2019 ACM SIGCOMM Dissertation Award (Best thesis in the field of Networking)
- 2019 ACM Dissertation Award, Honorable Mention (Runner-up, with one other, for best thesis across all of Computer Science)

Don't Mind the Gap: Bridging Network-wide Objectives and Device-level Configurations (Brief Reflections on Abstractions for Network Programming). Ryan Beckett, Ratul Mahajan, Todd Millstein, Jitu Padhye and David Walker. ACM Computer and Communications Review (CCR), Volume 49, Issue 5, Oct 2019. Invited editorial in the special issue on the first fifty years of ACM SIGCOMM.

ACM SIGCOMM 2016 best paper award for the paper entitled "Don't Mind the Gap: Bridging Network-wide Objectives and Device-level Configurations." With Ryan Beckett, Ratul Mahajan, Todd Millstein, and Jitu Padhye.

Symposium on SDN Research (SOSR '16) best paper award for the paper entitled "CacheFlow: Dependency-Aware Rule-Caching for Software-Defined Networks." With Naga Katta, Omid Alipourfard, and Jennifer Rexford.

ACM SIGPLAN Robin Milner Young Researcher Award, 2015.

USENIX Symposium on Networked Systems Design and Implementation (NSDI) Community Award for the paper entitled "Composing Software-Defined Networks." April 2013. With Josh Reich, Christopher Monsanto, Nate Foster and Jennifer Rexford.

Most Influential 1998 POPL Paper Award for the paper entitled "From System F to Typed Assembly Language" presented at the 1998 ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages. Awarded January 2008. With Greg Morrisett, Karl Crary and Neal Glew.

ACM SIGPLAN Research Highlight for the paper entitled "Fault-Tolerant Typed Assembly Language," published in PLDI 2007. Nominated Semptember 2008. With Frances Perry, Lester Mackey, George A. Reis, Jay Ligatti, and David I. August.

ACM SIGPLAN Research Highlight for the paper entitled "The Next 700 Data Description Languages," published in POPL 2006. Nominated Semptember 2008. With Kathleen Fisher and Yitzhak Mandelbaum.

ACM SIGPLAN Conference on Programming Language Design and Implementation (PLDI) Best Paper Award for the paper entitled "Fault-tolerant Typed Assembly Language," 2007. With Frances Perry, Lester Mackey, George A. Reis, Jay Ligatti, and David I. August.

Emerson Junior Faculty Award for Excellence in Research and Teaching, 2005

Alfred P. Sloan Fellow, 2004

NSF Career Award, 2003

Cornell Computer Science Outstanding TA Award. May 1996.

Prince of Wales Prize, Honorable Mention, 1995 (Awarded to the student with the 2nd highest standing in faculty of Arts and Science, Queens University)

R. W. Leonard Penultimate Year Scholarship, 1994 (Awarded to the student with the highest standing through 3 years of Bachelors of Science, Queens University)

General Academic Service

ACM SIGPLAN CARES (Committee to Aid Reporting on Discrimination and Harassment Policy Violations). Founding Co-chair. Dec 2019-

ACM SIGPLAN Symposium on Principles of Programming Languages. Steering Committee. 2015-2020.

ACM SIGPLAN Symposium on Principles of Programming Languages. Industrial Relations Chair. 2017-2020.

Associate Editor for ACM Transactions on Programming Languages and Systems. June 2007-2015.

Associate Editor for Foundations and Trends in Programming Languages. October 2012-.

Microsoft Think Tank on University Relations. 2009.

Core member, CCC Visioning Study on Multi-level Approaches to Reliability. November 2008-2010.

NSF-sponsored Summer School on Language-Based Techniques for Integrating with the External World. Steering Committee. July 2007.

NSF-sponsored Summer School on Language-Based Techniques for Concurrent and Distributed Software. Steering Committee. July 2006.

The Computer Science Futures Project, a DARPA-sponsored panel on the Future of Computer Science Research. Panelist, 2006.

NSF-sponsored Summer School on Reliable Computing. Organizing committee (co-chair). Eugene, OR, July 2005.

NSF-sponsored Summer School on Security: Theory to Practice. Organizing committee (co-chair). Eugene, OR, June 2004.

ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages. Publicity Chair. 2003-2006.

Panelist for a number of NSF funding programs.

Program Committees

European Symposium on Programming. Program Committee, 2019.

SIGCOMM Symposium on Software-Defined Networking Research. Program Committee, 2019.

ACM SIGPLAN Symposium on Principles of Programming Languages. Program Committee, 2018.

SIGCOMM Symposium on Software-Defined Networking Research. Program Committee, 2017.

Workshop on Programming Languages and Operating Systems. Program Committee, 2017.

ACM SIGPLAN Conference on Programming Language Design and Implementation, Program Committee (ERC), 2017.

ACM Symposium on SDN Research, Program Committee, 2017.

ACM SIGPLAN Conference on Functional Programming, Program Committee, 2016.

ACM SIGPLAN Conference on Programming Language Design and Implementation, Program Committee (light), 2016.

ACM SIGPLAN Conference on Principles of Programming Languages, Program Chair, 2015.

ACM SIGCOMM Symposium on SDN Research (SOSR), Program Committee, 2015.

PLVNET 2015: 1st Workshop on Programming Languages and Verification Technology for Networking. Program Committee, 2015.

The OCaml Users and Developers Workshop. Program Committee, 2013.

ACM SIGPLAN Conference on Principles of Programming Languages, Program Committee, 2013.

Off the Beaten Track: Underrepresented Problems for Programming Language Researchers, General Chair, Program Committee. 2013.

ACM SIGPLAN Conference on Principles of Programming Languages, ERC, 2012.

ACM Conference on Principles of Databases. ERC, 2012.

Off the Beaten Track: Underrepresented Problems for Programming Language Researchers, Program Chair, 2012

Programming Languages meet Program Verification, Program Committee, 2012.

20th European Symposium on Programming, Program Committee, 2011.

ACM SIGPLAN Conference on Programming Language Design and Implementation, Program Committee, 2010.

21st Symposium on Implementation and Application of Functional Languages, Program Committee, 2009.

ACM SIGPLAN Conference on Object-Oriented Systems, Languages, Programing and Applications (OOPSLA), Program Committee, 2009.

ACM SIGPLAN Workshop on Programming Languages and Analysis for Security, Program Committee, 2007.

Trends in Functional Programming, Program Committee, 2007.

ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, Program Committee, 2007.

ACM SIGPLAN International Symposium on Code Generation and Optimization, Program Committee, 2007.

Workshop on Foundations of Object-Oriented Languages, Program Committee, 2007.

ACM/NSF Summer School on Language-Based Techniques for Concurrent and Distributed Software, Steering Committee, July 2006.

33rd International Colloquium on Automata, Languages and Programming. Program Committee, 2006.

Workshop on Foundations of Aspect-Oriented Languages, Program Committee, 2006

Workshop on Semantics, Program Analysis and Computing Environments for Memory Management, Program Co-chair, 2006.

Workshop on Foundations of Aspect-Oriented Languages, Program Chair, 2005.

Workshop on Logics for Resources, Processes and Programs, Program Committee, 2004.

ACM SIGPLAN International Conference on Functional Programming, Program Committee, 2004.

ACM SIGPLAN International Workshop on Types in Language Design and implementation, Program Committee, 2004.

Workshop on Foundations of Aspect-Oriented Languages, Program Committee, March 2004.

New Jersey Programming Languages Seminar, Host, September 2003.

New Jersey Programming Languages Seminar, Program Chair, December, 2002.

Workshop on Partial Evaluation and Semantics-based Program Manipulation, Program committee, January 2002.

Princeton Service

Policy Committee on Athletics and Campus Recreation. Spring 2019.

C/3 Faculty Advisory Committee on Appointments and Advancements. 2017-2018.

Computer Science Department Lecturer Chair. Sept 2016-

Faculty Advisory Committee on Athletics and Campus Recreation, Sept 2014-June 2015, 2016-2017.

Academic-Athletic Fellow, Men's Hockey, Spring 2015-.

Computer Science Department Representative for Undergraduate Affairs, 2010-2014.

Princeton SEAS Curriculum Committee, 2010-2015.

Committee on Undergraduate Admission and Financial Aid 2007-2009, 2012-2013.

Computer Science Independent Work Co-ordinator, 2007-2009.

Computer Science BSE '09 academic advisor.

Freshman advisor 2004-2006, 2008-2009, 2011-2012, 2014-2015.

Former Post-doctoral Students

Michael Greenberg. February 2014-June 2015.
First position: Assistant professor, Pomona College

Josh Reich (Co-advised with Jen Rexford). September 2011-2013.
First position: AT&T Research.

Nate Foster. (Co-advised with Jen Rexford) September 2009-July 2010.
First position: Assistant professor, Cornell University

Kenny Q. Zhu. January 2006-August 2009.
First Position: Assistant Professor, Shanghai University.

Graduated Ph.D. Students

Ryan Beckett. Graduated Sept 2018.
Thesis title: Network Control Plane Synthesis and Verification. First position: Microsoft Research, Redmond.

Cole Schlesinger. Graduated June 2015.
Thesis title: Abstractions for Software-defined Networks. First position: Researcher, Samsung.

Christian J Bell. Graduated March 2014.
Thesis title: A Proof Theory for Loop-Parallelizing Transformations. First position: Post-doctoral researcher, MIT.

Frances Perry. Graduated August 2008.
Thesis title: Reasoning about Software in the Presence of Transient Faults.
First position: Researcher, Google.

Limin Jia. Graduated November 2007.
Thesis title: Linear Logic and Imperative Programming.
First position: Post-doctoral researcher, University of Pennsylvania.

Daniel S. Dantas. Graduated August 2007.
Thesis title: Analyzing Security Advice in Functional Aspect-Oriented Programming Languages.
First Position: Consultant.

Yitzhak Mandelbaum. Graduated August 2006.
Thesis title: The Theory and Practice of Data Description.
First position: Researcher, AT&T Labs Research.

Jay Ligatti. Graduated May 2006.
Thesis title: Policy Enforcement via Program Monitoring.
First position: Assistant Professor, University of South Florida.

Graduated Masters Students

Jonathan Frankle. 2014-2015.
Thesis title: Type-directed synthesis of products.

Ashkay Mittal. 2012-2014.

Nayden Nedev. 2011-2013.

Rob Harrison. Graduated May 2011.
Thesis title: Frenetic: A Network Programming Language.
Co-advised with Jen Rexford. First position: West Point Military Academy.

Current Post-doctoral Students

Praveen Tammana (co-advised with Jen Rexford).

Current Ph.D. Students

Devon Loehr. Supervised since Sept 2018.

Nick Giannarakis. Supervised since May 2018.

Anders Milter. Supervised since Sept 2015.

Current Masters Students

None

Undergraduate Research Advising

Yanjun Yang, Spring 2019, Fall/Spring 2019-2020.

Jake Waksbaum, Spring 2019

Fabian Roberts, Fall 2017

Leila Clark, Spring 2017

Vibhaalakshmi Sivaraman, Fall 2016-Spring 2017.

Lachie Kermode, Fall 2016

Evelyn Ding, Independent work, Spring 2015. (Co-advised with Barbara Englehardt)

Katherine Ye. Independent Work Project, Spring 2015.

- CRA Undergraduate Computer Science Research Awared Winner in Fall 2015

Ben Grange, Independent Work Project, Spring 2015

Mark Fillmore, Senior thesis, 2014-2015

Andrew Grasso, Senior thesis, 2014-2015

Jonathan Frankle. Fall 2014.

Ed Walker. Fall 2014.

Greg Owen, Independent Work Project, Spring 2014

Bobby Ullman, Independent Work Project, Spring 2014

Noah Apthorpe, Independent Work Project, Spring 2014

Vladimir Costescu, Independent Work Project, Spring 2014

Vladimir Costescu. Fall 2012-Spring 2013. Fall-Spring 2013-2014

Mario Alvarez. Fall 2012.

Dana Butnariu. Fall 2012.

Nathan Swaney. Implementing Map Reduce in O'Caml. Spring 2012.

Travis Perlee. Lambda the gathering. Fall 2011.

Adriana Susnea. A high-level parallel language for querying and visualizing CoMon Data. Spring 2011.

Patrick Wendell, Senior Independent Work, Fall 2010

Alex Ogier, Junior Independent Work, Fall 2010

Adam Sanders, Senior Thesis, Fall 2008-Spring 2009

Michael Dirolf, Senior Independent Work, Fall 2007

Aaron Potechin, Summer Research, 2007

David Costanzo, Summer Research, 2007; Senior Independent Work, Fall 2007

Zach Devito, Junior Independent Work, Fall 2006

Ben DeLoache, Senior Independent Work, Fall 2006

Lester Mackey, Junior Independent Work, Spring 2006; Summer Research 2006

- CRA Undergraduate Computer Science Research Awared Winner in Fall 2007

Jin Oh, Senior Independent Work, Spring 2006

Mark Daly, Senior Thesis, Fall 2005-Spring 2006

- Princeton Computer Science Department Senior Thesis Award, co-winner Mark Daly, 2006.

Michael Ten-Pow, Junior Independent Work. Fall 2005

Rob Simmons, Senior Thesis, Fall 2004-Spring 2005

- Princeton Computer Science Department Senior Thesis Award, co-winner Rob Simmons, 2005.

Jonathon Heinberg, Junior Independent Work, Fall 2002-Spring 2003

Bismark Paliz, Senior Independent Work, Fall 2002

Teaching

COS 226: Introduction to Algorithms and Data Structures (Precepting: Spring 07, Spring 11)

COS 326: Functional Programming (Fall 12, Fall 13, Fall 14, Fall 16, Fall 17, Fall 18, Fall 19)

COS 320: Compiling Techniques (Spring 03, Spring 04, Spring 05, Spring 06)

COS 441: Programming Languages (Undergraduate) (Fall 05, Fall 07, Fall 08, Fall 11)

COS JIW/397/SRT/497: Undergraduate Independent Research Program (Fall 07-Spring 09)

COS 510: Programming Languages (Graduate) (Fall 02, Fall 03, Spring 14)

COS 597: Computer Security Foundations (Fall 04)

COS 598: Foundations of Language-Based Security (Spring 02)

COS 598: Parallelism (Fall 10)

COS 598: Reasoning about Networks (Spring 14, Spring 19)

TACL Seminar: Princeton's research seminar on programming languages and compiler technology (Periodically)

Publications

2020

1. Adaptive Weighted Traffic Splitting in Programmable Data Planes. Kuo-Feng Hsu (Rice University), Praveen Tammana (Princeton University), Ryan Beckett (Microsoft Research), Ang Chen (Rice University), Jennifer Rexford (Princeton University), David Walker (Princeton University). ACM SIGCOMM Symposium on SDN Research (SOSR). March 2020.
2. Contra: A programmable system for performance-aware routing. Kuo-Feng Hsu, Ryan Beckett, Ang Chen, Jennifer Rexford, Praveen Tammana, and David Walker. Networked Systems Design and Implementation (NSDI), Feb 2020.
3. Enabling programmable transport protocols in high-speed NICs. Mina Tahmasbi Arashloo, Alexey Lavrov, Manya Ghobadi, Jennifer Rexford, David Walker, and David Wentzlaff. Networked Systems Design and Implementation (NSDI), Feb 2020.
4. Abstract interpretation of distributed network control planes. Ryan Beckett, Aarti Gupta, Ratul Mahajan and David Walker. ACM SIGPLAN Symposium on Principles of Programming Languages (POPL), Jan 2020.

2019

5. Don't Mind the Gap: Bridging Network-wide Objectives and Device-level Configurations (Brief Reflections on Abstractions for Network Programming). Ryan Beckett, Ratul Mahajan, Todd Millstein, Jitu Padhye and David Walker. ACM Computer and Communications Review (CCR), Volume 49, Issue 5, Oct 2019. Invited editorial in the special issue on the first fifty years of ACM SIGCOMM.
6. NV: An intermediate language for network verification. Ryan Beckett, Nick Giannarakis, Devon Loehr and David Walker. ACM SIGCOMM Workshop on Networking and Programming Languages (NetPL), Aug 2019.
7. Synthesizing Symmetric Lenses. Anders Miltner, Solomon Maina, Kathleen Fisher, Benjamin C. Pierce, David Walker, and Steve Zdancewic. ACM SIGPLAN International Conference on Functional Programming (ICFP), Aug 2019.
8. Efficient verification of network fault tolerance via counterexample-guided refinement. Nick Giannarakis, Ryan Beckett, Ratul Mahajan and David Walker. International Conference on Computer-Aided Verification (CAV), July 2019.

2018

9. Synthesizing Quotient Lenses. Solomon Maina, Anders Miltner, Kathleen Fisher, Benjamin Pierce, David Walker and Steve Zdancewic. ACM SIGPLAN International Conference on Functional Programming. September 2018.
10. Control Plane Compression. Ryan Beckett, Aarti Gupta, Ratul Mahajan and David Walker. ACM SIGCOMM, August 2018
11. Synthesizing Bijective Lenses. Anders Miltner, Kathleen Fisher, Benjamin C. Pierce, David Walker, and Steve Zdancewic. ACM Symposium on Principles of Programming Languages (POPL), January 2018.

2017

12. HotCocoa: Hardware Congestion Control Abstractions. Mina Tahmasbi Arashloo, Monia Ghobadi, Jennifer Rexford, and David Walker. ACM HotNets, November 2017.
13. A General Approach to Network Configuration Verification. Ryan Beckett, Aarti Gupta, Ratul Mahajan and David Walker. ACM SIGCOMM, August 2017.
14. Network Configuration Synthesis with Abstract Topologies. Ryan Beckett, Ratul Mahajan, Todd Millstein, Jitu Padhye and David Walker. ACM PLDI, June 2017.

2016

15. Hardware-Software Co-Design for Network Performance Measurement. Srinivas Narayana, Anirudh Sivaraman, Vikram Nathan, Mohammad Alizadeh, David Walker, Jennifer Rexford, Vimalkumar Jeyakumar, Changhoon Kim. ACM HotNets, November 2016.
16. Don't Mind the Gap: Bridging Network-wide Objectives and Device-level Mechanisms. Ryan Beckett, Ratul Mahajan, Todd Millstein, Jitu Padhye, and David Walker. ACM SIGCOMM, August 2016. **(Winner ACM SIGCOMM Best Paper Award)**
17. SNAP: Stateful Network-wide Abstractions for Packet Processing. Mina Arashloo, Yaron Koral, Michael Greenberg, Jennifer Rexford, and David Walker. ACM SIGCOMM, August 2016.
18. Temporal NetKAT. Ryan Beckett, Michael Greenberg and David Walker. ACM SIGPLAN Conference on Programming Language Design and Implementation (PLDI '16), June 2016.
19. Compiling Path Queries. Srinivas Narayana, Mina Arashloo, Jennifer Rexford, and David Walker. 13th USENIX Symposium on Networked Systems Design and Implementation (NSDI '16), May 2016.
20. CacheFlow: Dependency-Aware Rule-Caching for Software-Defined Networks. Naga Katta, Omid Alipourfard, Jennifer Rexford, David Walker. ACM Symposium on SDN Research (SOSR '16), March 2016. **Winner ACM SOSR Best Paper Award.**
21. Example-Directed Synthesis: A Type-Theoretic Interpretation. Jonathan Frankle, Peter-Michael, David Walker, and Steve Zdancewic. In ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages (POPL '16), January 2016.

2015

22. CoVisor: A Compositional Hypervisor for Software-Defined Networks. Xin Jin, Jennifer Gossels, Jennifer Rexford, and David Walker. 12th USENIX Symposium on Networked Systems Design and Implementation (NSDI '15), May 2015.
23. Tracking the Flow of Ideas through the Programming Languages Literature. Michael Greenberg, Kathleen Fisher, and David Walker. SNAPL: The Inaugural Summit on Advances in Programming Languages, May 2015.
24. Hone: Joint Host-Network Traffic Management in Software-Defined Neteworks. Peng Sun, Minlan Yu, Michael Freedman, Jennifer Rexford and David Walker. Journal of Network and Systems Management. Vol 23, Issue 2, pp 374-399. April 2015.

25. Temporal NetKAT. Ryan Beckett, Michael Greenberg and David Walker. PLVNET 2015: 1st Workshop on Programming Languages and Verification Technology for Networking. January 2015.

26. Type systems for SDN Controllers. Marco Gaboardi, Michael Greenberg and David Walker. PLVNET 2015: 1st Workshop on Programming Languages and Verification Technology for Networking. January 2015.

2014

27. Transparent, Live Migration of a Software-defined Network. Soudeh Ghorbani, Cole Schlesinger, Matthew Monaco, Eric Keller, Matthew Caesar, Jennifer Rexford, and David Walker. ACM Symposium on Cloud Computing. November, 2014.

28. Modular Protections against Non-control Data Attacks. Cole Schlesinger, Karthik Pattabiraman, Nikhil Swamy, David Walker, and Benjamin Zorn. Journal of Computer Security. Volume 22, Issue 6, November, 2014.

29. Concurrent NetCore: From Policies to Pipelines. Cole Schlesinger, Michael Greenberg and David Walker. ACM SIGPLAN International Conference on Functional Programming (ICFP). September 2014.

30. Compiling Path Queries in Software-defined Networks. Srinivas Narayana, Jennifer Rexford, and David Walker. ACM SIGCOMM HotSDN Workshop, August 2014.

31. An Assertion Language for Debugging SDN Applications. Ryan Beckett, X. Kelvin Zou, Shuyuan Zhang, Sharad Malik, Jennifer Rexford, and David Walker. ACM SIGCOMM HotSDN Workshop, August 2014.

32. Incremental Update for a Compositional SDN Hypervisor. Xin Jin, Jennifer Rexford, and David Walker. ACM SIGCOMM HotSDN Workshop, August 2014.

33. Infinite CacheFlow in Software-defined Networks. Naga Katta, Omid Alipourfard, Jennifer Rexford, and David Walker. ACM SIGCOMM HotSDN Workshop, August 2014. An earlier version appeared as Princeton Computer Science Technical Report TR-966-13, October 2013.

34. Programming Protocol-independent Packet Processors, Pat Bosshart, Dan Daly, Martin Izzard, Nick McKeown, Jennifer Rexford, Dan Talayco, Amin Vahdat, George Varghese and David Walker. Computer and Communications Review (CCR). Volume 44, Issue 3. July 2014.

35. NetKAT: Semantic Foundations for Networks. Carolyn Jane Anderson, Nate Foster, Arjun Guha, Jean-Baptiste Jeannin, Dexter Kozen, Cole Schlesinger, and David Walker. ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages. January 2014. (ACM Digital Library). Extended technical report issued by Cornell University October 2013 at http://hdl.handle.net/1813/34445.

36. The Frenetic Project: Declarative Languages for Programming Networks. Invited talk at the ACM Symposium on Practical Applications of Declarative Languages (PADL) 2014.

2013

37. An Efficient Distributed Implementation of One Big Switch. Nanxi Kang, Zhenming Liu, Jennifer Rexford and David Walker. Open Networking Summit, April 2013.
38. Languages for software-defined networks. Nate Foster, Michael J. Freedman, Arjun Guha, Rob Harrison, Naga Praveen Katta, Christopher Monsanto, Joshua Reich, Mark Reitblatt, Jennifer Rexford, Cole Schlesinger, Alec Story, and David Walker. IEEE Communication Magazine. 51(2):128-134, 2013.
39. Composing Software-defined Networks. Christopher Monsanto, Joshua Reich, Nate Foster, Jennifer Rexford, David Walker. 10th USENIX Symposium on Networked Systems Design and Implementation (NSDI '13). April 2013.

2012

40. Logic programming for software-defined networks. Naga Praveen Katta, Jennifer Rexford, and David Walker. Workshop on Cross-Model Language Design and Implementation (XLDI), September 2012.
41. Abstractions for Network Update. Mark Reitblatt, Nate Foster, Jennifer Rexford, Cole Schlesinger and David Walker. In ACM SIGCOMM, August 2012.
42. A Compiler and Run-time System for Network Programming Languages. Christopher Monsanto, Nate Foster, Rob Harrison, and David Walker. In ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, Philadelphia, PA, January 2012.
43. LearnPADS++: Incremental inference of Ad Hoc Data Formats. Kenny Q. Zhu, Kathleen Fisher and David Walker. ACM SIGPLAN International Symposium on Practical Aspects of Declarative Languages. January 2012.

2011

44. Consistent Updates for Software-Defined Networks: Change You Can Believe in! Mark Reitblatt, Nate Foster, Jennifer Rexford, and David Walker. In ACM Workshop on Hot Topics in Networks (HotNets), November 2011.
45. A Survey of the Practice of Computational Science. Prakash Prabhu, Thomas B. Jablin, Arun Raman, Yun Zhang, Jialu Huang, Hanjun Kim, Nick P. Johnson, Feng Liu, Soumyadeep Ghosh, Stephen Beard, Taewook Oh, Matthew Zoufaly, David Walker, David I. August. SC 11: The International Conference for High Performance Computing, Networking, Storage and Analysis. November 2011.
46. Frenetic: A Network Programming Language. Nate Foster, Rob Harrison, Michael J. Freedman, Christopher Monsanto, Jennifer Rexford, Alec Story, and David Walker. In ACM SIGPLAN International Conference on Functional Programming, September 2011.
47. Forest: A Language and Toolkit for Programming with Filestores. Kathleen Fisher, Nate Foster, David Walker and Kenny Q. Zhu. ACM SIGPLAN International Conference on Functional Programming, September 2011
48. Modular Protections against Non-control Data Attacks. Cole Schlesinger, Karthik Pattabiraman, Nikhil Swamy, David Walker and Ben Zorn. Computer Security Foundations Symposium. June 2011.

49. The PADS project: An Overview. Kathleen Fisher and David Walker. Invited paper. IEEE International Conference on Data Engineering. April 2011.

50. Linear Maps. Shuvendu Lahiri, Shaz Qadeer and David Walker. ACM SIGPLAN Workshop on Programming Languages meets Program Verification. January 2011.

2010

51. Frenetic: A High-Level Language for OpenFlow Networks. Nate Foster, Rob Harrison, Matthew L. Meola, Michael J. Freedman, Jennifer Rexford and David Walker. Workshop on Programmable Routers for Extensible Services of Tomorrow. November 2010.

52. Concurrent Separation Logic for Pipelined Parallelization. Christian J. Bell, Andrew Appel and David Walker. The 17th Annual Static Analysis Symposium. September 2010.

53. A Context-free Markup Language for Semi-structured Text. Qian Xi and David Walker. ACM SIGPLAN Conference on Programming Language Design and Implementation. June 2010.

54. Faulty Logic: Reasoning about Fault Tolerant Programs. Matthew L. Meola and David Walker. European Symposium on Programming. March, 2010.

55. Semantics and Algorithms for Data-dependent Grammars. Trevor Jim, Yitzhak Mandelbaum and David Walker. ACM SIGPLAN-SIGACT Symposium on Principles of Programming languages. January, 2010.

56. The Next 700 Data Description Languages. Kathleen Fisher, Yitzhak Mandelbaum and David Walker. Journal of the ACM. Volume 57, Issue 2, January 2010.

2009

57. An Overview of the Oregon Programming Languages Summer School. Jim Allen, Zena Ariola, Pierre-Louis Curien, Matthew Fluet, Jeff Foster, Dan Grossman, Robert Harper, Hugo Herbelin, Yannis Smaragdakis, David Walker and Steve Zdancewic. SIGPLAN Notices, Vol. 44, No. 11. November 2009.

58. Incremental Learning of System Log Formats. Kathleen Fisher, David Walker and Kenny Q. Zhu. Workshop on the Analysis of System Logs. October 2009.

59. Language Support for Processing Distributed Ad Hoc Data. Kenny Q. Zhu, Daniel S. Dantas, Kathleen Fisher, Limin Jia, Yitzhak Mandelbaum, Vivek Pai and David Walker. ACM SIGPLAN International Conference on Principles and Practice of Declarative Programming. September 2009.

60. Composing Expressive Run-time Security Policies. Lujo Bauer, Jay Ligatti, and David Walker. ACM Transactions on Software Engineering and Methodology, Volume 18, Issue 3, pp 9:1-9:43, May 2009.

61. Run-time Enforcement of Nonsafety Policies. Jay Ligatti, Lujo Bauer, and David Walker. ACM Transactions on Information and System Security, Volume 12, Issue 3, pp 1-41, January 2009.

62. Ad Hoc Data and the Token Ambiguity Problem. Qian Xi, Kathleen Fisher, Kenny Q. Zhu, and David Walker. ACM Symposium on Practical Applications of Declarative Languages, January 2009.

2008

63. Reasoning about Control Flow in the Presence of Transient Faults. Frances Perry and David Walker. 15th International Static Analysis Symposium, July 2008.
64. Reasoning About Faulty Programs. Matthew Meola, Frances Perry and David Walker. Second International Workshop on Proof-Carrying Code. June 2008.
65. Comparing Semantic and Syntactic Methods in Mechanized Proof Frameworks Christian Bell, Robert Dockins, Aquinas Hobor, Andrew W. Appel and David Walker. Second International Workshop on Proof-Carrying Code. June 2008.
66. LearnPADS: Fully Automatic Tool Generation From Ad Hoc Data. Kathleen Fisher, David Walker and Kenny Q. Zhu. ACM SIGMOD Demo Session. June 2008.
67. AspectML: A Polymorphic Aspect-oriented Functional Programming Language. Daniel S. Dantas, David Walker, Geoffrey Washburn, Stephanie Weirich. ACM Transactions on Programming Languages and Systems, Volume 30, Issue 3, pp 14:1-14:60, May 2008.
68. From Dirt to Shovels: Fully Automatic Tool Generation From Ad Hoc Data. Kathleen Fisher, David Walker, Kenny Q. Zhu and Peter White. ACM SIGPLAN-SIGACT Symposium on Principles of Programming languages. January 2008.

2007

69. Towards 1-click Tool Generation with PADS. David Burke, Kathleen Fisher, David Walker, Peter White and Kenny Q. Zhu. In the ICML-2007 Workshop on Challenges and Applications of Grammar Induction. June 2007.
70. Fault-tolerant Typed Assembly Language. Frances Perry, Lester Mackey, George A. Reis, Jay Ligatti, David I. August, and David Walker. ACM SIGPLAN Conference on Programming Language Design and Implementation. June 2007. Winner of the PLDI 07 Best Paper Award.
71. A Dual Semantics for the Data Description Calculus (Extended Abstract). Yitzhak Mandelbaum, Kathleen Fisher and David Walker. In the Eighth Symposium on Trends in Functional Programming, April 2007.
72. PADS/ML: A Functional Data Description Language. Yitzhak Mandelbaum, Kathleen Fisher, David Walker, Mary Fernandez, and Artem Gleyzer. ACM SIGPLAN-SIGACT Symposium on Principles of Programming languages. January 2007.

2006

73. A Type-Theoretic Interpretation of Pointcuts and Advice. Jay Ligatti, David Walker and Steve Zdancewic. Science of Computer Programming. Volume 63, Issue 3, pp 240-266. December 2006.
74. Expressing Heap-shape Contracts in Linear Logic. Frances Spalding, Limin Jia and David Walker. ACM SIGPLAN-SIGSOFT International Conference on Generative Programming and Component Engineering. October 2006.
75. Static Typing for a Faulty Lambda Calculus. David Walker, Lester Mackey, Jay Ligatti, George Reis, and David August. ACM SIGPLAN International Conference on Functional Programming. September 2006.

76. Mechanized Metatheory for User-Defined Type Extensions. Daniel Marino, Brian Chin, Todd Millstein, Gang Tan, Robert J. Simmons and David Walker. ACM SIGPLAN Workshop on Mechanizing Metatheory. September 2006.

77. PADS: An End-to-end System for Processing Ad Hoc Data. Mark Daly, Mary Fernandez, Kathleen Fisher, Robert Gruber, Yitzhak Mandelbaum, David Walker and Xuan Zheng. ACM SIGMOD demo. June 2006.

78. ILC: A Foundation for Automated Reasoning about Pointer Programs. Limin Jia and David Walker. European Symposium on Programming Languages. In Programming Languages and Systems, LNCS 3924, pp 131-145, Peter Sestoft editor. March 2006.

79. Making Extensibility of System Software Practical with the C4 Toolkit. Marco Yuen, Marc E. Fiuczynski, Robert Grimm, Yvonne Coady and David Walker. Workshop on Software Engineering Properties of Languages and Aspect Technologies. March 2006.

80. The Next 700 Data Description Languages. Kathleen Fisher, Yitzhak Mandelbaum and David Walker. ACM SIGPLAN-SIGACT Symposium on Principles of Programming languages. January 2006.

81. Harmless Advice. Daniel S. Dantas and David Walker. ACM SIGPLAN-SIGACT Symposium on Principles of Programming languages. January 2006.

82. LaunchPADS: A System for Processing Ad Hoc Data. Mark Daly, Mary Fernandez, Kathleen Fisher, Yitzhak Mandelbaum and David Walker. Demo Paper in PLAN-X 06: Programming Language Technologies for XML. January 2006.

2005

83. PolyAML: A Polymorphic Aspect-oriented Functional Programmming Language. Daniel S. Dantas, David Walker, Geoffrey Washburn and Stephanie Weirich. ACM SIGPLAN International Conference on Functional Programming. September 2005.

84. Enforcing Non-safety Security Policies with Program Monitors. Jay Ligatti, Lujo Bauer and David Walker. Tenth European Symposium on Research in Computer Security. September 2005.

85. Patch(1) Considered Harmful. Marc E. Fiuczynski, Robert Grimm, Yvonne Coady and David Walker. Workshop on Hot Topics in Operating Systems. June, 2005.

86. Certifying Compilation for a Language with Stack Allocation. Limin Jia, Frances Spalding, David Walker and Neal Glew. IEEE Symposium on Logic in Computer Science. June 2005.

87. A Refined Proof Theory for Reasoning About Separation. Limin Jia and David Walker. IEEE Symposium on Logic in Computer Science, short paper. June 2005.

88. Composing Security Policies in Polymer. Lujo Bauer, Jay Ligatti and David Walker. ACM SIGPLAN Conference on Programming Language Design and Implementation. June 2005.

89. Edit Automata: Enforcement Mechanisms for Run-time Security Policies. Jay Ligatti, Lujo Bauer and David Walker. International Journal of Information Security, Volume 4, Number 2, pp. 2-16, February 2005. ISSN: 1615-5262, Springer-Verlag.

90. Substructural Type Systems. Chapter 1 of Advanced Topics in Types and Programming Languages, Benjamin Pierce, ed., January 2005.

91. Harmless Advice. Daniel S. Dantas and David Walker. In Foundations of Object-Oriented Languages, January 2005.

2004

92. Dynamic Typing with Dependent Types (extended abstract). Xinming Ou, Gang Tan, Yitzhak Mandelbaum, and David Walker. In the 3rd IFIP International Conference on Theoretical Computer Science, August, 2004.
93. Specifying Properties of Concurrent Computations in CLF. Kevin Watkins, Iliano Cervesato, Frank Pfenning and David Walker. Workshop on Logical Frameworks and Meta-Logics. Cork, Ireland, July 2004.
94. Modal Proofs As Distributed Programs (extended abstract). Limin Jia and David Walker. In the European Symposium on Programming, LNCS 2986, David Schmidt (Ed.), pp. 219–233, Springer, April, 2004.
95. A Concurrent Logical Framework: The Propositional Fragment. Kevin Watkins, Iliano Cervesato, Frank Pfenning and David Walker. In S. Berari, M. Coppo, and F. Damiani, Ed, Types for Proofs and Programs, Lecture Notes in Computer Science 3085, Springer-Verlag, pages 355-377, 2004. Revised selected papers and from the Third International Workshop along the Types for Proofs and Program, Torino, Italy, April 2003.

2003

96. A Theory of Aspects. David Walker, Steve Zdancewic and Jay Ligatti. In the ACM SIGPLAN International Conference on Functional Programming, August 2003.
97. An Effective Theory of Type Refinements. Yitzhak Mandelbaum, David Walker and Robert Harper. In the ACM SIGPLAN International Conference on Functional Programming, August 2003.
98. Reasoning about Hierarchical Storage. Amal Ahmed, Limin Jia and David Walker. IEEE Symposium on Logic in Computer Science, pp. 33-44. Ottawa, Canada, June 2003.
99. Resource Usage Analysis Via Scoped Methods. Gang Tan, Xinming Ou and David Wnoalker. Foundations of Object-Oriented Languages. January, 2003.
100. The Logical Approach to Stack Typing. Amal Ahmed and David Walker. ACM workshop on Types in Language Design and Implementation. January, 2003.

2002

101. Types and Effects for Non-interfering Program Monitors. Lujo Bauer, Jarred Ligatti and David Walker. International Symposium on Software Security. Tokyo, November, 2002. Revised for printing in Software Security – Theory and Systems, LNCS 2609, Springer, pp 154–171. December 2002.
102. More Enforceable Security Policies. Lujo Bauer, Jarred Ligatti and David Walker. Workshop on Computer Security Foundations. Copenhagen, July 2002.
103. Stack-based Typed Assembly Language. Greg Morrisett, Karl Crary, Neal Glew, and David Walker. Journal on Functional Programming, 12(1):43-88, January 2002.

2001

104. On Linear Types and Regions. David Walker and Kevin Watkins. In ACM SIGPLAN International Conference on Functional Programming, September 2001.
105. On Linear Types and Regions. David Walker and Kevin Watkins. In the Workshop on Semantics, Program Analysis and Computing Environments for Memory Management. London, UK, January 2001.
106. Alias Types for Recursive Data Structures. David Walker and Greg Morrisett. Workshop on Types in Compilation. Montreal, Canada. Selected and revised papers printed in LNCS 2071 (Harper, ed.) March 2001.

2000

107. Typed Memory Management via Static Capabilities. David Walker, Karl Crary, and Greg Morrisett. ACM Transactions on Programming Languages and Systems, 22(4):701-771, July 2000.
108. Alias Types. Frederick Smith, David Walker, and Greg Morrisett. European Symposium on Programming. Published in Lecture Notes in Computer Science, Gert Smolka, editor, volume 1782, 366-381, Berlin, Germany, March 2000.
109. A Type System for Expressive Security Policies. David Walker. Twenty-Seventh ACM SIGPLAN Symposium on Principles of Programming Languages . pages 254-267, Boston, January 2000.

1999

110. A Type System for Expressive Security Policies. David Walker. In the FLOC '99 Workshop on Run-time Result Verification. Trento, Italy, July 1999.
111. From System F to Typed Assembly Language. Greg Morrisett, David Walker, Karl Crary, and Neal Glew. ACM Transactions on Programming Languages and Systems, 21(3):527-568, May 1999.
112. TALx86: A Realistic Typed Assembly Language. Greg Morrisett, Karl Crary, Neal Glew, Dan Grossman, Richard Samuels, Frederick Smith, Dave Walker, Stephanie Weirich, and Steve Zdancewic. In the ACM SIGPLAN Workshop on Compiler Support for System Software. pages 25-35, Atlanta, May 1999.
113. Typed Memory Management in a Calculus of Capabilities. Karl Crary, David Walker, and Greg Morrisett. Twenty-Sixth ACM SIGPLAN Symposium on Principles of Programming Languages. pages 262-275, San Antonio, January 1999.

1998

114. Stack-Based Typed Assembly Language. Greg Morrisett, Karl Crary, Neal Glew, and David Walker. 1998 Workshop on Types in Compilation (TIC '98). Kyoto, Japan. Published in Xavier Leroy and Atsushi Ohori, editors, Lecture Notes in Computer Science, volume 1473, pages 28-52. Springer, 1998.
115. From System F to Typed Assembly Language. Greg Morrisett, David Walker, Karl Crary, and Neal Glew. Twenty-Fifth ACM SIGPLAN Symposium on Principles of Programming Languages. pages 85-97, San Diego, January 1998.

Selected Talks and Lectures

Don't Mind the Gap: Bridging Network-wide Objectives and Device-level Mechanisms. CISCO PI Meeting, May 2016.

Confluences in Programming Languages Research. Invited talk at the Jay Modi Memorial Lecture. Drexel University, May 2016.

Confluences in Programming Languages Research. Invited talk at the ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages (POPL), 2016.

The Frenetic Project: Declarative Languages for Programming Networks. Distinguished Lecture Series, UT Dallas, 2015.

The Frenetic Project: Declarative Languages for Programming Networks. Invited talk at the ACM Symposium on Practical Applications of Declarative Languages (PADL '14) 2014.

Networking: A Killer App for Programming Language Researchers. Invited talk. MPI-SWS, Germany, January 2013.

Abstractions for Network Update. Invited talk. Harvard University, April 2012.

Frenetic: A Network Programming Language. Invited talk. Microsoft Research, Redmond. June, 2011.

Ad Hoc Data: From Uggh to Smug. David Walker. International Workshop on Relations and Data Integrity Constraints and Languages. May 2010.

Ad Hoc Data: Problems and Solutions. Google Seattle, December 2008.

Ad Hoc Data: Problems and Solutions. Microsoft Research, December 2008.

From Dirt to Shovels: Fully Automatic Tool Generation From Ad Hoc Data. AT&T UC Symposium, August 2007.

Summer School on Language-Based Techniques for Integrating with the External World. Invited lecturer. July 2007.

On Cosmic Rays, Bat Droppings and What We Can Do About Them. Yale University Computer Science Department Colloquium. November 2006.

On Cosmic Rays, Bat Droppings and What We Can Do About Them. IFIP Working Group 2.8. July 2006.

PADS/ML: A Functional Data Description Language. Carnegie Mellon POP Seminar. May 2006.

Logics for Checking Properties of Pointer Programs. Workshop on Intuitionistic Modal Logics and Applications. June 2005.

Stacks, Heaps and Regions: One Logic to Bind Them. Workshop on Semantics, Program Analysis, and Computing Environments for Memory Management. January, 2004.

Summer School on Foundations of Security. Invited lecturer. University of Oregon, June, 2003.

ACM State-of-the-art Summer School on Foundations of Internet Security. Invited lecturer. Poland, June 2002.

Poly stop a hacker. New Jersey Programming Languages Seminar. September, 2002.

Symposium on Cyber Security and Trustworthy Software. March, 2002.

Funding

Facebook Communications & Networking Research Award. $50,000. 03/2018.

NeTS: Medium: Collaborative Research: Network Configuration Synthesis: A Path to Practical Deployment (PI). $570,000 (Princeton portion). NSF CNS-1703493. 7/2017-.

ProNet: Programmable Networks Enabled by Fast In-Path Analytics (Jen Rexford PI). DARPA-16-41-DC-FP-011. $1,994,812. 4/2017-4/2021.

Synthesis and Verification of Distributed Control Planes. CISCO Faculty Gift. $91,425. 09/2016.

Synthesizing Data Wranglers (PI). DARPA BRASS FA8750-17-2-0028 (PRIME # DARPA HR0011623024). $423,891. 11/2016-10/2018.

Compiling High-level SDN Programming Languages to Modern Switch Hardware. CISCO Faculty Gift. $100,034. 01/2015.

Language and System Support for Managing Updates in Software-Defined Networks. Google Research Award. $33,690. July 2012.

Networks Opposing Botnets II (NoBot) (Co-PI). Office of Naval Research. $455,600. July 2012-June 2014.

TC: Large: Collaborative Research: High-level Language Support for Trustworthy Networks (PI). National Science Foundation CNS-1111520. $1,400,000. August 2011-July 2016.

SHF:Small:Language Support for Ad Hoc Data Processing (PI). National Science Foundation CCF-1016937. $500,000. August 2010-June 2012.

SI2-SSI: Accelerating the Pace of Research through Implicitly Parallel Programming (Co-PI). National Science Foundation OCI-1047879. $1,740,214 Oct 2010-Sep 2014.

Networks Opposing Botnets (NoBot) (Co-PI). National Science Foundation 5520404 (Prime ONR) $400,000. April 2009-April 2012.

Real-Time Network Forensic Analysis. DARPA subcontract under prime contract number FA8750-07-C-0014 (PI). $79,329. January-June 2007.

Well-typed, Trustworthy Computing in the Presence of Transient Faults. NSF award CNS-0627650 (PI). $1,100,000. August 2006 - August 2010.

Language Support for Data-centric Systems Monitoring. NSF award CNS 0615062 (PI). $798,975. July 2006 - July 2009.

Automatic Tool Generation for Ad Hoc Scientific Data. NSF award IIS 0612147 (PI). $493,124. July 2006 - July 2009.

Collaborative Research: CSR-PDOS: Managing OS Extensibilty via Aspect-oriented Programming Technology. NSF award CSR 0615213 (Co-PI). $315,319. July 2006 - July 2009.

Emerson Junior Faculty Award for Excellence in Research and Teaching. $30,000. May 2005.

Alfred P. Sloan Fellow. $40,000. Sept 2004-Sept 2006.

Assurance-Carrying Components. ARDA grant for BAA 03-03-FH, Co-PI. $759,910. Oct 2003-March 2005.

CAREER: Programming Languages for Secure and Reliable Component Software Systems. NSF Career Grant, CCR-0238328, PI. $437,700. July 2003-June 2008.

Collaborative Research: High-Assurance Common Language Runtime. NSF CCR-0208601, Co-PI. $1,114,468.00. July 2002-June 2007.

A Gift from Microsoft Research. $25,000. February 2002.

Scaling Proof-Carrying Code to Production Compilers and Security Policies. DARPA contract F30602-99-1-0519, Co-PI. December 2002-December 2003.

Efficient Logics For Network Security. ONR. Senior personnel (Carnegie Mellon University). February 2001-January 2002.