

George Necula

WWW: <http://www.cs.berkeley.edu/~necula>

Email: gcneacula@gmail.com

Research Interests

Testing, for distributed systems, client-server web applications, and graphical user interfaces; programming language techniques for security; programming language theory, design and implementation; compilers, program analysis and optimization; formal methods and its applications for program verification, security and software engineering; compiler verification; mobile-code security; Proof-Carrying Code and its applications.

Employment

2016-present **YouTube/Google Inc.**

- Senior Staff Software Engineer. Working on YouTube data processing infrastructure.

1998-2017 (retired) **University of California, Berkeley** **Berkeley, CA**

- Professor of Computer Science (2008-2017). Vice-chair of the Electrical Engineering and Computer Science Department (2013-2015).
- Associate Professor of Computer Science (2004-2008)
- Assistant Professor of Computer Science (1998-2004)

2007-2015 **Conviva Inc.**

- Principal Engineer (full time 2007-2009, part-time since 2009). Designed and built client software for browsers and mobile devices, put in place software engineering processes, designed and built software testing infrastructure for cloud-based distributed testing of backend and client components, built several domain-specific language tools to improve development and maintenance of software.

Summer 1994 **AT&T Bell Laboratories** **Murray Hill, NJ**

- Summer intern

Summer 1995 **Digital Equipment Corp., SRC** **Palo Alto, CA**

- Summer intern

1992-1993 **Sviluppo Software di Base** **Torino, Italy**

- Software Engineer.

Education

1993-1998 **Carnegie Mellon University** **Pittsburgh, PA**

- MS in Computer Science, Spring 1995
- Ph.D. in Computer Science, October 1998
- Advisor: [Peter Lee](#). Thesis: “Compiling with Proofs”

1987-1992 University Politehnica Bucharest, Romania

- BS in Computer Science, Spring 1992

Honors and Awards

- ACM Distinguished Scientist starting from 2016
- 2012 Award for the Most Influential 2002 paper in the ACM Symposium on Principles of Programming Languages (POPL)
- 2007 Award for the Most Influential 1997 paper in the ACM Symposium on Principles of Programming Languages (POPL)
- 2006 ACM SIGOPS Hall of Fame Award, for the paper “Safe Kernel Extensions without Run-Time Checking”, presented at OSDI 1996.
- European Association for Programming Languages and Systems Paper Award 2005, for “Data Slicing: Separating the Heap into Independent Regions”.
- Alfred P. Sloan Foundation Fellow, Computer Science, 2003.
- ACM Grace Murray Hopper Award, 2001.
- Okawa Foundation Fellow, 1999.
- Nominee for ACM Best Dissertation Award, Carnegie Mellon University, 1999.
- NSF Faculty Early Career Development Award, 1998.
- Graduate Student Teaching Award, Carnegie Mellon University, 1995.
- Best Paper Award at Operating System Design and Principles, 1996
- Allen Newel Award for Research Excellence, Carnegie Mellon University, 1997.
- Gold medal at the International Physics Olympiad (High school level), Jena, Germany, 1987
- Gold medal at the International Physics Olympiad (High school level), London, 1986

Publications

Book chapters

1. George C. Necula. “*Proof-Carrying Code*”. In *Advanced Topics in Types and Programming Languages*, Benjamin Pierce, editor. MIT Press, 2004.
2. George C. Necula. “*Proof-Carrying Code*”. In *Encyclopedia of Cryptography and Security*, Henk C.A. van Tilborg, Sushil Jajodia, editors, Springer-Verlag, 2011.

Refereed Journal Publications

3. George C. Necula, Jeremy Condit, Matthew Harren, Scott McPeak, Westley Weimer. “*CCured: Type-Safe Retrofitting of Legacy Software*”, In *ACM Transactions on Programming Languages and Systems (TOPLAS)*, Volume 27, Number 3, pp. 477-526, 2005.

4. Sumit Gulwani, George C. Necula. “*A Randomized Satisfiability Procedure for Arithmetic and Uninterpreted Functions*”. In Journal of Information and Computation, Volume 199, Issues 1-2, pp. 107-131, 2005
5. Sumit Gulwani, George C. Necula. “*A Polynomial-Time Algorithm for Global Value Numbering*”. In Journal of Science of Computer Programming, Volume 64, issue 1, pp. 94-114, 2007.
6. A. Chander, D. Espinosa, N. Islam, P. Lee, G. Necula. “*Enforcing Resource Bounds via Static Verification of Dynamic Checks*”. In ACM Transactions on Programming Languages and Systems (TOPLAS), Volume 29, No 5, pp. 28:1-18, August 2007.
7. Wes Weimer, George C. Necula. “*Exceptional Situations and Program Reliability*”. In ACM Transactions on Programming Languages and Systems, Volume 30, No 2, pp 8:1-51, March 2008.

Refereed Conference Publications

8. George C. Necula and Peter Lee. “*Safe Kernel Extensions without Run-Time Checking*”. In the *Proceedings of the Second Symposium on Operating System Design and Implementation (OSDI96)*, Seattle, October 1996, pp. 229-243. Best Paper Award and also the 2006 ACM SIGOPS Hall of Fame Award.
9. George C. Necula. “*Proof-Carrying Code*”. In the *Proceedings of the 24th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages (POPL’97)*, Paris, January 1997, pp. 106-119. This paper received the 2007 award for the Most Influential POPL 1997 paper.
10. George C. Necula and Peter Lee. “*The Design and Implementation of a Certifying Compiler*”. In the *Proceedings of the 1998 SIGPLAN Conference on Programming Language Design and Implementation (PLDI’98)*, Montreal, June 1998.
11. George C. Necula and Peter Lee. “*Efficient Representation and Validation of Proofs*”. In the *Proceedings of 1998 IEEE Symposium on Logic in Computer Science (LICS)*, Indianapolis, 1998.
12. George C. Necula and Peter Lee. “*Proof Generation in the Touchstone Theorem Prover*”. In the *Proceedings of the 17th International Conference on Automated Deduction (CADE)*, Pittsburgh, 2000.
13. George C. Necula. “*Translation Validation for an Optimizing Compiler*”. In the *Proceedings of the 2000 ACM SIGPLAN Conference on Programming Language Design and Implementation (PLDI)*, Vancouver, 2000.
14. C. Colby, P. Lee, G. Necula, F. Blau, M. Plesko. “*A Certifying Compiler for Java*”. In the *Proceedings of the 2000 ACM SIGPLAN Conference on Programming Language Design and Implementation (PLDI)*, Vancouver, 2000.
15. C. Colby, P. Lee, G. Necula. “*A Proof-Carrying Code Architecture for Java*”. In *Proceedings of the Computer Aided Verification (CAV)*, 2000.
16. George C. Necula and Shree P. Rahul. “*Oracle-Based Checking of Untrusted Software*”. In *Proceedings of the 28th ACM Symposium on Principles of Programming Languages (POPL)*, 2001.
17. George C. Necula, Scott McPeak, Westley Weimer: “*CCured: Type Safe Retrofitting of Legacy Code*”, *Proceedings of the ACM Symposium on*

- Principles of Programming Languages (POPL), 2002. This paper received the 2012 award for the Most Influential POPL 2002 Paper.
18. George C. Necula, Scott McPeak, S. P. Rahul, Westley Weimer: “*CIL: Intermediate Language and Tools for the Analysis and Transformation of C Programs*”, Proceedings of the Conference on Compiler Construction (CC), 2002.
 19. Robert Schneck, George C. Necula. “*A Gradual Approach to a Trustworthy, yet Scalable Proof-Carrying Code Infrastructure*”, Proceedings of Conference on Automated Deduction (CADE), 2002, pp.47-62.
 20. Tom A. Henzinger, Ranjit Jhala, Rupak Majumdar, George C. Necula, Gregoire Sutre, Westley Weimer: “*Temporal Safety Proofs for Systems Code*”, Proceedings of Conference on Automated Verification (CAV), 2002.
 21. George C. Necula, Robert Schneck: “*Proof-Carrying Code with Untrusted Proof Rules*”. In Proceedings of the International Software Security Symposium (ISSS), 2002, pp. 283-298.
 22. Sumit Gulwani, George C. Necula: “*Discovering Affine Equalities Using Random Interpretation*”. In Proceedings of the 30th ACM SIGPLAN Symposium on Principles of Programming Languages (POPL), 2003, pp. 74-84.
 23. Jeremy Condit, Matthew Harren, Scott McPeak, George C. Necula, Westley Weimer: “*CCured in the Real World*”, Proceedings of the ACM SIGPLAN 2003 Conference on Programming Language Design and Implementation (PLDI), 2003, pp. 232-244.
 24. Sumit Gulwani, George C. Necula: “*A Randomized Satisfiability Procedure for Arithmetic and Uninterpreted Function Symbols*”, Proceedings of the Conference on Automated Deduction (CADE), 2003.
 25. George C. Necula, Robert Schneck: “*A Sound Framework for Untrusted Verification-Condition Generators*”, Proceedings of the IEEE Symposium on Logic in Computer Science (LICS), pp. 248-260, 2003.
 26. Rob von Behren, Jeremy Condit, Feng Zhou, George C. Necula, and Eric Brewer: “*Capriccio: Scalable Threads for Internet Services*”. In Proceedings of the 19th ACM Symposium on Operating System Principles (SOSP), pp. 268-281, 2003.
 27. Matthew Harren, George C. Necula: “*Lightweight Wrappers for Interfacing with Binary Code in CCured*”. In Proceedings of the International Software Security Symposium (ISSS), 2003. pp. 209-225.
 28. Sumit Gulwani, George C. Necula: “*Global Value Numbering Using Random Interpretation*”, In Proceedings of the 31th ACM SIGPLAN Symposium on Principles of Programming Languages (POPL), 2004, pp. 342-352.
 29. Scott McPeak, George C. Necula: “*Elkhound: A Fast, Practical GLR Parser Generator*”, In Proceedings of the International Conference on Compiler Construction (CC), 2004.
 30. Nathan Whitehead, Martin Abadi, George C. Necula: “*By Reason and Authority: A System for Authorization of Proof-Carrying Code*”. In Proceedings of the 17th IEEE Computer Security Foundations Workshop (CSFW), 2004, pp. 236-250.

31. Sumit Gulwani, George C. Necula: “*Free Conditional Expression Diagrams*”, In Proceedings of the Static Analysis Symposium (SAS), pp. 328-344, 2004.
32. Sumit Gulwani, George C. Necula: “*A Polynomial-Time Algorithm for Global Value Numbering*”, In Proceedings of the Static Analysis Symposium (SAS), pp. 212-228, 2004.
33. Westley Weimer, George C. Necula: “*Finding and Preventing Run-Time Error Handling Mistakes*”, In Proceedings of the Conference on Object-Oriented Programming Systems, Languages and Applications (OOPSLA), pp. 419-431, 2004.
34. Sumit Gulwani, Ashish Tiwari, George C. Necula: “*Join Algorithms for the Theory of Uninterpreted Functions*”. In Proceedings of Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS), Vol 3328 of LNCS, p.311-323, 2004.
35. Bor-Yuh Evan Chang, Adam Chlipala, George C. Necula, Robert R. Schneck: “*Type-Based Verification of Assembly Language for Compiler Debugging*”. In Proceedings of ACM SIGPLAN Workshop on Types in Language Design and Implementation (TLDI’05), pp. 1-12, 2005.
36. Bor-Yuh Evan Chang, Adam Chlipala, George C. Necula, Robert R. Schneck: “*The Open Verifier Framework for Foundational Verifiers*”. In Proceedings of ACM SIGPLAN Workshop on Types in Language Design and Implementation (TLDI’05), pp. 91-102, 2005.
37. Sumit Gulwani, George C. Necula. “*Precise Interprocedural Analysis using Random Interpretation*”. In Proceedings of ACM SIGPLAN Symposium on Principles of Programming Languages (POPL), pp. 324-337, 2005.
38. Jeremy Condit, George C. Necula. “*Data Slicing: Separating the Heap into Independent Regions*”. In Proceedings of the Conference on Compiler Construction (CC’05), pp. 172-187, 2005. Paper received European Association for Programming Languages and Systems Award.
39. Westley Weimer, George C. Necula. “*Mining Temporal Specifications for Error Detection*”. In Proceedings of the Conference on Tools and Applications for the Construction and Analysis of Systems (TACAS’05), pp. 461-476, 2005.
40. A. Chander, D. Espinosa, N. Islam, P. Lee, G. Necula: “*Enforcing Resource Bounds via Static Verification of Dynamic Checks*”. In Proceedings of the European Symposium on Programming (ESOP’05), 2005.
41. A. Chander, D. Espinosa, N. Islam, P. Lee, G. Necula: “*JVer: A Java Verifier*”, In Proceedings of the Conference on Computer Aided Verification (CAV’05), pp. 144-147, 2005.
42. Scott McPeak, George C. Necula. “*Data Structure Specification via Local Equality Axioms*”. In Proceedings of the Conference on Computer Aided Verification (CAV’05), pp. 476-490, 2005.
43. Matthew Harren, George C. Necula. “*Using Dependent Types to Certify the Safety of Assembly Code*”. In Proceedings of the Symposium on Static Analysis (SAS’05), pp. 155-170, 2005.
44. Bor-Yuh Evan Chang, Adam Chlipala, George C. Necula, “*A Framework for Certified Program Analyses and Its Applications to Mobile-Code Safety*”. In

- Proceedings of the 7th International Conference on Verification, Model Checking and Abstract Interpretation (VMCAI'06), pp 174-190, 2006.
45. Adam Chlipala, George C. Necula. “*Cooperative Integration of an Interactive Proof Assistant and an Automated Prover*”. In Proceedings 6th International Workshop on Strategies in Automated Deduction. August 2006.
 46. Bor-Yuh Evan Chang, Matthew Harren, George C. Necula. “*Analysis of Low-Level Code Using Cooperating Decompilers*”. In Proceedings of 13th International Static Analysis Symposium (SAS'06), pp. 318-335, 2006.
 47. Ulfar Erlingsson, Martin Abadi, Michael Vrbale, Mihai Budiu, George C. Necula. “*XFI: Software Guards for System Address Spaces*”. In Proceedings of Operating System Design and Implementation (OSDI'06), 2006.
 48. Feng Zhou, Jeremy Condit, Zachary Anderson, Ilya Bagrak, Rob Ennals, Matthew Harren, George Necula, Eric Brewer. “*SafeDrive: Safe and Recoverable Extensions Using Language-Based Techniques*”. In Proceedings of Operating System Design and Implementation (OSDI'06), pp. 45-60, 2006.
 49. Jeremy Condit, Matthew Harren, Zachary Anderson, David Gay, and George C. Necula. “*Dependent Types for Low-Level Programming*”. In Proceedings of European Symposium on Programming (ESOP'07), pp. 520-535, 2007.
 50. Zachary Anderson, Eric Brewer, Jeremy Condit, Rob Ennals, David Gay, Matthew Harren, George Necula, and Feng Zhou. “*Beyond Bug-Finding: Sound Program Analysis for Linux*”, HotOS 2007.
 51. Bor-Yuh Evan Chang, Xavier Rival, George Necula. “*Shape Analysis with Structural Invariant Checkers*”. In Proceedings of Static Analysis Symposium (SAS'07), pp. 384-400, 2007.
 52. Jacob Burnim, George Necula, Koushik Sen, “*Separating Functional and Parallel Correctness using Nondeterministic Sequential Specifications*”, 2nd Usenix Workshop on Hot Topics in Parallelism, HotPAR 2010, June 2010.
 53. Jacob Burnim, George Necula, Koushik Sen , “*Specifying and Checking Semantic Atomicity for Multithreaded Programs*”, 16th International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS), 2011, pp. 79-90.
 54. Jacob Burnim, Tayfun Elmas, George Necula, Koushik Sen , “*NDSeq: Runtime Checking for Nondeterministic Sequential Specifications of Parallel Correctness*”. Appeared in ACM SIGPLAN Symposium on Programming Language Design and Implementation (PLDI), 2011, pp. 401-414.
 55. Jacob Burnim, Tayfun Elmas, George Necula, Koushik Sen, “*NDetermin: Inferring Nondeterministic Sequential Specifications of Parallel Correctness (poster paper)*”, 17th ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming (PPoPP), February 2012, pp. 329-330.
 56. Jacob Burnim, Tayfun Elmas, George Necula, Koushik Sen. “*CONCURRIT: Testing Concurrent Programs with Programmable State-Space Exploration*”, 4th USENIX Workshop on Hot Topics in Parallelism (HotPar), 2012.
 57. Philip Reames, George Necula. “*Towards Hinted Collection: Annotations for Decreasing Garbage Collector Pause Times*”. In International Symposium on Memory Management (ISMM), June 2013.

58. Jacob Burnim, Tayfun Elmas, George Necula, Koushik Sen, “*CONCURRIT: A Domain Specific Language for Reproducing Concurrency Bugs*”, In Proc. ACM SIGPLAN Conference on Programming Language Design and Implementation (PLDI’13), June 2013.
59. Wontae Choi, George C. Necula, Koushik Sen.”*Guided GUI Testing of Android Apps with Minimal Restart and Approximate Learning*”, In ACM Conference on Object-Oriented Programming, Systems, Languages & Applications (OOPSLA’2013), October 2013.
60. Michael Pradel, Parker Schuh, George Necula, Koushik Sen. “*EventBreak: Analyzing the Responsiveness of User Interfaces through Performance-Guided Test Generation.*” In ACM Conference on Object-Oriented Programming, Systems, Languages & Applications (OOPSLA’2014), October 2014.
61. Koushik Sen, George Necula, Liang Gong, Wontae Choi. “*MultiSE: Multi-Path Symbolic Execution using Value Summaries*”. In ACM SIGSOFT Symposium on the Foundations of Software Engineering (FSE’15), September 2015. Winner of an *ACM SIGSOFT Distinguished Paper Award*.
62. Wontae Choi, Satish Chandra, George Necula, Koushik Sen. “*SJS: A Type System for JavaScript with Fixed Object Layout*”. In International Static Analysis Symposium (SAS’15), September 2015.
63. Colin Scott, Aurojit Panda, Vjekoslav Brajkovic, George Necula, Arvind Krishnamurthy, Scott Shenker. “*Minimizing Faulty Executions of Distributed Systems*”. In 13th USENIX Symposium on Networked Systems Design and Implementation (NSDI ’16), March 2016.
64. Wontae Choi, Koushik Sen, George Necula, Wenyu Wang. “*DetReduce: Minimizing Android GUI Test Suites for Regression Testing*”. In 40th International Conference on Software Engineering (ICSE’18), IEEE, 2018.

Invited Papers and Non-Refereed Publications

65. George C. Necula and Peter Lee. “*Research on Proof-Carrying Code for Untrusted-Code Security*”. In Proceeding of the 1997 IEEE Symposium on Security and Privacy, Oakland, 1997.
66. George C. Necula and Peter Lee. “*Research on Proof-Carrying Code on Mobile-Code Security*”. In Proceedings of the Workshop on Foundations of Mobile Code Security, Monterey, 1997.
67. George C. Necula and Peter Lee. “*Safe, Untrusted Agents Using Proof-Carrying Code*”. June 1998. In Lecture Notes in Computer Science 1419: Special Issue on Mobile Agent Security (Ed. Giovanni Vigna).
68. George C. Necula. “*Enforcing Safety and Security through Proof-Carrying Code*”. In Proceedings of the Fifteenth Conference on Mathematical Foundations of Program Semantics, New Orleans, 1999.
69. George C. Necula. “*A Logic-Based Approach to Software System Safety and Security*”. Invited paper to the Millenium Issue of SigSoft Software Engineering Notes, 2000.

70. George C. Necula. “*Proof-Carrying Code: Design, Implementation and Applications*”. In Proceedings of the 2nd International Conference on Principles and Practice of Declarative Programming (PPDP2000), Montreal, 2000.
71. George C. Necula. “*A Scalable Architecture for Proof-Carrying Code*”. Invited paper at the 5th International Symposium on Functional and Logic Programming, Tokyo, Lecture Notes in Computer Science 2024, 2001.
72. George C. Necula. “*Proof-Carrying Code. Design and Implementation.*” Book chapter in “H. Schwichtenberg, R. Steinbrüggen (eds.): *Proof and System-Reliability*. NATO Science Series III, Proc. International Summer School Marktoberdorf, July 24 to August 5, 2001. Amsterdam: Kluwer Academic Publishers 2002.
73. George C. Necula, Peter Lee: “*The Design and Implementation of a Certifying Compiler*”, a retrospective accompanying the reprint of the paper with the same name in “20 Years of ACM SIGPLAN Conference on Programming Language Design and Implementation (1979-1999): A Selection”, 2004.
74. J. Knoop, G. Necula, and W. Zimmermann, "Preface," Electronic Notes in Theoretical Computer Science: Proc. 5th Intl. COCV Workshop, vol. 176, no. 3, pp. 1-2, July 2007.
75. G. Necula and P. Wadler, Eds., Proceedings of the 35th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages (POPL 2008), New York, NY: The Association for Computing Machinery, Inc., 2008
- 76.

Technical Reports

77. George C. Necula and Lal George. “Accounting for the Performance of Standard ML on the DEC Alpha”, September 1994.
78. George C. Necula and Peter Lee. “*Proof-Carrying Code*”. Technical report CMU-CS-96-165, Carnegie Mellon University, November 1996.
79. George C. Necula and Peter Lee. “*Efficient Representation and Validation of Logical Proofs*”. Technical report, CMU-CS-97-172, Carnegie Mellon University, October 1997.
80. Peter Dinda and George Necula and Morgan Price. “*MacFS: A Portable Macintosh File System Library*”, Technical report, CMU-CS-98-145, Carnegie Mellon University, July 1998.
81. Shree P. Rahul and George C. Necula. “*Proof Optimization Using Lemma Extraction*”. UCB Technical Report CSD-01-1143, May 2001.
82. Sumit Gulwani, George C. Necula. “A Randomized Satisfiability Procedure for Arithmetic and Uninterpreted Function Symbols”. UCB Technical Report CSD-03-1241.
83. Sumit Gulwani, George C. Necula. “*Global Value Numbering using Random Interpretation*”. UCB Technical Report CSD-03-1296.
84. Sumit Gulwani, George C. Necula. “Path-sensitive Analysis for Linear Arithmetic and Uninterpreted Functions”. UCB Technical Report CSD-04-1325.

85. Sumit Gulwani, George C. Necula, "*Precise Interprocedural Analysis using Random Interpretation*". UCB Technical Report CSD-04-1353.
86. Bor-Yuh Evan Chang, Adam Chlipala, George C. Necula, "*A Framework for Certified Program Analyses and Its Applications to Mobile-Code Safety*". UCB Technical Report ERL M05/32.
87. Bor-Yuh Evan Chang, Matthew Harren, George C. Necula. "Analysis of Low-Level Code Using Cooperating Decompilers". Technical Report UCB/EECS-2006-86.
88. Jeremy Condit, Matthew Harren, Zachary Anderson, David Gay, George Necula. "Dependent Types for Low-Level Programming". Technical Report UCB/EECS-2006-129, 2006.
89. Raluca Sauciu, George Necula. "Reverse Execution With Constraint Solving". Technical report UCB/EECS-2011-67, EECS Department, University of California, Berkeley, May 2011.
90. Jacob Brunim, Tayfun Elmas, George Necula, Koushik Sen. "NDetermin: Inferring Nondeterministic Sequential Specifications for Parallelism Correctness". Technical report UCB/EECS-2011-143, December 2011.
91. Derrick Coetzee, Anand Bhaskar, George Necula. "A Model and Framework for Reliable Build Systems". Technical Report UCB/EECS-2012-27, February 2012.
92. Koushik Sen, George Necula, Liang Gong, Philip Wontae Choi. "MultiSE: Multi-Path Symbolic Execution using Value Summaries". Technical Report UCB/EECS-2014-173, October 2014.
93. Wontae Choi, Satish Chandra, George Necula, Koushik Sen. "SJS: A Type System for JavaScript with Fixed Object Layout". Technical Report UCB/EECS-2015-13, April 2015.

Lectures and Invited Talks

This section does not include the presentation of the conference papers listed above.

94. "Proof-Carrying Code. Design and Implementation". Presented at AT&T Bell Laboratories (December 1996), Massachusetts Institute of Technology (February 1997), Harvard University (February 1997), Digital Equipment Corp., Systems Research Center (March 1997), University of California at Berkeley (April 1997), SRI International (July 1997) and Stanford University (July 1997).
95. "Proof-Carrying Code". *The DARPA/NSA Operating System Security Workshops*, Washington, November 1996 and July 1997.
96. Programmability with Proof Carrying Code. Invited paper at the OpenSIG Workshop, Pittsburgh, 1999.
97. "Oracle-Based Checking of Software Safety". **Invited Talk** at the Principles of Declarative Programming Conference, Montreal 2000.
98. "Oracle-Based Checking of Assembly Language". **Invited lecture** part of the Distinguished Lecture Series at SUNY Stony Brook, December 2000.
99. "Proof-Carrying Code: Design, Implementation and Applications", Keio University, Tokyo (2001); Princeton University (2001), Indiana University

- (2001), Korean Advanced Science and Technology Institute (2001), Microsoft Research (2000), AT&T Research (2000), XEROX PARC (1999), Intel Corporation (1999).
100. “A Certifying Compiler for JAVA”. **Invited lecture** at the International Workshop on Advanced Compiler Technology for High Performance and Embedded Applications, Bucharest, Romania, July 2001.
 101. “Proof-Carrying Code. Design and Implementation”. Series of lectures at the Marktoberdorf Summer School, Germany, July 2001.
 102. “Theorem Proving Techniques for Software Engineering”, **Invited lecture** at the University of Washington/Microsoft Research Summer Institute: Specifying and Checking Properties of Software, 2001.
 103. “Random Interpretation of Programs”, **Invited talk** at the “New Directions in Software Technology Workshops”, December 2002.
 104. “CCured: Type-Safe Retrofitting of Legacy Software”. NTT Labs, San Jose, 2003, University of Tokyo, 2002, Tao Systems Inc., 2002, Meeting of the US-Japan Working Group on Logical Methods for Formal Verification of Software, 2001.
 105. “Randomized Algorithms for Program Analysis and Verification”, **Invited lecture** at the 10th International SPIN Workshop, May 2003.
 106. “Extensible Untrusted Code Verification”. École Normale Supérieure, Paris, 2003.
 107. “CCured: Type-Safe Retrofitting of Legacy Software”, Microsoft Research, September 2003, EADS Corporate Research Center, June 2005.
 108. “Translation Validation for Optimizing Compilers”. Fujitsu Labs, Santa Clara, 2003.
 109. “Extensible Verification of Untrusted Code”, Microsoft Research, January 2004, Cornell University, Computer Science Department Colloquium Series, November 2004; École Polytechnique de Lausanne (EPFL), Switzerland, June 2004.
 110. “Data Structure Specification via Local Equality Axioms”, Microsoft Cambridge Research Lab, April 2005; **Invited talk** at “London Theory Day”, April 2005; École Normale Supérieure, Paris, May 2005.
 111. “The Open Verifier Infrastructure for Building Untrusted Verifiers”, Ecole Normale Supérieure, Paris, February 2005; University of Turin, Italy, February 2005; **Invited talk** at “Construction and Analysis of Safe, Secure and Interoperable Smart devices” (CASSIS) Workshop, Nice, March 2005; Universidad Politecnica de Madrid, Spain, May 2005; Université de Bordeaux, France, May 2005; École Polytechnique, France, June 2005.
 112. “Randomized Algorithms for Program Analysis and Verification”, Université Paris 7, May 2005; INRIA Roquencourt France, June 2005; École Polytechnique, June 2005; **Invited talk** at Conference on Computer Aided Verification (CAV), July 2005;
 113. “Certifying Compilers”, at Dagstuhl Seminar 05311 “Verifying Optimizing Compilers”, August 2005.

114. “Using Dependent Types to Port Type Systems to Low-Level Languages”, **Invited talk** at Conference on Compiler Construction, Vienna, March 2006. Also at Ecole Normale Supérieure, and INRIA Rocquencourt, June 2006.
115. “Data Structure Specification via Local Equality Axioms”, **Invited talk** at “Formal Methods 2006”, Hamilton, August 2006

Panels

116. *Panelist*. The Security in Innovative New Operating Systems panel at the 1997 IEEE Symposium on Security and Privacy, Oakland, May 1997.
117. *Panelist*. The Future of Proof-Carrying Code panel. The Workshop on Proof-Carrying Code (Santa Barbara, 2000).
118. *Panelist*. Intel Research Forum on Language-Based Security (2002)

Software Packages

- “TexPoint”. A Visual Basic for Applications software package that allows the integration of mathematical formulas written in Latex into Microsoft Office presentations. (<http://texpoint.necula.org>)
- “CIL”. A software package for the development of program analysis and transformation tools for programs written in the C programming language. Used by 600 researchers. (<http://www.cs.berkeley.edu/~necula/cil>)
- “CCured”. A software package for eliminating memory safety security vulnerabilities from programs written in the C programming language. Used by 200 researchers.

Patents

- Issued:
 - “Safe to Execute Verification of Software”, U.S. Patent No. 6,128,774, issued 2000.
 - “Performing Checks on the Resource Usage of Computer Programs”, U.S. Patent No. 7,574,695, issued 2009.
 - “Method and Apparatus for Enforcing Safety Properties of Computer Programs by Generating And Solving Constraints”, U.S. Patent No. 7,730,455, issued 2010.
 - “Monitoring the Performance of a Content Player”, U.S. Patent No. 8,874,725, issued 2014.
 - “Augmenting the Functionality of a Content Player”, U.S. Patent No. 9,100,288, issued 2015.
- Pending applications:
 - “Dynamic Client Logging and Reporting”, submitted 2017 (15/690,016)

University Activities

Instruction

- Upper-division undergraduate courses:

- Programming Language Design and Implementation (CS164), Spring 2000, Spring 2001, Spring 2002, Spring 2003, Spring 2004, Spring 2010, Fall 2012.
- Software Engineering (CS169), Fall 2004, Fall 2006, Spring 2008, Fall 2009, Spring 2011, Fall 2011, Spring 2013.
- Graduate courses:
- Principles of Programming Language (CS263), 1999-2012.
- Techniques for Automated Deduction (CS294), Spring 2000, Fall 2004.
- Formal Techniques for Software Reliability, Spring 2001.

Advising

- PhD Students (with graduation year)
 - Sumit Gulwani (2005), Scott McPeak (2005), Westley Weimer (2005), Robert Schneck (2004), Jeremy Condit (2007), Matthew Harren (2007), Evan Chang (2008), Adam Chlipala (2008), Raluca Sauciu (2011), Benjamin Hindman (since 2007), Derrick Coetzee (since 2010), Philip Reames (2013), Philip Wontae Choi (2017), Patrick Li (2017).

Committees

- Lower Division Study Committee, 2003
- Graduate student admissions, 1999, 2000, 2001, 2010, 2011.
- Computing Infrastructure Departmental Committee, 2001, 2004.
- Recruiting Committee, 2007, 2010.
- Vice-chair of the department for Master Programs.

Professional Activities

Technical program committees

- Computer Security Foundations Workshop (CSFW, 1999).
- ACM SIGPLAN Symposium on Principles of Programming Languages (POPL 2000).
- ACM SIGPLAN Conference on Programming Language Design and Implementation (PLDI, 2001).
- International Workshop on Advanced Compiler Technology for High Performance and Embedded Systems (IWACT 2001).
- Workshop on Resource Management for Safe Languages (RMSF 2002)
- Workshop on Compiler Optimization Meets Code Verification (COCV 2002)
- Workshop on Practical Aspects of Declarative Languages (PADL 2003).
- International Symposium on Software Security (ISSS 2003)

- Workshop in Compiler Optimization Meets Compiler Verification (COCV2004), Program co-chair.
- Programming Language Design and Implementation (PLDI 2004)
- International Conference on Theoretical Computer Science (TCS2004)
- Workshop in Compiler Optimization Meets Compiler Verification (COCV2005), Program co-chair.
- IFIP Working Conference on Verified Software: Theory, Tools and Experiments (2005).
- Co-organized Dagstuhl Seminar 05311 “Verifying Optimizing Compilers”, August 2005.
- Conference on Logic in Computer Science (LICS 2006)
- IEEE 2nd International Conference on Intelligent Computer Communication and Processing (ICCP 2006)
- ACM Workshop on Typed in Language Design and Implementation, (TLDI 2007), program chair.
- ACM SIGPLAN Symposium on Principles of Programming Languages (POPL 2008), general chair.
- Proof Carrying Workshop (2008)
- European Symposium on Programming, (ESOP 2009).
- Static Analysis Symposium (2010, 2011)
- Conference on Compiler Construction (2013)

Journal Referee

Journal of Universal Computer Science (2003); ACM Transactions on Programming Languages and Systems (TOPLAS: 2001, 2002, 2004); Transactions on Computer Systems (2001); Journal of Automated Reasoning (2003); Higher-Order and Symbolic Computation (1999); IEEE Transactions on Software Engineering (1998)

Conference Referee

Functional Programming and Computer Architecture (FPCA: 1994); Symposium on Static Analysis (SAS:1996); International Conference on Functional Programming (ICFP: 1996,2000); Formal Methods Europe (FME: 1997, 2001); Workshop on Tools and Algorithms for the Analysis of Systems (1998); Workshop on Run-Time Result Verification (RTRV: 1999); Programming Language Design and Implementation (PLDI: 1999,2000,2003); Computer Security Foundations Workshop (CSFW: 1999); Symposium on Operating System Principles (SOSP: 1999, 2000, 2003); Principles of Programming Languages (POPL: 1999, 2000, 2001; 2003; 2004); Symposium on Operating System Design and Implementation (OSDI: 2002), Automated Software Engineering (ASE:

2002); International Symposium on Software Security (ISSS: 2002); Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS: 2002); International Conference on Logic Programming (ICLP: 2005); European Symposium on Programming (ESOP:2004); Conference on Logic in Computer Science (LICS: 2005).

Review panels and committees

- Member of the DARPA-sponsored ISAT Security and Mobility Study Group (2000)
- Council of Physical Sciences of the Netherlands Organization for Scientific Research (2000)
- Panelist for the evaluation of research center at the Korean Advanced Institute of Science and Technology (2001)
- US-Israel Binational Science Foundation panelist (2001)
- National Sciences and Engineering Research Council of Canada (2002)
- NSF review panelist (2001, 2002, 2005,2009)
- University of California, MICRO program reviewer (2003; 2005, 2006).
- German Israel Foundation for Research and Development –proposal reviewer, 2010.
- ACM SIGPLAN Symposium on Principles of Programming Languages (POPL), steering committee, 2008-2010.

Expert Witness Experience

- *Interwoven Inc., vs. Vertical Computer Systems*, Civil Case No. 3:10-cv-04645-RS, United States District Court, Northern District of California, San Francisco Division. Wrote patent invalidity and non-infringement expert reports. Deposition. Retained by White & Case LLP, on behalf of Interwoven Inc.
- *Finjan Inc. vs. Blue Coat Systems Inc.*, Case 13-cv-03999-BLF, United States District Court, Northern District of California, San Jose Division. Wrote patent invalidity expert report. Deposition. Trial testimony. Retained by Wilson Sonsini Goodrich & Rosati, on behalf of Blue Coat Systems.

Research Grants

1. National Science Foundation. “CAREER: A Logic-Based Approach to Software system Integrity and Security”, NSF Faculty Early Career Advancement Award, 3/1/99-2/28/04, \$205,000. PI.
2. National Science Foundation. “US-Japan Cooperative Science: Logical Methods for Formal Verification of Software”, 4/1/99-3/31/02, \$67,670, co-PI.
3. Okawa Foundation. Research Grant, \$10,000.

4. National Science Foundation. “ITR/SW: Translation Validation for Advanced Compiler Optimizations”, 9/1/00-8/31/05, \$499,555, PI.
5. National Science Foundation. “ITR/SW: The Open Source Quality Project”, 9/1/00-8/31/05, \$2,942,113, co-PI with Alex Aiken and Tom Henzinger.
6. DARPA: “Lightweight Software Specifications”, co-PI, with Alex Aiken, Tom Henzinger, David Schmidt. 2/08/01-1/31/04, \$1,133,045.
7. National Science Foundation and NASA: “Verifying Properties of Systems Software” Berkeley, 10/1/02-9/30/05, \$640,000, co-PI with Alex Aiken.
8. National Science Foundation. “Foundations of Hybrid and Embedded Software Systems”, senior participant, with 13 other faculty. 9/1/02-8/31/07, \$13,000,000.
9. California Micro program: “Tools for Securing the Software Infrastructure”, 8/1/03-12/1/04, \$108,000, 8/1/05-12/1/06, \$53,000, co-PI with Rastislav Bodik.
10. National Science Foundation: “CSR-PDOS: A New Foundation for the Evolution of Critical Systems”, co-PI, with Eric Brewer, 9/1/05-9/1/08, \$620,000.
11. California Micro program: “Lightweight Specifications for Safety in C Programs”, 8/1/06-12/1/07, \$40,000.
12. National Science Foundation: “CT-ISG: Collaborative Research: Open Software Security: Principles and Systems”, co-PI, with Martin Abadi, 9/1/05-9/1/10, \$300,000.
13. National Science Foundation: “CSR-SMA: Predictive Testing of System Software”, co-PI, with Koushik Sen, 9/2007-9/2010, \$350,000.
14. National Science Foundation: “SHF: Small: Programming Support for Checkpointing and Replay”, PI, 7/2010-7/2013, \$400,000.
15. National Science Foundation: “SHF: Small: Specifying and Verifying Essential Deterministic Behavior of Concurrent Programs”, co-PI, with Koushik Sen, 7/2010-7/2013, \$475,000.
16. National Science Foundation: “Automated Graphical User Interface Testing with Learning”, co-PI, with Koushik Sen, 8/2014-7/2017, \$850,000.