Panagiotis (Pete) Manolios

College of Computer and Information Science
360 Huntington Avenue
Northeastern University
Boston, Massachusetts 02115 U.S.A.

Work: +1-617-373-3694
Home: +1-781-806-0256
pete@ccs.neu.edu
http://www.ccs.neu.edu/~pete

Education

Ph.D. in Computer Sciences, The University of Texas at Austin, 8/2001.
M.A. in Computer Sciences, Brooklyn College, 6/1992.
B.S. in Computer Sciences, Brooklyn College, 9/1991.

Academic Employment

Professor. 7/2013–present.
College of Computer and Information Science, Northeastern.

Associate Professor. 8/2007–6/2013.
College of Computer and Information Science, Northeastern.

Assistant Professor. 8/2001–6/2008.
College of Computing, Georgia Institute of Technology.

Adjunct Professor. 8/2003–present.
School of Electrical and Computer Engineering, Georgia Institute of Technology.

Research and Teaching Assistant. 9/1995–8/2001.
Department of Computer Sciences, University of Texas at Austin.

Summer Research Staff Member. 5/1998–7/1998.
Digital Systems Research Center, Palo Alto.

Adjunct Faculty. 1/1992–6/1994.
Department of Computer and Information Science. Brooklyn College.

Industrial Employment

Consultant. 2008-Present.
Consulting for industry and government on safety-critical systems, verification, validation, real-time systems and synthesis. Worked with GE, Intel, Apogee, BBN, NASA, DARPA.

Summer Research Staff Member. 5/1998–7/1998.
Digital Systems Research Center, Palo Alto.

Research Interests

My main research interest is computer-aided modeling, verification, validation and analysis of systems. What guides my research is the vision that formal methods can be used to revolutionize the modeling, design, implementation and analysis of highly reliable, robust, secure, and scalable systems in a variety of important application areas, ranging from large-scale component-based software systems to hardware systems to aerospace systems to public health. To this end, I develop algorithms, methodologies, and concepts to automate the analysis of systems. I have also devoted much of my research to applications and to building and experimentally validating tools. My other areas of interest include distributed computing, logic, programming languages, software engineering, algorithms, computer architecture, aerospace, and pedagogy.

I. TEACHING

A. Courses Taught

Semester/Year	Course	Number of Students	Notes
Spring 2016	CS2800 Logic and Computation	87	
Spring 2016	7805 Theory of Computation	15	
Spring 2015	CS2800 Logic and Computation	114	
Spring 2015	7805 Theory of Computation	21	
Spring 2013	CS2800 Logic and Computation	90	
Spring 2013	Topics in Formal Methods	16	
Spring 2012	CS2800 Logic and Computation	80	
Fall 2011	CS2800 Logic and Computation	35	
Spring 2011	CS2800 Logic and Computation	68	
Fall 2010	CS 7480 Computational Logic	12	
Spring 2010	CS 2800 Logic and Computation	97	Two sections
Fall 2009	CS 2800 Logic and Computation	22	
Spring 2009	CSU 290 Logic and Computation	52	
Fall 2008	CSG 369 Special Topics in PL: Formal Methods	7	
Spring 2008	CSU 290 Logic and Computation	40	New course
Fall 2007	CSG 379 Decision Procedures for Verification	11	New course
Spring 2007	CS 3510 Algorithms	56	
Spring 2007	CS 8803 Computational Logic	7	
Fall 2006	CS 8801 SPARC Seminar	17	
Spring 2006	CS 8803 Computational Logic	13	
Spring 2005	CS 8803 Computational Logic	11	Redesigned Course
Spring 2005	CS 3510 Algorithms	59	
Fall 2004	CS 8801 Hardware Verification	7	New; cross-listed ECE 8801
Spring 2004	CS 3500 Theory I	60	
Spring 2004	CS 8803 Formal Methods	18	Cross-listed ECE 8823 B
Spring 2003	CS 8803 Formal Methods	10	
Fall 2002	CS 3220 Processor Design	34	Redesigned Course
Spring 2002	CS 8803 Formal Modeling and Analysis of Computing Systems	12	New Course
Summer 1994	Introduction to Mathematical Reasoning and Computer Science	19	
Spring 1994	Introduction to Mathematical Reasoning and Computer Science	35	
Spring 1994	Computer Organization	27	
Fall 1993	Computer Organization	14	
Fall 1993	Introduction to Computing Using Pascal	38	
Summer 1993	Computer Organization	12	
Summer 1993	Microcomputer Business Applications	23	
Spring 1993	Computer Organization	26	
Spring 1993	Introduction to Mathematical Reasoning and Computer Science	37	
Fall 1992	Data Structures	26	
Fall 1992	Introduction to Mathematical Reasoning and Computer Science	40	

B. Curriculum Development
B1. Northeastern University

CSG 369: Special Topics in Programming Languages: Formal Methods. This is an advanced Ph.D.-level course on the current state-of-the-art in a few selected topics in formal methods. The topics are: termination analysis, satisfiability modulo theories, theorem proving, and refinement. The major goal is the completion of a project that extends the state-of-the-art and leads to publishable results.

CSU 290: Logic and Computation. The goal of this undergraduate freshman class is introduce formal logic to freshmen by showing them how to formally and mechanically reason about their programs, using both paper and pencil and the ACL2 Sedan theorem-proving system. By the end of the semester most students are able to formally prove theorems about programs that required the construction of appropriate generalizations and non-trivial inductive proofs.

CSG 379: Decision Procedures for Verification. This is a graduate course on decision procedures for logics ranging from propositional logic to temporal logic to logics that include arithmetic, uninterpreted functions, equality, and arrays. We also discuss abstraction, refinement, automated theorem proving, and techniques for combining decision procedures.

B2. Georgia Institute of Technology

CS 4560: Verification of Systems. This undergraduate course was developed with Alex Orso for the Georgia Tech threads program in Fall 2005. The course provides an introduction to the basic methods and tools for verifying and validating computing systems. Students acquire a clear understanding of the core issues in verification and validation and learn how to prioritize, conduct, and evaluate verification and validation efforts. Topics covered include: modeling of systems, model checking, decision procedures, theorem proving, testing, runtime verification, evaluation, and certification.

CS 6382: Computational Logic. This graduate course, offered starting Spring 2005, is a refinement of the Formal Modeling and Analysis of Computing Systems course. The class is a breadth course in Software Engineering and Information Security. Much of the material is part of the Software Engineering and Programming Language qualifiers. This class provides a graduate-level introduction to the fundamental ideas in modern logic that underlie computing. Topics covered include first-order logic, Gödel's completeness and incompleteness theorems, decision problems in first-order logic, mechanical verification, modeling computing systems, formal semantics of programming languages, decision procedures, temporal logic, model checking, abstraction, and refinement.

CS 8801: Hardware Verification. New graduate seminar (Spring 2004) on current research on modeling and formally verifying hardware. Half of the semester is devoted to introducing the topic, including: an introduction to the ACL2 theorem-proving system, which we use to model DLX-like machines; an introduction to correctness criteria focusing on recent work on refinement; the use of ACL2 to prove the DLX machine correct; and a discussion of decision procedures, including SAT-solvers and solvers for fragments of first-order logic with uninterpreted functions. The rest of the semester consists of student presentations on current research topics based on student interest.

CS 8803: Formal Modeling and Analysis of Computing Systems. New graduate course (Spring 2002) on the fundamental techniques for modeling and formally analyzing computing systems, with a focus on applications in software, hardware, and security. Students learn the fundamentals of classical logic, induction and recursion, program semantics, rewriting, reactive systems, temporal logic, model checking, and abstraction. We examine how these methods can be used to verify

software, hardware, and security protocols. Students learn how to use various tools, including theorem-proving and model-checking tools, and work in groups to apply the tools to various domains. We discuss the limitations of current techniques and systems and we examine promising research directions including building more useful systems and developing more powerful techniques.

CS 3220: Processor Design. This is a junior-level undergraduate course where students use FPGAs to design and build a microprocessor at the bit level. We distribute the FPGAs and associated software to all the students for the duration of the semester. This allows us to assign in-depth projects that provide invaluable engineering experience and insight that is difficult to impart without the constant availability of the FPGAs. The class also provides an in-depth treatment of verification.

C. Individual Student Guidance

1. Research Scientists Supervised

Eugene Goldberg: July 2009-July 2013. **Research:** B.44, B.41, B.36, B.34, F.4; **Grants:** J.23, J.22, J.21, J.20. Eugene was promoted to Research Assistant Professor in July of 2013 and left the university in 2015.

Carlos Pacheco: October 2001-August 2002. Later joined MIT and obtained a Ph.D.

2. Ph.D. Students Supervised

Mitesh Jain: Fall 2011-Present. **Research:** B.50, B.49, G.18

Harsh Raju Chamarthi: Spring 2009-Present (joined Ph.D. program Fall 2010.) **Research:** B40, B.37, G.18, G.16, I.5

Vasilis Papavasiliou: Fall 2010-Summer 2015. Vasili successfully defended on June 2015.
Research: B.51, B.48, B.43, B.39, B.38, G.15, I.6 **Notes:** Vasilis accepted a position as a postdoctoral researcher at Université Paris Diderot (Paris 7), Laboratoire Preuves, Programmes, et Systèmes.

Peter Dillinger: Fall 2003-December 2010. Peter successfully defended on December 2010.
Research: B.37, B.26, B.12, G.16, G.14, G.12, G.10, G.8, G.5, I.5, I.3, and I.2
Notes: Peter was accorded an honorable mention for his NSF Graduate Research Fellowship application. Peter accepted a position at Coverity Inc.

Sudarshan Kumar Srinivasan: Fall 2003-August 2007, Sudarshan successfully defended on August 10, 2007.
Research: B.29, B.24, B.19, B.18, B.17, B.15, B.14, B.13, B.10, E.9, E.8, E.5, F.3, G.7, H.3, and I.7
Notes: Sudarshan is a tenured associate professor at North Dakota State University.

Daron Vroon: January 2002-August 2007. Daron successfully defended on August 7, 2007.
Research: B.37, B.35, B.33, B.29, B.28, B.27, B.26, B.24, B.22, B.20, B.11, B.6, E.6, E.4, G.11, G.10, G.4, H.4, I.7, I.6, I.5, and I.1.
Notes: Daron was the recipient of an NSF Graduate Research Fellowship.

3. Other Ph.D. Special Problems Students

Jorge Pais (Fall 2013-Spring 2015, **Research:** B.51), Ben Chambers (Spring 2008-Spring 2009, **Research:** B.33), Jed Davis (Spring 2008-Fall 2008), Christine Hang (Spring 2008-Summer 2008, **Research:** B.38), Aaron Turon (Spring 2008-Summer 2008, **Research:** B.32), Yimin Zhang (Fall 2006-Fall 2007, **Research:** B.21), Viswanath Nagarajan (Fall 2003), Christoph Csallner (Fall 2003), Shan Shan Huang (Fall 2003), Kemin Yang (Spring & Summer 2003), David Dagon (Spring 2003), Tom Bankston (Fall 2002), and Jim Bowring (Spring 2002).

4. Masters Students Supervised

Roma Kane: Spring 2005-Present, Graduated with a masters degree in Spring 2007.
Research: B.19.

Gayatri Subramanian: Spring 2005-Spring 2006, Graduated Spring 2006.
Research: B.28 and I.6.
Notes: Graduated with a Master's thesis under my supervision, entitled "Automating Component-Based System Assembly."

5. Other M.S. Special Problems Students

Christopher Church (Fall 2004), Ivan Raikov (Fall 2004)

6. Undergraduate Special Problems Students

Numan Salati (Summer 2003), Lazy Rewriting. Phu C. Le (Fall 2004) Processor Verification. Daniel E. Vogh (Fall 2005) ACL2 and compiler verification. Denis Bueno (Spring 2006-Spring 2007) Expressive Package Management.

7. Visiting Special Problems Students

Marc Galceran Oms [see B.25] (Spring-Summer 2006) and Sergi Oliva Valls [see B.25] (Spring 2006) Using SAT solving techniques for computational biology.

II. RESEARCH AND CREATIVE SCHOLARSHIP

A. Dissertation

1. Panagiotis Manolios. Mechanical Verification of Reactive Systems. The University of Texas at Austin, Department of Computer Sciences, Austin, TX, August 2001.

B. Conference Papers

51. Panagiotis Manolios, Jorge Pais and Vasilis Papavasileiou. The Inez Mathematical Programming Modulo Theories Framework. *CAV, Computer-Aided Verification*. July 2015.

50. Mitesh Jain and Panagiotis Manolios: Skipping Refinement. *CAV, Computer-Aided Verification*. July 2015.

49. Greg Eakman, Howard Reubenstein, Tom Hawkins, Mitesh Jain and Panagiotis Manolios. Practical Formal Verification of Domain-Specific Language Applications. *NFM, NASA Formal Methods*. 2015.

48. Panagiotis Manolios, Vasilis Papavasileiou, and Mirek Riedewald. ILP Modulo Data. *FMCAD, Formal Methods in Computer-Aided Design (FMCAD 2014)* October 2014.

47. Justin Slepak, Olin Shivers, and Panagiotis Manolios. An Array-Oriented Language with Static Rank Polymorphism. *ESOP, European Symposium on Programming (ESOP 2014)* April 2014. (Best paper award)

46. Eugene Goldberg and Panagiotis Manolios. Partial Quantifier Elimination. *Haifa Verification Conference* 2014.

45. Eugene Goldberg and Panagiotis Manolios. Software for Quantifier Elimination in Propositional Logic. ICMS, 2014.

44. Eugene Goldberg and Panagiotis Manolios. Quantifier Elimination via Clause Redundancy. *FMCAD, Formal Methods in Computer-Aided Design (FMCAD 2013)* October 2013.

43. Panagiotis Manolios and Vasilis Papavasileiou. ILP Modulo Theories. *CAV, Computer-Aided Verification (CAV 2013)* July 2013.

42. Panagiotis Manolios. Counterexample Generation Meets Interactive Theorem Proving: Current Results and Future Opportunities. *Invited Tutorial, ITP, Interactive Theorem Proving (ITP 2013)* July 2013.

41. Eugene Goldberg and Panagiotis Manolios. Quantifier Elimination by Dependency Sequents. *FMCAD, Formal Methods in Computer-Aided Design (FMCAD 2012)*, October 2012.

40. Harsh Chamarthi and Panagiotis Manolios. Automated Specification Analysis Using an Interactive Theorem Prover. *FMCAD, Formal Methods in Computer-Aided Design (FMCAD 2011)*, October 2011.

39. Panagiotis Manolios and Vasilis Papavasileiou. Pseudo-Boolean Solving by Incremental Translation to SAT. *FMCAD, Formal Methods in Computer-Aided Design (FMCAD 2011)*, October 2011.

38. Christine Hang, Panagiotis Manolios, and Vasilis Papavasileiou. Synthesizing Cyber-Physical Architectural Models with Real-Time Constraints. *CAV 2011, Twenty Third International Conference on Computer Aided Verification*, July 2011.

37. Harsh Chamarthi, Peter Dillinger, Panagiotis Manolios, Daron Vroon. The ACL2 Sedan Theorem Proving System. *TACAS, International, 2011 Conference on Tools and Algorithms for the Construction and Analysis of Systems*. Springer, March 2011.

36. Eugene Goldberg and Panagiotis Manolios. SAT-Solving Based on Boundary Point Elimination. *HVC 2010, Haifa Verification Conference*. Springer, Oct. 2010.

35. Panagiotis Manolios and Daron Vroon. Interactive Termination Proofs Using Termination Cores. *ITP 2010, Interactive Theorem Proving* July, 2010.

34. Eugene Goldberg and Panagiotis Manolios. Generating High-Quality Tests for Boolean Circuits by Treating Tests as Proof Encoding. *TAP 2010, Tests and Proofs*, Springer, July 2010.

33. Benjamin Chambers, Panagiotis Manolios, and Daron Vroon. Faster SAT Solving with Better CNF Generation. *DATE 2009, Design Automation and Test in Europe*, ACM, April 2009.

32. Panagiotis Manolios and Aaron Turon. All-Termination(T). *TACAS 2009, International Conference on Tools and Algorithms for the Construction and Analysis of Systems*. Springer, March 2009.

31. Matthew Might and Panagiotis Manolios. A posteriori soundness for non-deterministic abstract interpretations. *VMCAI 2009, Tenth International Conference on Verification, Model Checking, and Abstract Interpretation*. Springer, January 2009.

30. Panagiotis Manolios. Automating the Assembly of Avionics Systems. *AIAA Guidance, Navigation, and Control Conference*. August 2008. (Invited)

29. Panagiotis Manolios, Sudarshan Srinivasan, and Daron Vroon. BAT: The Bit-Level Analysis Tool. *CAV 2007, Nineteenth International Conference on Computer Aided Verification*. Springer, July 2007.

28. Panagiotis Manolios, Gayatri Subramanian, and Daron Vroon. Automating Component-Based System Assembly. *ISSTA 2007, International Symposium on Software Testing and Analysis*. ACM, July 2007.

27. Panagiotis Manolios and Daron Vroon. Efficient Circuit to CNF Conversion. *SAT 2007, The Tenth International Conference on Theory and Applications of Satisfiability Testing*. Springer, May 2007.

26. Peter Dillinger, Panagiotis Manolios, J S. Moore, Daron Vroon. ACL2s: "The ACL2 Sedan." *ICSE'07, The 29th International Conference on Software Engineering, Research Demonstration Track*. ACM, May 2007.

25. Panagiotis Manolios, Marc Galceran Oms, and Sergi Oliva Valls. Checking Pedigree Consistency with SAT. *TACAS 2007, International Conference on Tools and Algorithms for the Construction and Analysis of Systems*. Springer, March 2007.

24. Panagiotis Manolios, Sudarshan K. Srinivasan, and Daron Vroon. Automatic Memory Reductions for RTL-Level Verification. *ICCAD 2006, ACM-IEEE International Conference on Computer Aided Design*. IEEE Computer Society, November 2006.

23. William G.J. Halfond, Alessandro Orso, and Panagiotis Manolios. Using Positive Tainting and Syntax-Aware Evaluation to Counter SQL Injection Attacks. *The Twelfth ACM SIGSOFT Symposium on Foundations of Software (FSE'06)*. ACM, November 2006.

22. Panagiotis Manolios and Daron Vroon. Termination Analysis with Calling Context Graphs. *Computer-Aided Verification (CAV-2006)*, volume 4144 of *LNCS*, pages 401–414. Springer, August 2006.

21. Panagiotis Manolios and Yimin Zhang. Implementing Survey Propagation on Graphics Processing Units. *Ninth International Conference on Theory and Applications of Satisfiability Testing (SAT-2006)*, volume 4121 of *LNCS*, pages 311–324. Springer, August 2006.

20. Panagiotis Manolios and Daron Vroon. Integrating Static Analysis and General-Purpose Theorem Proving for Termination Analysis. *ICSE'06, The 28th International Conference on Software Engineering, Emerging Results*, pages 873–876. ACM, May 2006.

19. Roma Kane, Panagiotis Manolios, and Sudarshan K. Srinivasan. Monolithic Verification of Deep Pipelines with Collapsed Flushing. *DATE 2006, ACM-IEEE Design, Automation, and Test in Europe*, pages 1234-1239. European Design and Automation Association, March 2006.

18. Panagiotis Manolios and Sudarshan K. Srinivasan. Verification of Executable Pipelined Machines with Bit-Level Interfaces. *ICCAD 2005, ACM-IEEE International Conference on Computer Aided Design*, pages 855–862. IEEE Computer Society, November 2005.

17. Panagiotis Manolios and Sudarshan K. Srinivasan. A Complete Compositional Reasoning Framework for the Efficient Verification of Pipelined Machines. *ICCAD 2005, ACM-IEEE International Conference on Computer Aided Design*, pages 863–870. IEEE Computer Society, November 2005.

16. Panagiotis Manolios. The Challenge of Hardware-Software Co-Verification. *VSTTE, IFIP Working Conference on Verified Software: Theories, Tools, Experiments, Part of ETH's 150th anniversary celebration.* Volume 4171 of *LNCS*, pages 438–447. Springer, Zurich, October 2005.

15. Panagiotis Manolios and Sudarshan K. Srinivasan. A Parameterized Benchmark Suite of Hard Pipelined-Machine-Verification Problems. *Advanced Research Working Conference on Correct Hardware Design and Verification Methods (CHARME 2005)*, volume 3725 of *LNCS*, pages 363–366. Springer, October 2005.

14. Panagiotis Manolios and Sudarshan K. Srinivasan. A Computationally Efficient Method Based on Commitment Refinement Maps for Verifying Pipelined Machines. *ACM-IEEE International Conference on Formal Methods and Models for Codesign (MEMOCODE 2005)*, pages 188-197. IEEE, July 2005.

13. Panagiotis Manolios and Sudarshan K. Srinivasan. Refinement Maps for Efficient Verification of Processor Models. *DATE 2005, Design, Automation, and Test in Europe*, pages 1304–1309. IEEE Computer Society, March 2005.

12. Peter C. Dillinger and Panagiotis Manolios. Bloom Filters in Probabilistic Verification. *Formal Methods in Computer-Aided Design (FMCAD 2004)*, volume 3312 of *LNCS*, pages 367–381. Springer, November 2004.

11. Panagiotis Manolios and Daron Vroon. Integrating Reasoning about Ordinal Arithmetic into ACL2. *Formal Methods in Computer-Aided Design (FMCAD 2004)*, volume 3312 of *LNCS*, pages 82–97. Springer, November 2004.

10. Panagiotis Manolios and Sudarshan K. Srinivasan. Automatic Verification of Safety and Liveness for XScale-Like Processor Models Using WEB Refinements. *Design, Automation, and Test in Europe (DATE 2004)*, pages 168–175. IEEE Computer Society, February 2004.

9. Panagiotis Manolios. A Compositional Theory of Refinement for Branching Time. *CHARME 2003, the 12th Advanced Research Working Conference on Correct Hardware Design and Verification Methods*, volume 2860 of *LNCS*, pages 304–318. Springer, October 2003.

8. Panagiotis Manolios and Richard Trefler. A Lattice-Theoretic Approach to Safety and Liveness. *Twenty-Second ACM Symposium on Principles of Distributed Computing (PODC 2003)*, pages 325–333. ACM Press, July 2003.

7. Panagiotis Manolios. Branching Time Refinement. Brief Announcements, *Twenty-Second ACM Symposium on Principles of Distributed Computing (PODC 2003)*, pages 334–334. ACM Press, July 2003.

6. Panagiotis Manolios and Daron Vroon. Algorithms for Ordinal Arithmetic. *Nineteenth International Conference on Automated Deduction (CADE)*, volume 2741 of *LNCS*, pages 243–257. Springer, July 2003.

5. Panagiotis Manolios and Richard Trefler. Safety and liveness in branching time. In Joseph Halpern, editor, *Logic in Computer Science–LICS 2001*, pages 366–374. IEEE Computer Society, June 2001.

4. Panagiotis Manolios. Correctness of pipelined machines. In Warren A. Hunt, Jr. and Stephen D. Johnson, editors, *Formal Methods in Computer-Aided Design–FMCAD 2000*, volume 1954 of *LNCS*, pages 161–178. Springer, November 2000.

3. Panagiotis Manolios, Kedar Namjoshi, and Robert Sumners. Linking theorem proving and model-checking with well-founded bisimulation. In Nicolas Halbwachs and Doron Peled, editors, *Computer-Aided Verification–CAV '99*, volume 1633 of *LNCS*, pages 369–379. Springer, July 1999.

2. Yuan Yu, Panagiotis Manolios, and Leslie Lamport. Model checking TLA^+ specifications. In Laurence Pierre and Thomas Kropf, editors, *Correct Hardware Design and Verification Methods, CHARME '99*, volume 1703 of *LNCS*, pages 54–66. Springer, September 1999.

1. Don L. Scarborough, Panagiotis Manolios, and Jacqueline A. Jones. MusicSoar: Soar as an architecture for music cognition. In *Proceedings of the Fourteenth Annual Conference of the Cognitive Science Society*, pages 1104–1109, July 1992.

C. Edited Proceedings

2. Aarti Gupta and Panagiotis Manolios, editors. *Proceedings of the ACM/IEEE Formal Methods in Computer-Aided Design Conference.* IEEE, November 2006.

1. Panagiotis Manolios and Matthew Wilding, editors. *Proceedings of the Sixth International Workshop on the ACL2 Theorem Prover and its Applications.* ACM, August 2006.

D. Books

4. Matt Kaufmann, Panagiotis Manolios, and J Strother Moore. *Computer-Aided Reasoning: An Approach.* Paperback Edition (of D.2), August 2002.

   Note: this is the ACL2 textbook. ACL2 is part of the Boyer-Moore family of provers, winner of the 2005 ACM Software System Award.

3. Matt Kaufmann, Panagiotis Manolios, and J Strother Moore, editors. *Computer-Aided Reasoning: ACL2 Case Studies.* Paperback Edition (of D.1), August 2002.

2. Matt Kaufmann, Panagiotis Manolios, and J Strother Moore. *Computer-Aided Reasoning: An Approach.* Kluwer Academic Publishers, July 2000.

1. Matt Kaufmann, Panagiotis Manolios, and J Strother Moore, editors. *Computer-Aided Reasoning: ACL2 Case Studies.* Kluwer Academic Publishers, June 2000.

E. Journal Articles

11. Eugene Goldberg and Panagiotis Manolios. Quantifier elimination by dependency sequents. *Formal Methods in System Design*. volume 45(2), pages 111-143, August 2014.

10. Panagiotis Manolios and Richard Trefler. A Semantic Characterization of Safety and Liveness for Branching Time Logics, 41 pages. *To appear. Distributed Computing*.

9. Panagiotis Manolios and Sudarshan K. Srinivasan. Automatic Verification of Safety and Liveness for Pipelined Machines Using WEB Refinement. *ACM Transactions on Design Automation of Electronic Systems*, volume 13(3), article 45 (19 pages), 2008.

8. Panagiotis Manolios and Sudarshan K. Srinivasan. A Refinement-Based Compositional Reasoning Framework for Pipelined Machine Verification. *IEEE Transactions on VLSI Systems*, volume 16(4), pages 353–364, 2008.

7. William G.J. Halfond, Alessandro Orso, and Panagiotis Manolios. WASP: Detecting and Preventing SQL Injection Attacks Using Positive Tainting and Syntax-Aware Evaluation. *IEEE Transactions on Software Engineering, special issue on Software Engineering for Secure Systems*, volume 34(1), pages 65–81, 2008.

6. David A. Greve, Matt Kaufmann, Panagiotis Manolios, J S. Moore, Sandip Ray, Jose L. Ruiz-Reina, Rob Sumners, Daron Vroon, and Matthew Wilding. Efficient Execution in an Automated Reasoning Environment. *Journal of Functional Programming*, volume 18(1), pages 15–46, 2008.

5. Panagiotis Manolios and Sudarshan K. Srinivasan. A Framework for Verifying Bit-Level Pipelined Machines Based on Automated Deduction and Decision Procedures, 26 pages. *Journal of Automated Reasoning*, volume 37(1–2), pages 93–116, 2006.

4. Panagiotis Manolios and Daron Vroon. Ordinal Arithmetic: Algorithms and Mechanization. *Journal of Automated Reasoning*, volume 34(4), pages 387–423, 2005.

3. Panagiotis Manolios and J Strother Moore. Partial Functions in ACL2. *Journal of Automated Reasoning*, volume 31(2), pages 107–127, 2003.

2. Panagiotis Manolios and J Strother Moore. On the desirability of mechanizing calculational proofs. *Information Processing Letters*, 77(2–4): 173–179, January 2001.

1. Panagiotis Manolios and Robert Fanelli. First order recurrent neural networks and deterministic finite automata. *Neural Computation*, volume 6, pages 1155–1173, MIT Press, November 1994.

F. Book Chapters

4. Eugene Goldberg and Panagiotis Manolios. Boundary Points and Resolution. Chapter 7 in P. Khatri and Kanupriya Gulati, editors, *Advanced Techniques in Logic Synthesis, Optimizations and Applications*, pages 109–128, Springer, Dec. 2010.

3. Panagiotis Manolios and Sudarshan K. Srinivasan. Verifying Pipelines with BAT. In S. Hardin, editor, *Design and Verification of Microprocessor Systems for High-Assurance Applications*. Springer, March 2010.

2. Panagiotis Manolios. Refinement and Theorem Proving. *Formal Methods for Hardware Verification*, volume 3965 of *LNCS*, pages 176–210. Springer, May 2006.

1. Panagiotis Manolios. Mu-calculus model-checking. In Matt Kaufmann, Panagiotis Manolios, and J Strother Moore, editors, *Computer-Aided Reasoning: ACL2 Case Studies*, pages 93–111. Kluwer Academic Publishers, June 2000.

G. Refereed Workshop Papers

18. Mitesh Jain and Panagiotis Manolios: Proving Skipping Refinement with ACL2s. ACL2 2015.

17. Harsh Raju Chamarthi, Peter C. Dillinger, and Panagiotis Manolios. Data Definitions in the ACL2 Sedan. *ACL2 2014*, pages 27-48, 2014.

16. Harsh Chamarthi, Peter C. Dillinger, Matt Kaufmann, and Panagiotis Manolios. Integrating Testing and Interactive Theorem Proving. *Tenth International Workshop on the ACL2 Theorem Prover and Its Applications (ACL2 2011)*, 2011.

15. Panagiotis Manolios and Vasilis Papavasileiou. Virtual Integration of Cyber-Physical Systems by Verification. *Analytic Virtual Integration of Cyber-Physical Systems Workshop*, November 2010.

14. Peter C. Dillinger and Panagiotis Manolios. Fast, All-Purpose State Storage. *The 16th International SPIN Workshop*, Springer LNCS 5578. June 2009.

13. Panagiotis Manolios. Automating System Assembly of Aerospace Systems. *The Sixth NASA Langley Formal Methods Workshop*, April 2008.

12. Peter Dillinger, Matt Kaufmann, and Panagiotis Manolios. Hacking and Extending ACL2. *Seventh International Workshop on the ACL2 Theorem Prover and Its Applications (ACL2 2007)*, November 2007.

11. Matt Kaufmann, Panagiotis Manolios, J S. Moore, and Daron Vroon. Integrating CCG Analysis into ACL2. *The Eight International Workshop on Termination*, part of FLoC'06, August 2006.

10. Peter C. Dillinger, Panagiotis Manolios, J S. Moore, and Daron Vroon. ACL2s: "The ACL2 Sedan". *User Interfaces for Theorem Provers Workshop*, part of FLoC'06, August 2006.

9. Panagiotis Manolios. Automating the Verification of RTL-Level Pipelined Machines. *DCC'06, The Seventh International Workshop on Designing Correct Circuits*, April, 2006.

8. Peter C. Dillinger and Panagiotis Manolios. Enhanced Probabilistic Verification with 3Spin and 3Murphi. *SPIN 2005, 12th International SPIN Workshop on Model Checking of Software*, volume 3639 of *LNCS*, pages 272–276. Springer, August 2005.

7. Panagiotis Manolios and Sudarshan Srinivasan. A Suite of Hard ACL2 Theorems Arising in Refinement-Based Processor Verification. *Fifth International Workshop on the ACL2 Theorem Prover and Its Applications (ACL2 2004)*. November 2004.

6. Marcio Gameiro and Panagiotis Manolios. Formally Verifying an Algorithm Based on Interval Arithmetic for Checking Transversality. *Fifth International Workshop on the ACL2 Theorem Prover and Its Applications (ACL2 2004)*, November 2004.

5. Peter C. Dillinger and Panagiotis Manolios. Fast *and* Accurate Bitstate Verification for SPIN. *SPIN 2004, 11th International SPIN Workshop on Model Checking of Software*, volume 2989 *LNCS*, pages 57–75. Springer-Verlag, April 2004.

4. Panagiotis Manolios and Daron Vroon. Ordinal Arithmetic in ACL2. *Fourth International Workshop on the ACL2 Theorem Prover and Its Applications (ACL2-2003)*, July 2003.

3. Panagiotis Manolios and Matt Kaufmann. Adding a Total Order to ACL2. In Matt Kaufmann and J Strother Moore, editors, *The Third International Workshop on the ACL2 Theorem Prover*. April 2002.

2. Panagiotis Manolios and J Strother Moore. Partial functions in ACL2. In Matt Kaufmann and J Strother Moore, editors, *Proceedings of the ACL2 Workshop 2000*. October 2000.

1. Panagiotis Manolios. Verification of pipelined machines in ACL2. In Matt Kaufmann and J Strother Moore, editors, *Proceedings of the ACL2 Workshop 2000*. October 2000.

H. Technical Reports

4. David A. Greve, Matt Kaufmann, Panagiotis Manolios, J S. Moore, Sandip Ray, Jose L. Ruiz-Reina, Rob Sumners, Daron Vroon, and Matthew Wilding. Efficient Execution in an Automated Reasoning Environment. The University of Texas at Austin, Department of Computer Sciences. Technical Report TR-06-59. November 20, 2006. 53 pages. (Expanded version of Journal paper E.6)

3. Panagiotis Manolios and Sudarshan K. Srinivasan. Automatic Verification of Safety and Liveness for XScale-Like Processor Models Using WEB-Refinements. CERCS TR# GIT-CERCS-03-17, September 2003. (Corresponds to Conference paper B.10)

2. Panagiotis Manolios. Enumerating the strings of a regular expression. Technical Report TR2000-30, Department of Computer Sciences, The University of Texas at Austin, December 2000.

1. Panagiotis Manolios, Kedar Namjoshi, and Robert Sumners. Linking theorem proving and model-checking with well-founded bisimulation. Technical Report TR-99-02, Department of Computer Sciences, The University of Texas at Austin, January 1999.

I. Software

8. Panagiotis Manolios and Vasilis Papavasileiou. Inez, a constraint solver that implements the ILP Modulo Theories (IMT) scheme, as described in our CAV 2013 paper. An IMT instance is an Integer Linear Programming instance, where some symbols have interpretations in background theories. We have used the IMT approach to solve industrial synthesis and design problems with real-time constraints arising in the development of the Boeing 787. Inez can be used to solve problems involving linear constraints and optimization. Inez is OCaml-centric. The preferred mode of interacting with the solver is via scripts written in a Camlp4-powered superset of OCaml.

7. Panagiotis Manolios, Sudarshan Srinivasan, and Daron Vroon. BAT, The Bit-level Analysis Tool, is a bounded model checker that can be used to verify properties of models written in the BAT language, a strongly typed language with three different kinds of types: bit vectors, memories, and multiple value types. BAT also allows user-defined functions and performs type inference. Overall, it makes it very convenient to define hardware models at the bit level. One of the novel aspects of BAT is that it implements a new sound and complete memory-abstraction algorithm that allows us to reduce the size of addresses and a technique for simplifying memories that combines term-rewriting with SAT solving. BAT can automatically verify bit-level pipelined machine designs beyond the reach of other state-of-the-art methods.

6. Panagiotis Manolios, Gayatri Subramanian, Vasilis Papavasiliou, and Daron Vroon. CoBaSA, Component Based System Assembly, is a tool for automatically solving one of the major challenges in the development of large component-based software systems, namely the system assembly problem: which components should be selected and how should they be connected, integrated, and assembled so that the overall system requirements are satisfied? CoBaSA includes an expressive language for declaratively describing system-level requirements, component interfaces and dependencies, resource requirements, safety properties, objective functions, and various types of constraints, including replication, timing, scheduling, and separation constraints (among others). We then automatically solve the system assembly problem using constraint-solving techniques that take advantage of current SAT-based methods. CoBaSA is being successfully applied to several large-scale industrial examples involving the Boeing 787 Dreamliner.

5. Harsh Chamarthi, Peter Dillinger, Panagiotis Manolios, and Daron Vroon. ACL2s. This is a version of the ACL2 theorem-proving system designed to make formal reasoning more widely accessible. To this end, we have developed a modern integrated development environment using Eclipse; we developed and implemented several abstractions that enable users to quickly determine the state of the theorem prover; and we have developed and implemented algorithms and techniques for simplifying the logic and user interactions with the theorem prover (*e.g.*, by providing powerful termination analysis methods).

4. Panagiotis Manolios. Bloom Filter Calculator. I developed a Web application to enable users of Bloom filters to optimally configure the data structure for their applications. The tool has been used over 25,000 times by a wide variety of users. The tool is available online at `http://www.cc.gatech.edu/ manolios/bloom-filters/calculator.html`.

3. Peter Dillinger and Panagiotis Manolios. We developed a new version of SPIN, called 3SPIN. SPIN is a popular model-checking tool that can be used for the formal verification of distributed software systems. The tool was developed at Bell Labs by Gerard Holzmann in the original Unix group of the Computing Sciences Research Center, starting in 1980, and continues to evolve to keep pace with new developments in the field. In April 2002, Gerard Holzmann was awarded the prestigious ACM System Software Award, for his work on SPIN. Our 3SPIN provides significant improvements with regard to the utilization of Bloom filters. Peter Dillinger worked with Holzmann at NASA JPL as a summer intern and some of these improvements have subsequently been added to SPIN.

2. Peter Dillinger and Panagiotis Manolios. We developed a new version of Murphi, called 3Murphi. Murphi is a popular model-checking tool developed at Stanford University by David Dill's group. Our modifications enhance the probabilistic algorithms and data structures for storing visited states, making them more effective and more usable for verifying huge transition systems. 3Murphi also supports a verification methodology designed to minimize time to finding errors, or to reaching a desired certainty that no errors exist. 3Murphi and 3SPIN are the only tools to offer this support, and do so with the most powerful and flexible currently-available implementations of the underlying algorithms and data structures.

1. Panagiotis Manolios and Daron Vroon. We modified ACL2's logic, specifically the internal representation and handling of the ordinals. We also added a powerful library of definitions and theorems about ordinal arithmetic and modified the extensive corpus of ACL2 case studies to conform. Our modifications have appeared in every version of ACL2 since version 2.8 (released in March 2004).

J. Research Proposals and Grants

24. **FORMED: Bringing Formal Methods to the Engineering Desktop**
    AFRL
    PI: BAE, Gregory Eakman
    NEU PI: Panagiotis Manolios
    Amount: $900,000, March 2014
    Duration: 2 years

23. **SHF: Small: Dynamic Abstractions for Verification**
    NSF
    PI: Panagiotis Manolios
    Co-PI: Eugene Goldberg
    Amount: $450,000 September 2013
    Duration: 3 years

22. **Efficient Bit-Level Solvers for Quantified Formulas**
    Semiconductor Research Corporation
    PI: Panagiotis Manolios
    Co-PI: Eugene Goldberg
    Amount: $25,000, September 2013
    Duration: 3 months

21. **Efficient Bit-Level Solvers for Quantified Formulas**
    Semiconductor Research Corporation
    PI: Panagiotis Manolios
    Co-PI: Eugene Goldberg
    Amount: $75,000, August 2012
    Duration: 1 years

20. **SHF: Small: Generation of High-Quality Tests by Treating Tests as Proof Encoding**
    NSF
    PI: Panagiotis Manolios
    Co-PI: Eugene Goldberg
    Amount: $495,258, September 2011
    Duration: 3 years

19. **GnoSys: Raising the Level of Discourse in Programming Systems**
    DARPA
    PI: Olin Shivers
    NEU Co-PIs: Matthias Felleisen, Panagiotis Manolios, and Mitch Wand
    Utah Co-PIS: Matthew Might and Matthew Flatt
    Amount: $4,384,488, November 2010
    Duration: 5 years

18. **Common Computing System Hosted Function Allocation Automation**
    Boeing Aerospace Company
    PI: Panagiotis Manolios
    Amount: $99,999, Jan 2010
    Duration: 1 year

17. **Common Computing System Hosted Function Allocation Automation**
    Boeing Aerospace Company
    PI: Panagiotis Manolios
    Amount: $207,000, Jan 2009
    Duration: 1 year

16. **Decision Procedures for RTL Verification**
    Semiconductor Research Corporation
    PI: Panagiotis Manolios
    Amount: $300,000, October 2008
    Duration: 3 years

15. **Towards a Theory of Service Assembly**
    IBM Faculty Award
    PI: Panagiotis Manolios
    Amount: $40,000, June 2008
    Duration: None (gift)

14. **Common Computing System Hosted Function Allocation Automation**
    Boeing Aerospace Company
    PI: Panagiotis Manolios
    Amount: $171,493, May 2008
    Duration: 1 year

13. **Hierarchical Component-Based Framework for the Formal Verification and Validation of Complex Aerospace Software Systems**

NASA, Aviation Safety: Integrated Vehicle Health Management Project
PI: Panagiotis Manolios
Co-PIs: Eric Feron (Georgia Tech, Aerospace), Cesar Munoz (National Institute of Aerospace), Arnaud Venet (Kestrell Technology)
Amount: $1,568,750, February 2007
Duration: 5 years

12. **Software Verification: Common Computing System Hosted Function Allocation Automation and Modeling/Analysis of Fault Propagation**
Boeing Aerospace Company
Co-PI with Mary Jean Harrold
Amount: $250,000, January 2007
Duration: 1 year
Note: This is to continue the 2006 work.

11. **Software Verification: Common Computing System Hosted Function Allocation Automation and Modeling/Analysis of Fault Propagation**
Boeing Aerospace Company
Co-PI with Mary Jean Harrold
Amount: $250,000, January 2006
Duration: 1 year

10. **Software Verification: Hosted Function Allocation Automation and Fault Propagation Analysis**
Boeing Aerospace Company
Co-PI with Mary Jean Harrold
Amount: $180,000, February 2005
Duration: 1 year

9. **An Eclipse-Based Integrated Development Environment for Integrating Computer-Aided Reasoning into the Computer Science Curriculum**
IBM, Eclipse Innovation Grant
Amount: $15,000, 2005
Duration: None, gift

8. **SoD: Collaborative: Language towers as design frameworks**
NSF, Science of Design Program
PI: Olin Shivers
CO-PIs: Panagiotis Manolios and Matthew Flatt (University of Utah)
Amount: $629,999 in total, $450,000 for Georgia Tech January 2005
Duration: 3 years; extended to 5

7. **Software Verification Through Interface Analysis and Change Impact Analysis**
Boeing Aerospace Company
PI: Panagiotis Manolios
Amount: $67,230, December 2004
Duration: 1 year

6. **System-Level Processor Verification Using Refinement**
NSF, Computing Processes & Artifacts Program
PI: Panagiotis Manolios
Amount: $250,000, September 2004
Duration: 3 years; extended to 5

5. **Integrating Functional Computer-Aided Reasoning into the Computer Science Curriculum**
   NSF, CISE Combined Research and Curriculum Development and Educational Innovation Program
   PI: Panagiotis Manolios
   Co-PI's: J Moore and Olin Shivers
   Amount: $470,000, August 2004
   Duration: 3 years; extended to 5

4. **Software Verification Through Interface Analysis and Change Impact Analysis**
   Boeing Aerospace Company
   Co-PI with Mary Jean Harrold
   Amount: $180,000, January 2004
   Duration: 1 year

3. **Software Verification Through Interface Analysis and Change Impact Analysis**
   Boeing Aerospace Company
   Co-PI with Mary Jean Harrold
   Amount: $50,000, Fall 2003
   Duration: 1 year

2. **Reconfigurable Logic Kits for Hardware-Centric Courses**
   Georgia Tech, Technology Fee Funds
   Co-PI with Kenneth Mackenzie
   Amount: $25,750, August 2002
   Duration: None (lifetime of equipment)

1. **Altera Donation: MAX+PLUS II/Quartus II**
   Altera Corporation
   PI: Panagiotis Manolios
   This was an equipment donation valued at $286,000, July 2001
   Duration: None (lifetime of equipment)

III. SERVICE

A. Editorial Boards

1. Associate Editor of ACM Transactions on Design Automation of Electronic Systems (TODAES). 3 Year term starting 2011.

B. Steering and Advisory Committee Memberships

7. Judge, SAT Competition 2014. The SAT Competition is a competitive event for solvers for the Boolean Satisfiability problem (SAT).

6. Technical Advisory Board Member, Reveal Design Automation, Inc., a startup company developing formal verification technology.

5. Steering Committee Member, International Workshop on the ACL2 Theorem Prover and Its Applications (ACL2) 2010-Present.

4. Steering Committee Member, ACM-IEEE Conference on Formal Methods in Computer-Aided Design (FMCAD). 2007-2013.

3. Advisory Panel Member, SAT-Race 2010. SAT-Race is a competitive event for solvers of the Boolean Satisfiability problem (SAT).

2. Steering Committee Member, International Workshop on the ACL2 Theorem Prover and Its Applications (ACL2) 2006-2009.

1. Advisory Panel Member, SAT-Race 2008. SAT-Race is a competitive event for solvers of the Boolean Satisfiability problem (SAT).

C. Chairmanship and Organizational Activities

4. Local arrangements chair, Eighth International Workshop on the ACL2 Theorem Prover and Its Applications (ACL2) 2009. Held at Northeastern, May, 2009.

3. Co-organized The New England Symposium on Formal Hardware Verification. Held at Northeastern, November, 2007.

2. Program and general chair (with Aarti Gupta), ACM-IEEE Conference on Formal Methods in Computer-Aided Design (FMCAD) 2006.

1. Program and general chair (with Matt Wilding), ACM International Workshop on the ACL2 Theorem Prover and Its Applications (ACL2) 2006.

D. Conference Program Committee Memberships

41. Formal Methods in Computer-Aided Design (FMCAD) 2015.

40. International Conference on Tests & Proofs (TAP) 2015.

39. International Conference on Theory and Applications of Satisfiability Testing (SAT) 2015.

38. Seventh NASA Formal Methods Symposium (NFM) 2015.

37. Formal Methods in Computer-Aided Design (FMCAD) 2014.

36. International Conference on Theory and Applications of Satisfiability Testing (SAT) 2014.

35. Interactive Theorem Proving (ITP) 2014.

34. Formal Methods in Computer-Aided Design (FMCAD) 2013.

33. Verified Software: Theories, Tools, and Experiments (VSTTE) 2013.

32. Interactive Theorem Proving (ITP) 2013.

31. Design Automation Conference (DAC) 2013.

30. International Conference on Theory and Applications of Satisfiability Testing (SAT) 2012.

29. Verified Software: Theories, Tools, and Experiments (VSTTE) 2012.

28. Formal Methods in Computer-Aided Design (FMCAD) 2011.

27. Interactive Theorem Proving (ITP) 2011.

26. International Conference on Theory and Applications of Satisfiability Testing (SAT) 2011.

25. International Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS) 2011.

24. International Symposium on Stabilization, Safety, and Security of Distributed Systems (SSS) 2010.

23. Formal Methods in Computer-Aided Design (FMCAD) 2010.
22. Interactive Theorem Proving (ITP) 2010.
21. International Conference on Theory and Applications of Satisfiability Testing (SAT) 2010.
20. Design, Automation & Test in Europe (DATE) 2010.
19. Formal Methods in Computer-Aided Design (FMCAD) 2009.
18. The Twelveth International Conference on Theory and Applications of Satisfiability Testing (SAT) 2009.
17. The 22nd International Conference on Theorem Proving in Higher Order Logics (TPHOLs) 2009.
16. The 15th International Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS) 2009.
15. International Conference on Verification, Model Checking, and Abstract Interpretation (VMCAI) 2009.
14. The IFIP Working Conference on Verified Software: Theories, Tools, Experiments (VSTTE) 2008.
13. ACM-IEEE Conference on Formal Methods in Computer-Aided Design (FMCAD) 2008.
12. The Eleventh International Conference on Theory and Applications of Satisfiability Testing (SAT) 2008.
11. ACM-IEEE International Conference on Computer-Aided Design (ICCAD) 2007.
10. ACM-IEEE Conference on Formal Methods in Computer-Aided Design (FMCAD) 2007.
9. The Tenth International Conference on Theory and Applications of Satisfiability Testing (SAT) 2007.
8. Asia and South Pacific Automation Conference (ASP-DAC) 2007.
7. ACM-IEEE International Conference on Computer-Aided Design (ICCAD) 2006.
6. ACM-IEEE International Conference on Formal Methods and Models for Codesign (MEMOCODE) 2006.
5. Advanced Research Working Conference on Correct Hardware Design and Verification Methods (CHARME) 2005.
4. Program Committee Member, International Conference on Theorem Proving in Higher Order Logics (TPHOLs) 2005.
3. Computer-Aided Verification (CAV) 2004.
2. Computer-Aided Verification (CAV) 2003.
1. International Conference on Computer Design (ICCD) 2002.

E. Workshop Program Committee Memberships

15. Automated Verification of Critical Systems Workshop (AVoCS) 2015.

14. Thirteenth International Workshop on the ACL2 Theorem Prover and Its Applications (ACL2) 2015.
13. Twelfth International Workshop on the ACL2 Theorem Prover and Its Applications (ACL2) 2014.
12. Eleventh International Workshop on the ACL2 Theorem Prover and Its Applications (ACL2) 2013.
11. Tenth International Workshop on the ACL2 Theorem Prover and Its Applications (ACL2) 2011.
10. Eighth International Workshop on the ACL2 Theorem Prover and Its Applications (ACL2) 2009.
9. Automated Formal Methods (AFM) 2009.
8. Automated Formal Methods (AFM) 2008.
7. The 1st International Workshop on Bit-Precise Reasoning (BPR) 2008.
6. Automated Formal Methods (AFM) 2007.
5. The International Workshop on the ACL2 Theorem Prover and Its Applications (ACL2) 2007.
4. Automated Formal Methods (AFM) 2006.
3. The Fifth International Workshop on the ACL2 Theorem Prover and Its Applications (ACL2) 2004.
2. International Workshop on the ACL2 Theorem Prover and Its Applications (ACL2) 2002.
1. International Workshop on the ACL2 Theorem Prover and Its Applications (ACL2) 2000.

F. Other Reviewing

17. Innovations in Systems and Software Engineering.
16. External Expert Reviewer for DAC (Design Automation Conference) 2012.
15. Journal of the ACM (JACM).
14. Transactions on Programming Languages and Systems (TOPLAS).
13. Journal of Automated Reasoning (JAR).
12. Information Processing Letters (IPL).
11. Formal Methods in Systems Design (FMSD).
10. Design and Test (D&T).
9. Computer-Aided Verification (CAV) 1999, 2000, 2006, 2008.
8. Formal Methods Europe (FME) 2001.
7. Correct Hardware Design and Verification Methods (CHARME) 1999.
6. International Conference on Computer Design (ICCD) 2002.
5. International Conference on Software Engineering. (ICSE) 2002.

4. ACM Transactions on Embedded Computing Systems.

3. Logic in Computer Science (LICS) 2004.

2. ACM Programming Language and Implementation Conference (PLDI) 2007.

1. ACM Symposium on Principles of Programming Languages (POPL) 2009.

G. Project Reviewer

5. NSF Panel Reviewer for the Software and Hardware Foundations Program, Washington DC. May 2014.

4. NSF Panel Reviewer for the Software and Hardware Foundations Program, Washington DC. January 2013.

3. NSF Panel Reviewer for the Software and Hardware Foundations Program, Washington DC. April 2009.

2. The National Security Agency's Mathematical Sciences Program.

1. NSF Panel Reviewer for the Software Engineering and Programming Languages Program, Washington DC. March 2002.

H. Honors and Activities

9. Member of IFIP Working group 1.9/2.15 on Verified Software, 2011–Present.

8. IBM Faculty Award, 2005, 2008.

7. Member of the ACM (9/2002-), the IEEE (2006-), and the Honor Society of Phi Kapppa Phi.

6. Member of Edsger W. Dijkstra's ATAC (Austin Tuesday Afternoon Club), 1998–2001.

5. Certificate of Achievement for Outstanding Achievement in Teaching for the Year of 1998-1999 (Outstanding Teaching Assistant Award), The University of Texas at Austin, 1999.

4. NSF grant to attend the Marktoberdorf Summer School on Calculational System Design, an Advanced Study Institute of the NATO Science Committee, 1998.

3. Micro-Electronics and Computer Development Fellowship, The University of Texas at Austin, 1994-1996.

2. Awarded Certificate - Seminar on Teaching in Urban Colleges, funded by FIPSE, 1992.

1. Elected an Associate Member of Sigma Xi, The Scientific Research Society in 1992. Promoted to Full Member in 2007.

I. On-campus Committees

15. College of Computer and Information Science, Northeastern University. Hiring Committee. Chairman. 2014-2015 Academic Year.

14. Northeastern University, College of Computer and Information Science Representative. Graduate Council. 2012-2013 Academic Year.

13. College of Computer and Information Science, Northeastern University. Merit Committee. 2012-2013 Academic Year.

12. College of Computer and Information Science, Northeastern University. Undergraduate Committee. 2012-2013 Academic Year.

11. College of Computer and Information Science, Northeastern University. Graduate Curriculum Committee (Ad Hoc). 2011-2012 Academic Year.

10. College of Computer and Information Science, Northeastern University. Ph.D. Committee. Chairman. 2011-2012 Academic Year.

9. College of Computer and Information Science, Northeastern University. Ph.D. Committee. Chairman. 2010-2011 Academic Year.

8. College of Computer and Information Science, Northeastern University. College Review (Ad Hoc) Committee. 2010-2011 Academic Year.

7. College of Computer and Information Science, Northeastern University. Merit Evaluation (Ad Hoc) Committee. 2009-2010 Academic Year.

6. College of Computer and Information Science, Northeastern University. Ph.D. Committee. 2009-2010 Academic Year.

5. College of Computer and Information Science, Northeastern University. Ph.D. Committee. 2008-2009 Academic Year.

4. College of Computer and Information Science, Northeastern University. Ph.D. Committee. 2007-2008 Academic Year.

3. College of Computing, Computing and Systems Division, Georgia Tech. Graduate Committee. 2005-2006 Academic Year.

2. College of Computing, Core Computing Division. Steering Committee. 2003-2005 Academic Years.

1. College of Computing, Recruiting Committee. 2002-2003 Academic Year.

J. External Committees

12. FMCAD 2013 Panel Leader for “Teaching Formal Methods: Needs, Challenges, Experiences, and Opportunities”

11. ACL2 2013 Panel Leader for “Using ACL2 to Teach Undergraduates”

10. ACL2 2013 Panel Member for “ACL2 Extension Fragmentation”

9. Ph.D. Examining Committee for Vlad Slavici, College of Computer and Information Science, Northeastern University. Title: TBD. Advisor: Gene Cooperman.

8. Ph.D. Examining Committee for Carl Eastlund, College of Computer and Information Science, Northeastern University. Title: TBD. Advisor: Matthias Felleisen.

7. Ph.D. Examining Committee for Dan Kunkle, College of Computer and Information Science, Northeastern University. Title: Roomy: A New Approach to Parallel Disk-based Computation, January 2011. Advisor: Gene Cooperman.

6. Ph.D. Examining Committee for Eric Robinson, College of Computer and Information Science, Northeastern University. Title: Large Implicit State Space Enumeration: Overcoming Memory and Disk Limitations, April 2008. Advisor: Gene Cooperman.

5. Ph.D. Examining Committee for Matthew Might, College of Computing, Georgia Tech. Title: Environment Analysis of Higher-Order Languages, June 2007. Advisor: Olin Shivers.

4. Verified Software: Theories, Tools, Experiments. Member of panel on Tool Interoperability. Charged to come up with a report on languages and language families that can be employed for diverse tools to cooperate in solving verification problems. This is a critical issue for the Verified Software Grand Challenge. 2006.

3. Ph.D. Examining Committee for Jaehwan Lee, School of Electrical and Computer Engineering, Georgia Tech. Title: Hardware/Software Deadlock Avoidance for Multiprocessor Multiresource System-on-a-Chip, Oct. 2004. Advisor: Vincent J. Mooney III.

2. Ph.D. Examining Committee for Zachary Kurmas, College of Computing, Georgia Tech. Title: Generating and Analyzing Synthetic Workloads using Iterative Distillation, May 2004. Advisor: Kishore Ramachandran.

1. Masters Examining Committee for Tiffany Danielle Goble, School of Mathematics, Georgia Tech. Title: Automated Reasoning: Computer Assisted Proofs in Set Theory Using Godel's Algorithm for Class Formation, Aug. 2004. Advisor: Johan G.F. Belinfante.

K. Workshops and External Courses

7. Counterexample Generation Meets Interactive Theorem Proving: Current Results and Future Opportunities. Invited Tutorial, ITP, Interactive Theorem Proving (ITP 2013), July 2013.

6. SMT for Scheduling and Synthesis of Cyber-Physical Architectural Models. Second International SAT/SMT Summer School. Trento, Italy, June 12-15, 2012.

5. Reasoning About Programs with ACL2. Summer School on Logic and Theorem Proving in Programming Languages. University of Oregon (Eugene, Oregon), July 22-30, 2008.

4. Refinement and Theorem Proving. Lectures at SFM-06:HV, The 6th International School on Formal Methods for the Design of Computer, Communication and Software Systems: Hardware Verification. Bertinoro, Italy. May 22-27, 2006. The lectures were published by Springer Verlag, see [F.2].

3. ACL2 Tutorial for the Conference on Automated Deduction (CADE 18), part of The 2002 Federated Logic Conference (FLoC'02). Copenhagen, Denmark. July 2002.

2. ACL2 Tutorial for the European Joint Conferences on Theory and Practice of Software (ETAPS). Grenoble, France. April 2002.

1. Organizing Committee Member, ACL2 Workshop 1999.

L. Invited Talks

I have given over 50 invited presentations ranging from lecturing at the International School on Satisfiability Modulo Theories, to giving invited tutorials at ITP and FMCAD, to invited talks at conferences such as the NSA's High Confidence Software Systems Conference to giving colloquia at various institutions including the following: AT&T Research Labs, Bell Labs, Brown University, Carnegie Mellon (School of Computer Science and Software Engineering Institute), Cornell, Kestrel Institute, Microsoft Research (Cambridge, Redmond, and Silicon Valley), MIT, NASA Jet Propulsion Laboratory, New York University, Northeastern University, Ohio State University, Oxford University, SRI International, Tufts, University of Illinois at Urbana-Champaign, University of Michigan, University of Texas at Austin, University of Iowa, University of Utah, University of Virginia, Washington University in St. Louis., and WPI