CURRICULUM VITAE

YANHONG ANNIE LIU
Computer Science Department
Stony Brook University
Stony Brook, NY 11794-4400
(631) 632-8463
liu@cs.stonybrook.edu
http://www.cs.stonybrook.edu/~liu

November 6, 2016

RESEARCH INTERESTS

Languages and algorithms, especially systematic methods for design and optimizations. Program analysis and transformation for incremental computation and concurrent and distributed computation. Applications in optimizing compilers, interactive environments, real-time and embedded systems, database systems, semantic web, distributed systems, big data analysis, and security.

EDUCATION

08/90 – 09/95	Cornell University	GPA 4.0+	• Ph.D.,	Computer Science,	January 1996
			• M.S.,	Computer Science,	May 1992
09/87 – 01/90	Tsinghua University	GPA 4.0	• M.Eng.,	Computer Science,	November 1988
09/83 – 08/87	Peking University	GPA 3.9	• B.S.,	Computer Science,	June 1987

RESEARCH AND WORK EXPERIENCE

08/08 –present Professor, Computer Science Department, Stony Brook University
07/08 – 07/08 Visiting Researcher, Microsoft Research Asia
09/04 – 07/06 Member of Chair Professor Team of Andrew Yao, Tsinghua University
08/00 – 08/08 Associate Professor, Computer Science Department, Stony Brook University
08/96 – 08/00 Assistant Professor, Computer Science Department, Indiana University

09/95 – 07/96 Post-Doctoral Associate, Computer Science Department, Cornell University
05/91 – 09/95 Graduate Research Assistant, Computer Science Department, Cornell University
05/92 – 08/92 Summer Research Intern, System Science Lab., Xerox Webster Research Center
08/90 – 12/91 Teaching Assistant, Computer Science Department, Cornell University

01/88 – 07/90 Researcher, Res. Inst. of Petroleum Exploration & Development and Tsinghua Univ.
09/87 – 01/90 Research Assistant, Computer Science Department, Tsinghua University
01/88 – 07/88 Teaching Assistant, Computer Science Department, Tsinghua University

07/87 – 08/87 Analyst in a study of software industry in major cities of northeast China
07/86 – 07/86 Developer of a logic DB query language interpreter for CAI at Peking University
05/85 – 09/85 Programmer/Analyst for MIS development at Nanjing Automobile Corporation
07/84 – 10/84 Programmer for the personnel administration system at Peking University

AWARDS AND HONORS

- Excellence in Research, Computer Science Department, Stony Brook University, August 2016.
- Best Student Paper Award, for High-Level Executable Specifications of Distributed Algorithms by Y. A. Liu, S. D. Stoller, and B. Lin, 14th International Symposium on Stabilization, Safety, and Security of Distributed Systems, Toronto, October 2012.
- Chancellor's Award for Excellence in Scholarship and Creative Activities, State University of New York, April 2010.
- Undergraduate College Faculty Fellow, Stony Brook University, 2009.
- Recognition of Service Award, Association for Computing Machinery (ACM), May 1999.
- Elected Member of IFIP Working Group 2.1, Oxford, U.K., January 1998.
- Liu Memorial Award, for Excellent Progress in the Doctoral Program and High Potential for a Successful Academic Career, The Graduate School, Cornell University, May 1992.
- Awarded guaranteed admission to all graduate schools in China, Peking Univ., June 1986.
- Numerous awards and prizes at Peking University and Tsinghua University, 1984–1989.
- Winner (fourteenth place) of the National Mathematics Contest, Beijing region, May 1982.

GRANTS

- Co-Principal Investigator, *Algorithm Diversity for Resilient Systems*, with Scott Stoller, Principal Investigator. Office of Naval Research, 2015–2018.
- Principal Investigator, *SHF: Large: From Clarity to Efficiency for Distributed Algorithms*, with Scott Stoller, co-Principal Investigator. National Science Foundation, 2014–2018.
- Co-Principal Investigator, *BIGDATA: F: DKM: DKA: Big Data Modeling and Analysis with Depth and Scale*, with C.R. Ramakrishnan, Principal Investigator, Maureen O'Leary, I.V. Ramakrishnan, Scott A. Smolka, and David S. Warren, co-Principal Investigators. National Science Foundation, 2014–2018.
- Co-Principal Investigator, *Collaborative Research: Innovative Active Learning Using Tablets*, with Ahmad Esmaili, Principal Investigator. National Science Foundation, 2013–2016.
- Principal Investigator, *EAGER: From Clarity to Efficiency for Distributed Algorithms*, with Scott Stoller, co-Principal Investigator. National Science Foundation, 2012–2016.
- Principle Investigator, *NSF/TCPP Curriculum: Concurrent and Distributed Algorithms*, with Scott Stoller, co-Principle Investigator. NSF/IEEE-TCPP Curriculum Initiative on Parallel and Distributed Computing Early Adopter Program, 2012-2013.
- Co-Principal Investigator, *SHF: Medium: Performance Analysis and Optimization for Logic Rule Engines*, with Michael Kifer, Principal Investigator, and David Warren, co-Principal Investigator. National Science Foundation, 2010–2016.
- Principal Investigator, *Generating Incremental Implementations from Set-Based Specifications*. Hengsoft, 2009-2010.
- Principal Investigator, *Invariant Rules for Software Producibility and Assurance*, with Scott Stoller, co-Principal Investigator. Office of Naval Research, 2009–2010.

- Principal Investigator, *Nuclear Power Plant Digital Feedwater Control System Modeling and Simulation.* Brookhaven National Laboratory, 2007–2008.
- Principal Investigator, *Clarity and Efficiency in Design*, with Scott Stoller, co-Principal Investigator. National Science Foundation, 2006–2011.
- Co-Principal Investigator, *Tools for Detecting and Reconciling Security Policy Conflicts*, with Scott Stoller, Principal Investigator. Office of Naval Research through Naval Research Laboratory, 2006–2008.
- Co-Principal Investigator, *Runtime Monitoring and Model Checking for High-Confidence System Software*, with Erez Zadok, Principal Investigator, and Radu Grosu, Scott Smolka, and Scott Stoller, co-Principal Investigators. National Science Foundation, 2005–2010.
- Principal Investigator, *Ad Hoc Command, Control, and Communication Networks for Homeland Security*, with Alex Mohr, co-Principal Investigator. Convergent Technologies Consortium, 2005.
- Co-Principal Investigator, *Generating Efficient Trust Management Software from Policies*, with Scott Stoller, Principal Investigator. Office of Naval Research, 2004–2007.
- Principal Investigator, *From Rules to Analysis Algorithms with Time and Space Guarantees.* National Science Foundation, 2003–2008.
- Co-Principal Investigator, *A Deductive Engine for the Semantic Web*, with Michael Kifer, Principal Investigator, and C.R. Ramakrishnan and I.V. Ramakrishnan, co-Principal Investigators. National Science Foundation, 2003–2007.
- Principal Investigator, *From Rules to Analysis Algorithms with Time and Space Guarantees.* National Science Foundation, 2002–2004.
- Principal Investigator, *Motion Controller Scripting Language.* Anorad Corporation and Stony Brook Strategic Partnership for Industrial Resurgence Grant, 2002.
- Project Director, *Establishing Design and Analysis Research Lab.* Stony Brook Graduate Research Initiative and University Research Development Fund, 2001–2002.
- Principal Investigator, *Program Transformation and Analysis for Reactive Systems*, with Scott Stoller, co-Principal Investigator. Office of Naval Research, 2000–2004.
- Principal Investigator, *Transformational Development of Reactive Systems.* Office of Naval Research, 1998–2001.
- Principal Investigator, *A General and Powerful Method for Program Optimization*, with Scott Stoller, co-Principal Investigator. National Science Foundation, 1997–2001.
- Principal Investigator, *Automating Incrementalization—A Key Optimization to Produce Robust High-Performance Software from High-Level Specifications.* Motorola University Partnership in Research Grant, 1997–2000.
- Principal Investigator, *Investigation into Incrementalization.* Motorola University Research Grant, 1997.

PUBLICATIONS

Book

1. Y. A. Liu. *Systematic Program Design: From Clarity To Efficiency.* Cambridge University Press, 2013.

Ph.D. Dissertation

2. Y. A. Liu. *Incremental Computation: A Semantics-Based Systematic Transformational Approach.* Ph.D. Dissertation, Cornell University, Ithaca, New York, January 1996.

Book Articles (Refereed)

3. Y. A. Liu and J. J. P. Tsai. System monitoring. In Benjamin W. Wah, editor, *Wiley Encyclopedia of Computer Science and Engineering*, John Wiley & Sons, 2008.
4. Y. A. Liu and S. D. Stoller. Dynamic programming via static incrementalization. In *Automatic Program Development: a Tribute to Robert Paige*, Springer, 2008.
5. Y. A. Liu and S. D. Stoller. Role-based access control: A corrected and simplified specification. In *Department of Defense Sponsored Information Security Research: New Methods for Protecting Against Cyber Threats*, pages 425–439, John Wiley & Sons, 2007.
6. S. D. Stoller and Y. A. Liu. Generating efficient security software from policies. In *Department of Defense Sponsored Information Security Research: New Methods for Protecting Against Cyber Threats*, pages 416-424, John Wiley & Sons, 2007.
7. Y. A. Liu and J. J. P. Tsai. System monitoring. In John G. Webster, editor, *Wiley Encyclopedia of Electrical and Electronics Engineering*, volume 21, pages 293–301, John Wiley & Sons, 1999.
8. Y. A. Liu. Principled strength reduction. In Richard S. Bird and Lambert Meertens, editors, *Algorithmic Languages and Calculi*, pages 357–381. Chapman & Hall, London, 1997.

Journal Articles (Refereed)

9. Y. A. Liu, S. D. Stoller, and B. Lin. From clarity to efficiency for distributed algorithms. *ACM Transactions on Programming Languages and Systems.* To appear.
10. K. T. Tekle and Y. A. Liu. Precise complexity guarantees for pointer analysis via Datalog with extensions. *Theory and Practice of Logic Programming*, 16(5-6):916-932, September 2016. Cambridge University Press.
11. Y. Liu, W. Chen, Y. A. Liu, J. Sun, S. J. Zhang, and J. S. Dong. Verifying linearizability via optimized refinement checking. *IEEE Transactions on Software Engineering*, 39(7):1018–1039, July 2013.
12. Y. A. Liu and S. D. Stoller. From Datalog rules to efficient programs with time and space guarantees. *ACM Transactions on Programming Languages and Systems*, 31(6):1–38, August 2009.
13. Y. A. Liu, S. D. Stoller, N. Li, and T. Rothamel. Optimizing aggregate array computations in loops. *ACM Transactions on Programming Languages and Systems*, 27(1):1–35, January 2005.
14. Y. A. Liu and S. D. Stoller. Eliminating dead code on recursive data. *Science of Computer Programming*, 47(2–3):221–242, May–June 2003.

15. Y. A. Liu and S. D. Stoller. Dynamic programming via static incrementalization. *Higher-Order and Symbolic Computation*, 16(1–2):37–62, March–June 2003. Special issue in memory of Bob Paige.
16. S. D. Johnson, Y. A. Liu, and Y. Zhang. A systematic incrementalization technique and its application to hardware design. *International Journal on Software Tools for Technology Transfer*, 4(2):211–223, 2003.
17. Y. A. Liu and G. Gómez. Automatic accurate cost-bound analysis for high-level languages. *IEEE Transactions on Computers*, 50(12):1295–1309, December 2001.
18. Y. A. Liu, S. D. Stoller, and T. Teitelbaum. Strengthening invariants for efficient computation. *Science of Computer Programming*, 41(2):139–172, October 2001.
19. Y. A. Liu. Efficiency by incrementalization: An introduction. *Higher-Order and Symbolic Computation*, 13(4):289–313, December 2000.
20. Y. A. Liu, S. D. Stoller, and T. Teitelbaum. Static caching for incremental computation. *ACM Transactions on Programming Languages and Systems*, 20(3):546–585, May 1998.
21. Y. A. Liu and T. Teitelbaum. Systematic derivation of incremental programs. *Science of Computer Programming*, 24(1):1–39, February 1995.
22. Y. Liu. Analysis and study of multi-factor combination problems. *Jinan Natural Science and Medical Science Journal*, pages 129–132, October 1988. Special issue for selected papers from 1st National Symposium on Intelligence Science. (In Chinese.)

Conference Publications (Refereed Papers)

23. S. Chand, Y. A. Liu, and S. D. Stoller. Formal verification of multi-Paxos for distributed consensus. In *Proceedings of the 21st International Symposium on Formal Methods*, Limassol, Cyprus, November 2016. Springer. To appear.
24. K. T. Tekle and Y. A. Liu. Precise complexity guarantees for pointer analysis via Datalog with extensions. In *Proceedings of the 32nd International Conference on Logic Programming*, New York City, New York, October 2016. Cambridge University Press.
25. Y. A. Liu, J. Brandvein, S. D. Stoller, and B. Lin. Demand-driven incremental object queries. In *Proceedings of the 18th International Symposium on Principles and Practice of Declarative Programming*, pages 228–241, Edinburgh, United Kingdom, September 2016. ACM Press.
26. J. Brandvein and Y. A. Liu. Removing runtime overhead for optimized object queries. In *Proceedings of the ACM SIGPLAN 2016 Workshop on Partial Evaluation and Program Manipulation*, pages 73–84, St. Petersburg, Florida, January 2016. ACM Press.
27. Y. A. Liu, S. D. Stoller, B. Lin, and M. Gorbovitski. From clarity to efficiency for distributed algorithms. In *Proceedings of the 27th ACM SIGPLAN Conference on Object-Oriented Programming, Systems, Languages and Applications*, pages 395–410, Tucson, Arizona, October 2012. ACM Press.
28. Y. A. Liu, S. D. Stoller, and B. Lin. High-level executable specifications of distributed algorithms. In *Proceedings of the 14th International Symposium on Stabilization, Safety, and Security of Distributed Systems*, volume 7596 of *Lecture Notes in Computer Science*, pages 95–110, Toronto, Canada, October 2012. Springer. (Best Student Paper Award.)

29. M. Gorbovitski, Y. A. Liu, S. D. Stoller, and T. Rothamel. Composing transformations for instrumentation and optimization. In *Proceedings of the ACM SIGPLAN 2012 Workshop on Partial Evaluation and Program Manipulation*, pages 53–62, Philadelphia, Pennsylvania, January 2012. ACM Press.

30. K. T. Tekle and Y. A. Liu. More efficient Datalog queries: Subsumptive tabling beats magic sets. In *Proceedings of the 2011 ACM SIGMOD International Conference on Management of Data*, pages 661–672, Athens, Greece, June 2011. ACM Press.

31. M. Gorbovitski, Y. A. Liu, S. D. Stoller, T. Rothamel, and K.T. Tekle. Alias analysis for optimization of dynamic languages. In *Proceedings of the 6th Symposium on Dynamic Languages*, pages 27–42, Reno, Nevada, October 2010. ACM Press.

32. K. T. Tekle and Y. A. Liu. Precise complexity analysis for efficient Datalog queries. In *Proceedings of the 12th International ACM SIGPLAN Symposium on Principles and Practice of Declarative Programming*, pages 35–44, Hagenberg, Austria, July 2010. ACM Press.

33. K. T. Tekle, M. Gorbovitski, and Y. A. Liu. Graph queries through Datalog optimizations. In *Proceedings of the 12th International ACM SIGPLAN Symposium on Principles and Practice of Declarative Programming*, pages 25–34, Hagenberg, Austria, July 2010. ACM Press.

34. Y. Liu, W. Chen, Y. A. Liu, and J. Sun. Model checking linearizability via refinement. In *Proceedings of the 16th International Symposium on Formal Methods*, pages 321–337, Eindhoven, The Netherlands, November 2009. Springer.

35. Y. A. Liu, M. Gorbovitski, and S. D. Stoller. A language and framework for invariant-driven transformations. In *Proceedings of the 8th International Conference on Generative Programming and Component Engineering*, pages 55–64, Denver, Colorado, October 2009. ACM Press.

36. S. J. Zhang, Y. Liu, J. Sun, J. S. Dong, W. Chen, and Y. A. Liu. Formal verification of scalable nonzero indicators. In *Proceedings of the 21st International Conference on Software Engineering and Knowledge Engineering*, pages 406–411, Boston, Massachusetts, July 2009.

37. T. Rothamel and Y. A. Liu. Generating incremental implementations of object-set queries. In *Proceedings of the 7th International Conference on Generative Programming and Component Engineering*, pages 55–66, Nashville, Tennessee, October 2008. ACM Press.

38. M. Gorbovitski, K. T. Tekle, T. Rothamel, S. D. Stoller, and Y. A. Liu. Analysis and transformations for efficient query-based debugging. In *Proceedings of the 8th IEEE International Working Conference on Source Code Analysis and Manipulation*, pages 174–183, Beijing, China, September 2008. IEEE CS Press.

39. K. T. Tekle, K. Hristova, and Y. A. Liu. Generating specialized rules and programs for demand-driven analysis. In *Proceedings of the 12th International Conference on Algebraic Methodology and Software Technology*, pages 346–361, Urbana, Illinois, July 2008. Springer.

40. M. Gorbovitski, T. Rothamel, Y. A. Liu, and S. D. Stoller. Efficient runtime invariant checking: A framework and case study. In *Proceedings of the 6th International Workshop on Dynamic Analysis*, pages 43–49, Seattle, Washington, July 2008. ACM Press.

41. K. Hristova, K. T. Tekle, and Y. A. Liu. Efficient trust management policy analysis from rules. In *Proceedings of the 9th ACM SIGPLAN International Conference on Principles and*

Practice of Declarative Programming, pages 211–220, Wroclaw, Poland, July 2007. ACM Press.

42. T. Rothamel and Y. A. Liu. Efficient implementation of tuple pattern based retrieval. In *Proceedings of the ACM SIGPLAN 2007 Workshop on Partial Evaluation and Program Manipulation*, pages 81–90, Nice, France, January 2007. ACM Press.
43. T. Rothamel, Y. A. Liu, C. L. Heitmeyer, and E. I. Leonard. Generating optimized code from SCR specifications. In *Proceedings of the ACM SIGPLAN/SIGBED 2006 Conference on Languages, Compilers, and Tools for Embedded Systems*, pages 135–144, Ottawa, Canada, June 2006. ACM Press.
44. K. Hristova, T. Rothamel, Y. A. Liu, and S. D. Stoller. Efficient type inference for secure information flow. In *Proceedings of the ACM SIGPLAN Workshop on Programming Languages and Analysis for Security*, pages 85–94, Ottawa, Canada, June 2006. ACM Press.
45. Y. A. Liu and S. D. Stoller. Querying complex graphs. In *Proceedings of the 8th International Symposium on Practical Aspects of Declarative Languages*, volume 3819 of *Lecture Notes in Computer Science*, pages 199–214, Charleston, South Carolina, January 2006. Springer.
46. Y. A. Liu, C. Wang, M. Gorbovitski, T. Rothamel, Y. Cheng, Y. Zhao, and J. Zhang. Core role-based access control: Efficient implementations by transformations. In *Proceedings of the ACM SIGPLAN 2006 Workshop on Partial Evaluation and Semantics-Based Program Manipulation*, pages 112–120, Charleston, South Carolina, January 2006. ACM Press.
47. K. Hristova and Y. A. Liu. Improved algorithm complexities for linear temporal logic model checking of pushdown systems. In *Proceedings of the 7th International Conference on Verification, Model Checking and Abstract Interpretation*, volume 3855 of *Lecture Notes in Computer Science*, pages 190–206, Charleston, South Carolina, January 2006. Springer.
48. Y. A. Liu, S. D. Stoller, M. Gorbovitski, T. Rothamel, and Y. E. Liu. Incrementalization across object abstraction. In *Proceedings of the 20th ACM Conference on Object-Oriented Programming, Systems, Languages, and Applications*, pages 473–486, San Diego, California, October 2005. ACM Press.
49. Y. A. Liu and S. D. Stoller. A declarative framework for transformation and translation. In *Proceedings of the 2nd International Conference on Knowledge Economy and Development of Science and Technology*, Beijing, China, September 17–19, 2004. Tsinghua University Press and Springer.
50. Y. A. Liu, T. Rothamel, F. Yu, S. D. Stoller, and N. Hu. Parametric regular path queries. In *Proceedings of the ACM SIGPLAN 2004 Conference on Programming Languages Design and Implementation*, pages 219–230, Washington, DC, June 2004. ACM Press.
51. Y. A. Liu and S. D. Stoller. From Datalog rules to efficient programs with time and space guarantees. In *Proceedings of the 5th ACM SIGPLAN International Conference on Principles and Practice of Declarative Programming*, pages 172–183, Uppsala, Sweden, August 2003. ACM Press.
52. Y. A. Liu and S. D. Stoller. Optimizing Ackermann's function by incrementalization. In *Proceedings of the ACM SIGPLAN 2003 Workshop on Partial Evaluation and Semantics-Based Program Manipulation*, pages 85–91, San Diego, California, June 2003. ACM Press.

53. L. Unnikrishnan, S. D. Stoller, and Y. A. Liu. Optimized live heap bound analysis. In *Proceedings of the 4th International Conference on Verification, Model Checking and Abstract Interpretation*, volume 2575 of *Lecture Notes in Computer Science*, pages 70–85, New York City, New York, January 2003. Springer.

54. Y. A. Liu and F. Yu. Solving regular path queries. In *Proceedings of the 6th International Conference on Mathematics of Program Construction*, volume 2386 of *Lecture Notes in Computer Science*, pages 195–208, Schloss Dagstuhl, Germany, July 2002. Springer.

55. Y. A. Liu and S. D. Stoller. Program optimization using indexed and recursive data structures. In *Proceedings of the ACM SIGPLAN 2002 Workshop on Partial Evaluation and Semantics-Based Program Manipulation*, pages 108–118, Portland, Oregon, January 2002. ACM Press.

56. G. Gómez and Y. A. Liu. Automatic time-bound analysis for a higher-order language. In *Proceedings of the ACM SIGPLAN 2002 Workshop on Partial Evaluation and Semantics-Based Program Manipulation*, pages 75–86, Portland, Oregon, January 2002. ACM Press.

57. R. Grosu, Y. A. Liu, S. Smolka, S. D. Stoller, and J. Yan. Automated software engineering using concurrent class machines. In *Proceedings of the 16th IEEE Automated Software Engineering Conference*, pages 297–304, San Diego, California, November 2001. IEEE CS Press.

58. Y. A. Liu, N. Li, and S. D. Stoller. Solving regular tree grammar based constraints. In *Proceedings of the 8th International Static Analysis Symposium*, volume 2126 of *Lecture Notes in Computer Science*, pages 213–233, Paris, France, July 2001. Springer.

59. L. Unnikrishnan, S. D. Stoller, and Y. A. Liu. Automatic accurate live memory analysis for garbage-collected languages. In *Proceedings of the ACM SIGPLAN 2001 Workshop on Languages, Compilers, and Tools for Embedded Systems*, pages 102–111, Snowbird, Utah, June 2001. ACM Press.

60. S. D. Stoller and Y. A. Liu. Transformations for model checking distributed Java programs. In *Proceedings of the 8th International SPIN Workshop on Model Checking of Software*, volume 2057 of *Lecture Notes in Computer Science*, pages 192-199, Toronto, Canada, May 2001. Springer.

61. S. D. Stoller, L. Unnikrishnan, and Y. A. Liu. Efficient detection of global properties in distributed systems using partial-order methods. In *Proceedings of the 12th Conference on Computer-Aided Verification*, volume 1855 of *Lecture Notes in Computer Science*, Chicago, Illinois, July 2000. Springer.

62. Y. A. Liu and S. D. Stoller. From recursion to iteration: What are the optimizations? In *Proceedings of the ACM SIGPLAN 2000 Workshop on Partial Evaluation and Semantics-Based Program Manipulation*, pages 73–82, Boston, Massachusetts, January 2000. ACM Press.

63. S. D. Johnson, Y. A. Liu, and Y. Zhang. A systematic incrementalization technique and its application to hardware design. In *Proceedings of 10th IFIP WG10.5 Advanced Research Working Conference on Correct Hardware Design and Verification Methods*, volume 1703 of *Lecture Notes in Computer Science*, pages 334–337, Bad Herrenalb, Germany, September 1999. Springer.

64. Y. A. Liu and S. D. Stoller. Eliminating dead code on recursive data. In *Proceedings of the 6th International Static Analysis Symposium*, volume 1694 of *Lecture Notes in Computer Science*, pages 211–231, Venice, Italy, September 1999. Springer.

65. Y. A. Liu and S. D. Stoller. Dynamic programming via static incrementalization. In *Proceedings of the 8th European Symposium on Programming*, volume 1576 of *Lecture Notes in Computer Science*, pages 288–305, Amsterdam, The Netherlands, March 1999. Springer.

66. S. D. Stoller and Y. A. Liu. Efficient symbolic detection of global properties in distributed systems. In *Proceedings of the 10th Conference on Computer-Aided Verification*, volume 1427 of *Lecture Notes in Computer Science*, pages 357–368, Vancouver, British Columbia, Canada, June–July, 1998. Springer.

67. Y. A. Liu and G. Gómez. Automatic accurate time-bound analysis for high-level languages. In *Proceedings of the ACM SIGPLAN 1998 Workshop on Languages, Compilers, and Tools for Embedded Systems*, volume 1474 of *Lecture Notes in Computer Science*, pages 31–40, Montreal, Canada, June 1998. Springer.

68. Y. A. Liu. Dependence analysis for recursive data. In *Proceedings of the IEEE 1998 International Conference on Computer Languages*, pages 206–215, Chicago, Illinois, May 1998. IEEE CS Press.

69. Y. A. Liu and S. D. Stoller. Loop optimization for aggregate array computations. In *Proceedings of the IEEE 1998 International Conference on Computer Languages*, pages 262–271, Chicago, Illinois, May 1998. IEEE CS Press.

70. Y. A. Liu. Principled strength reduction. In *Proceedings of the IFIP TC2 Working Conference on Algorithmic Languages and Calculi*, Le Bischenberg, Alsace, France, February 1997.

71. Y. A. Liu, S. D. Stoller, and T. Teitelbaum. Discovering auxiliary information for incremental computation. In *Proceedings of the 23rd Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages*, pages 157–170, St. Petersburg Beach, Florida, January 1996. ACM Press.

72. Y. A. Liu. CACHET: An interactive, incremental-attribution-based program transformation system for deriving incremental programs. In *Proceedings of the 10th IEEE Knowledge-Based Software Engineering Conference*, pages 19–26, Boston, Massachusetts, November 1995. IEEE CS Press.

73. Y. A. Liu. Selectively caching intermediate results for incremental computation. In *Proceedings of the 4th International Conference for Young Computer Scientists*, pages 367–374, Beijing, China, July 1995. Peking University Press.

74. Y. A. Liu and T. Teitelbaum. Caching intermediate results for program improvement. In *Proceedings of the ACM SIGPLAN Symposium on Partial Evaluation and Semantics-Based Program Manipulation*, pages 190–201, La Jolla, California, June 1995. ACM Press.

75. Y. A. Liu. Deriving incremental programs. In *Proceedings of the 3rd International Conference for Young Computer Scientists*, Beijing, China, July 1993. Tsinghua University Press.

76. Y. Liu, B. Zhang, and J. Wang. A formalized uncertainty reasoning model that combines qualitative partitions and quantitative descriptions in multi-factor combination problems. In *Proceedings of the 3rd International Fuzzy Systems Association World Congress*, Seattle, Washington, August 1989.

77. Y. Liu, B. Zhang, and J. Wang. The quantitative and qualitative inexactness and reasoning in multi-factor combination problems. In *Proceedings of the 1st International Symposium for Young Computer Scientists*, Beijing, China, August 1989. The Publishing House of Surveying and Mapping.

78. Y. Liu. The quantitative and qualitative inexactness and reasoning in multi-factor combination problems. In *Proceedings of the 1st Beijing Symposium for Young Computer Scientists*, Beijing, November 1988. Received Excellent Paper Award and selected for *Proceedings of the 2nd National Symposium for Young Computer Scientists*, Nanjing, May 1989. (In Chinese.)

79. Y. Liu. Analysis and study of multi-factor combination problems. In *Proceedings of 1st National Symposium on Intelligence Science*, Guangzhou, September 1988. Selected to print as full paper in *Jinan Natural Science and Medical Science Journal*, October 1988. (In Chinese.)

Other Conference Publications (Refereed Abstracts, Invited Papers, etc.)

80. Y. A. Liu, B. Lin, and S. D. Stoller. Programming and optimizing distributed algorithms: An overview. In *Proceedings of the 8th International Conference and Expo on Emerging Technologies for a Smarter World*, Hauppauge, New York, November 2–3, 2011. IEEE Press. (Refereed paper, but strangely missing in proceedings)

81. M. Gorbovitski and Y. A. Liu. Composition in Incrementalization and Application to Constrained RBAC. CEWIT 2008 International Conference on Wireless & Information Technologies, October 16, 2008. (Refereed abstract for poster)

82. M. Gorbovitski, T. Rothamel, Y. A. Liu, and S. D. Stoller. Implementing incrementalization across object abstraction. In *Conference Companion of the 20th ACM Conference Object-Oriented Programming, Systems, Languages, and Applications*, pages 116–117, San Diego, California, October 2005. ACM Press. (Refereed abstract for poster)

83. S. D. Stoller and Y. A. Liu. Security policy languages and enforcement. In *Proceedings of the 3rd Russian National Conference on Mathematics and Information Technology Security*, Moscow, Russia, October 2004. (Invited paper)

84. Y. A. Liu. Iterate, incrementalize, and implement: A systematic approach to efficiency improvement and guarantees. In *Proceedings of the 5th International Workshop on Implicit Computational Complexity*, volume 90(1) of *Electronic Notes in Theoretical Computer Science*, pages 45-47, Ottawa, Canada, June 2003. Elsevier. (Abstract for invited talk.)

85. R. Grosu, E. Zadok, S. A. Smolka, R. Cleaveland, and Y. A. Liu. High-confidence operating systems. In *Proceedings of the 10th ACM SIGOPS European Workshop on 'Can We Really Depend on an OS?'*, pages 205–208, Saint-Emilion, France, September 2002. ACM Press. (Refereed abstract)

86. Y. Zhang and Y. A. Liu. Automating derivation of incremental programs. In *Proceedings of the 1998 ACM SIGPLAN International Conference on Functional Programming*, page 350, Baltimore, Maryland, September 1998. ACM Press. (Refereed abstract for poster)

87. Y. A. Liu. Efficient computation via incremental computation. In *Proceedings of the International Symposium on Computing and Microelectronics Technologies*, pages 204–221, Peking University, Beijing, China, May 1998. Peking University Press. (Invited paper)

Miscellaneous Publications

88. D. Goyal and Y. A. Liu. Automated development of software for program analysis and transformation. *ACM SIGSOFT Software Engineering Notes*, 25(1), January 2000.
89. Y. A. Liu and S. D. Stoller. ETAPS'99 report. *ACM SIGPLAN Notices*, 34(6):16–17, June 1999, and *Bulletin of the EATCS*, 68:196–197, June 1999.

Technical Reports Not Published Elsewhere

90. K. T. Tekle and Y. A. Liu. Precise complexity guarantees for pointer analysis via Datalog with extensions. *ArXiv*, 1608.01594, August 2016.
91. Y. A. Liu and S. D. Stoller. The founded semantics and constraint semantics of logic rules. *ArXiv*, 1606.06269, June 2016.
92. S. Chand, Y. A. Liu, and S. D. Stoller. Formal verification of multi-Paxos for distributed consensus. *ArXiv*, 1606.01387, June 2016.
93. Y. A. Liu, S. D. Stoller, and B. Lin. From clarity to efficiency for distributed algorithms. *ArXiv*, 1412.8461, December 2014 (Revised January 2015 and January 2016).
94. Y. Liu and T. Wakayama. Incremental line-breaking algorithms. Technical Report, Xerox Webster Research Center, Webster, New York, August 1992.
95. Y. Liu, B. Zhang, and J. Wang. CT-MI: An expert system model for multi-factor combination problems, Technical Report, Department of Computer Science, Tsinghua University, Beijing, December 1989.

SOFTWARE SYSTEMS

- B. Lin and Y. A. Liu. DistAlgo—A very high-level language for distributed algorithms. Stony Brook U., 2010–present.
- J. Brandvein, Y. A. Liu, and B. Lin. IncOQ—A system for generating efficient implementations of demand-driven incremental object queries. Stony Brook U., 2010-present.
- M. Gorbovitski and Y. A. Liu. InvTS—A system for invariant-driven transformations. Stony Brook U., 2005–present.
- Y. A. Liu and T. Tekle. A system for generating efficient implementations from Datalog rules and graph queries. Stony Brook U., 2003–present.
- Y. A. Liu, J. Liu, C. Yu, M. Fazylova, B. Yim, and K. Jakkula. MultiRunner and MultiQuery—Assistants for traversing links and answering queries on the Web. Stony Brook U., 2001–2006.
- G. Gomez and A. Liu. ALPA—Automatic language-based performance analysis tools that generate accurate worst-case running time and space usage given bounds on input sizes. Indiana University and Stony Brook U., 1997–2006.
- Y. A. Liu and N. Li. CACHET continued—A program transformation system for drastic program optimization based on automated incrementalization and powerful program analysis. Indiana University, 1997–2000.
- Y. A. Liu. CACHET—An incremental-attribution-based interactive system that uses systematic program analysis and transformation techniques to derive efficient incremental programs. Cornell University, 1993–1996.

- Y. A. Liu. OGGEB—An expert system for evaluation of oil and gas generation in basins, with interfaces to several graphical tools for geochemical analysis. Research Institute of Petroleum Exploration and Development (CD-RIED) and Tsinghua University, Beijing, 1988–1990.

PATENTS

- Y. A. Liu and T. Wakayama. Incremental algorithms for optimal line-breaking in text layout. U.S. Patent No. 6647533. Issued November 11, 2003.

LECTURES

Presentations at Conferences and Workshops

1. Distributed programming and ubiquitous programming. The 7th IEEE Annual Ubiquitous Computing, Electronics & Mobile Communication Conference, New York City, October 21, 2016. (Keynote Address.)
2. DistAlgo: A language for distributed algorithms. The 72th Meeting of IFIP Working Group 2.1, Bennington, Vermont, December 12, 2014.
3. Programming distributed algorithms, with Scott Stoller and Bo Lin. Fall School Tutorials, the 2014 ACM SIGPLAN Conference on Systems, Programming, Languages and Applications: Software for Humanity, Portland, Oregon, October 24, 2014. (Tutorial, 90 minutes.)
4. Programming distributed algorithms, with Scott Stoller and Bo Lin. The 19th ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming, Orlando, Florida, February 15, 2014. (Tutorial, 180 minutes.)
5. Incrementalization: from clarity to efficiency. Joint Mathematics Meetings AMS Special Session on Mathematical Underpinnings of Multivariate Complexity Theory and Algorithm Design, and Its Frontiers AND the Field of Incrementalization. San Diego, California, January 11, 2013.
6. From clarity to efficiency for distributed algorithms. The 27th ACM SIGPLAN Conference on Object-Oriented Programming, Systems, Languages and Applications, Tucson, Arizona, October 24, 2012.
7. From clarity to efficiency for distributed algorithms. The 69th Meeting of IFIP Working Group 2.1, Ottawa, Canada, October 10, 2012.
8. High-level executable specifications of distributed algorithms. The 14th International Symposium on Stabilization, Safety, and Security of Distributed Systems, Toronto, Canada, October 2, 2012.
9. From clarity to efficiency for distributed algorithms. ACM SIGPLAN Workshop on Languages for Distributed Algorithms, Philadelphia, Pennsylvania, January 23, 2012.
10. Programming and optimizing distributed algorithms. The 8th International Conference and Expo on Emerging Technologies for a Smarter World, Hauppauge, New York, November 3, 2011.
11. Languages for distributed algorithms. The Inaugural Meeting of Working Group on Language Design, Mountain View, California, June 2, 2011.

12. Programming and optimizing distributed algorithms. The 66th Meeting of IFIP Working Group 2.1, Atlantic City, New Jersey, September 21, 2010.
13. Systematic program design: from clarity to efficiency. Dagstuhl Seminar on Software Synthesis, Leibniz Center for Informatics, Schloss Dagstuhl, Germany, December 10, 2009.
14. A language and framework for invariant-driven transformations. The 8th International Conference on Generative Programming and Component Engineering, Denver, Colorado, October 4, 2009.
15. Analysis and transformations for efficient query-based debugging. The 8th IEEE International Working Conference on Source Code Analysis and Manipulation, Beijing, China, September 28, 2008.
16. From clear specifications to efficient implementations. The 62nd Meeting of IFIP Working Group 2.1, Namur, Belgium, December 10, 2006.
17. Querying complex graphs. The 8th International Symposium on Practical Aspects of Declarative Languages, Charleston, South Carolina, January 10, 2006.
18. Core role-based access control: Efficient implementations by transformations. ACM SIGPLAN 2006 Workshop on Partial Evaluation and Semantics-Based Program Manipulation, Charleston, South Carolina, January 10, 2006.
19. Incrementalization across object abstraction. The 20th ACM Conference Object-Oriented Programming, Systems, Languages, and Applications, San Diego, California, October 20, 2005.
20. Incrementalization across object abstraction. The 60th Meeting of IFIP Working Group 2.1, Monterey Bay, California, May 25, 2005.
21. Querying complex graphs. The 60th Meeting of IFIP Working Group 2.1, Monterey Bay, California, May 25, 2005.
22. From rules to efficient programs with time and space guarantees. The 5th ACM SIGPLAN International Conference on Principles and Practice of Declarative Programming, Uppsala, Sweden, August 28, 2003.
23. Iterate, incrementalize, and implement: A systematic approach to efficiency improvement and guarantees. The 5th International Workshop on Implicit Computational Complexity, Ottawa, Canada, June 27, 2003. (Invited Talk.)
24. Optimizing Ackermann's function by incrementalization. ACM SIGPLAN Workshop on Partial Evaluation and Semantics-Based Program Manipulation, San Diego, California, June 7, 2003.
25. Analysis of object queries and updates. The 57th Meeting of IFIP Working Group 2.1, New York City, April 3, 2003.
26. From rules to analysis programs with time and space guarantees. The 57th Meeting of IFIP Working Group 2.1, New York City, March 30, 2003.
27. Analysis of object queries and updates. Dagstuhl Seminar on Program Analysis for Object-Oriented Evolution, International Conference and Research Center for Computer Science, Schloss Dagstuhl, Germany, February 27, 2003.

28. Solving regular path queries. The 6th International Conference on Mathematics of Program Construction, Schloss Dagstuhl, Germany, July 8, 2002.
29. Program optimization using indexed and recursive data structures. ACM SIGPLAN Workshop on Partial Evaluation and Semantics-Based Program Manipulation, Portland, Oregon, January 15, 2002.
30. From recursion to iteration: What are the optimizations? The 56th Meeting of IFIP Working Group 2.1, Ameland, The Netherlands, September 10, 2001.
31. Solving regular tree grammar based constraints. The 8th International Static Analysis Symposium, Paris, France, July 17, 2001.
32. Systematic and powerful program optimization by incrementalization. The 38th Meeting of IFIP Working Group 2.4, San Miniato, Italy, May 29, 2001.
33. From recursion to iteration: What are the optimizations? Dagstuhl Seminar on Code Optimisation: Trends, Challenges and Perspectives, International Conference and Research Center for Computer Science, Schloss Dagstuhl, Germany, September 20, 2000.
34. Automatic accurate time-bound analysis for high-level languages. The 54th Meeting of IFIP Working Group 2.1, London, U.K., April 5, 2000.
35. Loop optimization for aggregate array computations. The 53rd Meeting of IFIP Working Group 2.1, Potsdam, Germany, June 18, 1999.
36. Dynamic programming via static incrementalization. The 8th European Symposium on Programming, Amsterdam, The Netherlands, March 24, 1999.
37. Dependence analysis for recursive data. The 52nd Meeting of IFIP Working Group 2.1, Quebec City, Canada, October 6, 1998.
38. Loop optimization for aggregate array computations. IEEE International Conference on Computer Languages, Chicago, Illinois, May 16, 1998.
39. Dependence analysis for recursive data. IEEE International Conference on Computer Languages, Chicago, Illinois, May 15, 1998.
40. Efficient computation via incremental computation. International Symposium on Computing and Microelectronics, Peking University, Beijing, China, May 3, 1998.
41. Dynamic programming via static incrementalization. The 51st Meeting of IFIP Working Group 2.1, Magdalen College, Oxford, U.K., January 6, 1998.
42. What does cache-and-prune give us? The 50th Meeting of IFIP Working Group 2.1, Le Bischenberg, Alsace, France, February 20, 1997.
43. Principled strength reduction. IFIP Working Conference on Algorithmic Languages and Calculi, Le Bischenberg, Alsace, France, February 19, 1997.
44. Incrementalization: A general systematic approach for efficiency improvement. The 49th Meeting of IFIP Working Group 2.1, Rancho Santa Fe, California, June 11, 1996.
45. Discovering auxiliary information for incremental computation. The 23rd Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, St. Petersburg Beach, Florida, January 23, 1996.

46. CACHET: A system for deriving incremental programs. The 10th Knowledge-Based Software Engineering Conference, Boston, Massachusetts, November 13, 1995.
47. Selectively caching intermediate results for incremental computation. The 4th International Conference for Young Computer Scientists, Beijing, China, July 19, 1995.
48. Caching intermediate results for program improvement. ACM SIGPLAN Symposium on Partial Evaluation and Semantics-Based Program Manipulation, La Jolla, California, June 23, 1995.
49. Systematic derivation of incremental programs. Dagstuhl Seminar on Incremental Computation and Dynamic Algorithms, International Conference and Research Center for Computer Science, Schloss Dagstuhl, Germany, May 5, 1994.
50. Deriving incremental programs. The 3rd International Conference for Young Computer Scientists, Beijing, China, July 15, 1993.

Demonstrations of Software at Conferences and Workshops

51. DistAlgo, part of tutorial, Programming distributed algorithms, with Scott Stoller and Bo Lin. Fall School Tutorials, the 2014 ACM SIGPLAN Conference on Systems, Programming, Languages and Applications: Software for Humanity, Portland, Oregon, October 24, 2014.
52. DistAlgo, part of tutorial, Programming distributed algorithms, with Scott Stoller and Bo Lin. The 19th ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming, Orlando, Florida, February 15, 2014.
53. InvTS: A system for invariant-driven transformations. Dagstuhl Seminar on Software Synthesis, Leibniz Center for Informatics, Schloss Dagstuhl, Germany, December 11, 2009.
54. CACHET: A system for deriving incremental programs. The 10th Knowledge-Based Software Engineering Conference, Boston, Massachusetts, November 14, 1995.
55. CACHET: A system for deriving incremental programs. Dagstuhl Seminar on Incremental Computation and Dynamic Algorithms, International Conference and Research Center for Computer Science, Schloss Dagstuhl, Germany, May 2–6, 1994.

Invited Talks and Lectures at Universities and Research Institutes

56. From clarity to efficiency for distributed algorithms. Kestrel Institute, Palo Alto, California, July 24, 2015.
57. Incrementalization: from clarity to efficiency. The Graduate Center, City University of New York, April 3, 2014.
58. High-level executable specifications of distributed algorithms. The 2013 IBM Programming Languages Day, IBM Watson Research Center, Yorktown Heights, New York, September 23, 2013.
59. From clarity to efficiency for distributed algorithms. Laboratoire d'Informatique de Paris 6, University Pierre and Marie Curie, Paris, France, March 27, 2013.
60. From clarity to efficiency for distributed algorithms. The 2012 IBM Programming Languages Day, IBM Watson Research Center, Hawthorne, New York, June 28, 2012.

61. Systematic program design: From clarity to efficiency. Kestrel Institute, Palo Alto, California, June 6, 2011.
62. Alias analysis for optimization of dynamic languages. New England Programming Languages Symposium, Yale University, New Haven, Connecticut, April 29, 2010.
63. A language and framework for invariant-driven transformations. Department of Computer Science, University of Toronto, Toronto, Canada, July 20, 2009.
64. Systematic Design: From clear specifications to efficient implementations. Institute of Software, School of Electronics Engineering and Computer Science, Peking University, Beijing, China, September 17, 2008.
65. Systematic design: From clear specifications to efficient implementations. Institute of Software, Academy of Sciences, Beijing, China, August 26, 2008.
66. Systematic design: From clear specifications to efficient implementations. Microsoft Research Asia, Beijing, China, July 14, 2008.
67. Systematic design: From clear specifications to efficient implementations. Department of Informatics, School of Mathematical Sciences, Peking University, Beijing, China, September 17, 2007.
68. Core role-based access control: efficient implementations by transformations. School of Information, Renmin University, Beijing, China, September 17, 2007.
69. Systematic design: From clear specifications to efficient implementations. Computer Science and Artificial Intelligence Laboratory, Massachusetts Institute of Technology, Boston, Massachusetts, June 21, 2007.
70. Role-based access control: Specification and implementation. Department of Informatics, School of Mathematical Sciences, Peking University, Beijing, China, September 16, 2005.
71. Incrementalization across object abstraction. Microsoft Research Asia, Beijing, China, July 20, 2005.
72. Incrementalization across object abstraction. Institute of Software, Academy of Sciences, Beijing, China, June 24, 2005.
73. Design by building up and breaking through abstractions. Computer Science Department, Tsinghua University, Beijing, China, October 13, 2004.
74. Parametric regular path queries. Department of Informatics, School of Mathematical Sciences, Peking University, Beijing, China, October 12, 2004.
75. Parametric regular path queries. School of Computer Science, University of Waterloo, Waterloo, Canada, June 18, 2004.
76. Iterate, incrementalize, and implement: A systematic approach to efficiency improvement and guarantees. Institute of Software, Academy of Sciences, Beijing, China, October 9, 2003.
77. Iterate, incrementalize, and implement: A systematic approach to efficiency improvement and guarantees. Computer Science Department, Peking University, Beijing, China, October 8, 2003.
78. Iterate, incrementalize, and implement: A systematic approach to efficiency improvement and guarantees. Computer Science Department, Tsinghua University, Beijing, China, October 8, 2003.

79. From rules to analysis programs with time and space guarantees. Department of Informatics, School of Mathematical Sciences, Peking University, Beijing, China, September 24, 2003.
80. From rules to analysis programs with time and space guarantees. The 4th IBM Programming Languages Day, IBM Watson Research Center, Hawthorne, New York, April 25, 2003.
81. Solving regular path queries. New England Programming Languages Symposium, Yale University, New Haven, Connecticut, August 7, 2002.
82. Program optimization using indexed and recursive data structures. New Jersey Programming Languages Seminar, Avaya Labs, Basking Ridge, New Jersey, February 22, 2002.
83. Systematic and powerful program optimization by incrementalization. Department of Computer Science, New York University, New York City, New York, December 7, 2000.
84. Incrementalization: A powerful approach to efficiency improvement. Department of Computer Science, Butler University, Indianapolis, Indiana, April 21, 2000.
85. Incrementalization: A powerful approach to efficiency improvement. Computer Science Department, State University of New York at Stony Brook, Stony Brook, New York, March 20, 2000.
86. Incrementalization: A powerful approach to efficiency improvement. Bell Labs, Lucent Technologies, Naperville, Illinois, November 18, 1999.
87. Loop optimization for aggregate array computations. BRICS, Department of Computer Science, University of Aarhus, Denmark, June 29, 1999.
88. Dynamic programming via static incrementalization. BRICS, Department of Computer Science, University of Aarhus, Denmark, June 29, 1999.
89. Incrementalization: A powerful approach to efficiency improvement. BRICS, Department of Computer Science, University of Aarhus, Denmark, June 28, 1999.
90. Automatic accurate time-bound analysis for high-level languages. DIKU, Department of Computer Science, University of Copenhagen, Denmark, June 23, 1999.
91. Dynamic programming via static incrementalization. DIKU, Department of Computer Science, University of Copenhagen, Denmark, June 22, 1999.
92. Incrementalization: A powerful approach to efficiency improvement. DIKU, Department of Computer Science, University of Copenhagen, Denmark, June 21, 1999.
93. Dependence analysis for recursive data. Département de Mathématiques et d'Informatique, Ecole Normale Supérieure, Paris, France, March 19, 1999.
94. Dependence analysis for recursive data. IBM Thomas J. Watson Research Center, Yorktown Heights, New York, August 21, 1998.
95. Dependence analysis for recursive data. Department of Computer Science, University of Waterloo, Waterloo, Canada, June 22, 1998.
96. Incrementalization: A general systematic approach for efficiency improvement. Department of Computer Science, Graduate School of University of Science and Technology of China, Academy of Sciences, Beijing, China, May 8, 1998.
97. Incrementalization: A general systematic approach for efficiency improvement. Department of Computer Science, City University of Hong Kong, Hong Kong, May 16, 1997.

98. Incrementalization: A general systematic approach for efficiency improvement. Software Technology Center, Motorola Land Mobile Products Sector, Schaumburg, Illinois, December 6, 1996.
99. Incrementalization: A general systematic approach for efficiency improvement. Department of Computer Science and Engineering, Pennsylvania State University, University Park, Pennsylvania, May 2, 1996.
100. Incrementalization: A general systematic approach for efficiency improvement. Department of Computer and Information Science, University of Pennsylvania, Philadelphia, Pennsylvania, April 30, 1996.
101. Incrementalization: A general systematic approach for efficiency improvement. Department of Computer Science, University of Waterloo, Waterloo, Canada, April 15, 1996.
102. Incrementalization: A general systematic approach for efficiency improvement. Department of Computer Sciences, Purdue University, West Lafayette, Indiana, April 1, 1996.
103. Incrementalization: A general systematic approach for efficiency improvement. Department of Computer Science, New York University, New York City, New York, March 26, 1996.
104. Incrementalization: A general systematic approach for efficiency improvement. AT&T Bell Labs, Murray Hill, New Jersey, March 21, 1996.
105. Incrementalization: A general systematic approach for efficiency improvement. Computer Science Department, Indiana University, Bloomington, Indiana, March 18, 1996.
106. Incrementalization: A general systematic approach for efficiency improvement. Information Sciences Institute, University of Southern California, Los Angeles, California, March 4, 1996.
107. Efficient computation via incremental computation. Ph.D. Super-Colloquium, Department of Computer Science, Cornell University, Ithaca, New York, February 1, 1996.
108. Systematic derivation of incremental programs. Kestrel Institute, Palo Alto, California, July 1, 1994.
109. Automatic derivation of incremental programs. System Science Laboratory, Xerox Webster Research Center, Webster, New York, July 6, 1992.
110. Dynamic and static incremental attribute evaluation algorithms. System Science Laboratory, Xerox Webster Research Center, Webster, New York, June 10–11, 1992.

Other Presentations

111. Microlabs in Computer Science. Workshop for an NSF DUE Project, Suffolk County Community College, November 21, 2015.
112. Microlabs in Computer Science. Workshop for an NSF DUE Project, Pace University, November 15, 2014.
113. Starting Your Job and Planning Your Career in Academia. The CRA-W 2006 Grad Cohort Workshop, San Francisco, California, April 1, 2006. (Invited Speaker.)

PROFESSIONAL ACTIVITIES

Member

1. IFIP Working Group 2.1, Algorithmic Languages and Calculi, since January 1998.

Member of Editorial Board

2. ACM Books Editorial Board, Area Editor responsible for Programming Languages, since October 2016.

Conference Chair and Organization Roles

3. Co-Chair, with David Warren, Workshop on Applications of Logic Programming, New York City, New York, October 17, 2016.
4. Co-Organizer, with Michael Kifer, Symposium on Logic Programming: Systems and Applications, Stony Brook, New York, September 21–22, 2012.
5. Co-Chair, with Andrew Black, ACM SIGPLAN Workshop on Languages for Distributed Algorithms, Philadelphia, Pennsylvania, January 23–24, 2012.
6. Steering Committee Member, ACM SIGPLAN Workshop and Symposium on Partial Evaluation and Semantics-Based Program Manipulation, 2005–2011.
7. Steering Committee Member at Large, ACM SIGPLAN/SIGBED Conference on Languages, Compilers, and Tools for Embedded Systems, 2004–2009.
8. Finance Chair, ACM SIGPLAN Symposium on Languages, Compilers, and Tools for Embedded Systems, San Diego, California, June 11–13, 2003.
9. Co-Local Organizer, with Robert Dewar, The 57th Meeting of IFIP Working Group 2.1 on Algorithmic Languages and Calculi, New York City, New York, March 30–April 3, 2003.
10. Steering Committee Member, ACM SIGPLAN Symposium on Languages, Compilers, and Tools for Embedded Systems, 2001–2004.
11. Steering Committee Member, ACM SIGPLAN Workshop on Languages, Compilers, and Tools for Embedded Systems, 1999–2001.
12. Co-Chair, with Reinhard Wilhelm, ACM SIGPLAN Workshop on Languages, Compilers, and Tools for Embedded Systems, Atlanta, Georgia, May 5, 1999.

Member of Scientific Committee

13. Summer School on Generative and Transformational Techniques in Software Engineering, Braga, Portugal, July 4–8, 2005.

Member of Program Committee

14. The 8th International Conference on Computational Logics, Algebras, Programming, Tools, and Benchmarking, Athens, Greece, February 19–23, 2017.
15. ACM SIGPLAN 2017 Workshop on Partial Evaluation and Program Manipulation, Paris, France, January 16–17, 2017.
16. The 45th International Conference on Parallel Processing, Philadelphia, Pennsylvania, August 16–19, 2016.
17. The 13th Asian Symposium on Programming Languages and Systems, Pohang, Korea, November 30–December 2, 2015.

18. ACM SIGAda Annual International Conference on High Integrity Language Technology, Portland, Oregon, October 18–19, 2014.
19. The 24th International Symposium on Logic-Based Program Synthesis and Transformation, Canterbury, United Kingdom, September 10–11, 2014.
20. The 30th International Conference on Logic Programming, Vienna, Austria, July 19–22, 2014.
21. ACM SIGAda Annual International Conference on High Integrity Language Technology, Pittsburgh, Pennsylvania, November 10–14, 2013.
22. The 15th International Symposium on Principles and Practice of Declarative Programming, Madrid, Spain, September 16–18, 2013.
23. The 23nd European Symposium on Programming, Rome, Italy, March 16–24, 2013.
24. ACM SIGPLAN 2013 Workshop on Partial Evaluation and Program Manipulation, Rome, Italy, January 20–21, 2013.
25. The 11th International Conference on Generative Programming and Component Engineering, Dresden, Germany, September 26–28, 2012.
26. The 2nd Workshop on the Resurgence of Datalog in Academia and Industry, Datalog 2.0, Vienna, Austria, September 11–13, 2012.
27. ACM SIGPLAN/SIGBED 2012 Conference on Languages, Compilers, Tools and Theory for Embedded Systems, Beijing, China, June 12–13, 2012.
28. The 10th International Conference on Generative Programming and Component Engineering, Portland, Oregon, October 23–24, 2011.
29. The 21st International Symposium on Logic-Based Program Synthesis and Transformation, Odense, Denmark, July 18–20, 2011.
30. The 14th International Workshop on Software and Compilers for Embedded Systems, Schloss Rheinfels, Germany, June 27–28, 2011.
31. ACM SIGPLAN/SIGBED 2011 Conference on Languages, Compilers, Tools and Theory for Embedded Systems, Chicago, Illinois, April 12–14, 2011.
32. The 20th International Conference on Compiler Construction, Saarbrücken, Germany, March 26–April 4, 2011.
33. The 20th International Symposium on Logic-Based Program Synthesis and Transformation, Hagenberg, Austria, July 23–25, 2010.
34. The 13th International Workshop on Software and Compilers for Embedded Systems, Rheinfels, Germany, June 29–30, 2010.
35. ACM SIGPLAN 2010 Workshop on Partial Evaluation and Program Manipulation, Madrid, Spain, January 18–19, 2010.
36. The 2nd International Workshop on Cyber-Physical Systems, Montreal, Canada, June 22, 2009.
37. The 12th International Workshop on Software and Compilers for Embedded Systems, Nice, France, April 23–24, 2009.
38. International Conference on Compiler Construction, York, United Kingdom, March 22–29, 2009.

39. The 24th Annual ACM Symposium on Applied Computing, Programming Languages Track, Honolulu, Hawaii, March 8–12, 2009.
40. The 2008 IEEE/IFIP International Conference On Embedded and Ubiquitous Computing, Shanghai, China, December 17–20, 2008.
41. ACM SIGPLAN/SIGBED 2008 Conference on Languages, Compilers, and Tools for Embedded Systems, Tucson, Arizona, June 12–13, 2008.
42. The 11th International Workshop on Software and Compilers for Embedded Systems, Munich, Germany, March 13–14, 2008.
43. The 5th Asian Symposium on Programming Languages and Systems, Singapore, November 29–December 1, 2007.
44. ACM SIGPLAN/SIGBED 2007 Conference on Languages, Compilers, and Tools for Embedded Systems, San Diego, California, June 13–15, 2007.
45. The 10th International Workshop on Software and Compilers for Embedded Systems, Nice, France, April 20, 2007.
46. ACM SIGPLAN 2007 Workshop on Partial Evaluation and Semantics-Based Program Manipulation, Nice, France, January 14–15, 2007.
47. ACM SIGPLAN/SIGBED 2006 Conference on Languages, Compilers, and Tools for Embedded Systems, Ottawa, Canada, June 14–16, 2006.
48. ACM SIGPLAN Workshop on Programming Languages and Analysis for Security, Ottawa, Canada, June 10, 2006.
49. The 9th International Workshop on Software and Compilers for Embedded Systems, Dallas, Texas, September 29–October 1, 2005.
50. ACM SIGPLAN/SIGBED 2005 Conference on Languages, Compilers, and Tools for Embedded Systems, Chicago, Illinois, June 15–17, 2005.
51. The 8th International Workshop on Software and Compilers for Embedded Systems, Amsterdam, The Netherlands, September 2–3, 2004.
52. ACM SIGPLAN 2004 Symposium on Partial Evaluation and Semantics-Based Program Manipulation, Verona, Italy, August 24–25, 2004.
53. The 7th International Workshop on Software and Compilers for Embedded Systems, Vienna, Austria, September 24–26, 2003.
54. The 8th ACM SIGPLAN International Conference on Functional Programming, Uppsala, Sweden, August 25–29 2003.
55. ACM SIGPLAN ASIAN Symposium on Partial Evaluation and Semantics-Based Program Manipulation, Aizu, Japan, September 12–14, 2002.
56. ACM SIGPLAN Joint Conference on Languages, Compilers, and Tools for Embedded Systems, and Software and Compilers for Embedded Systems, Berlin, Germany, June 19–21, 2002.
57. ACM SIGPLAN 2002 Conference on Programming Languages Design and Implementation, Berlin, Germany, June 17–19, 2002.

58. ACM SIGPLAN 2001 Workshop on Languages, Compilers, and Tools for Embedded Systems, Snowbird, Utah, June 22–23, 2001.
59. The 2nd International Symposium on Programs as Data Objects, Aarhus, Denmark, May 21–23, 2001.
60. ACM SIGPLAN-SIGSOFT 1999 Workshop on Program Analysis for Software Tools and Engineering, Toulouse, France, September 6, 1999.
61. The 5th International Conference for Young Computer Scientists, Nanjing, China, August 17–20, 1999.
62. ACM SIGPLAN 1999 Workshop on Languages, Compilers, and Tools for Embedded Systems, Atlanta, Georgia, May 5, 1999.
63. ACM SIGPLAN 1998 Workshop on Languages, Compilers, and Tools for Embedded Systems, Montréal, Canada, June 19–20 1998.
64. ACM SIGPLAN 1997 Symposium on Partial Evaluation and Semantics-Based Program Manipulation, Amsterdam, The Netherlands, June 12–13, 1997.

Conference Session Chair

65. The 23nd European Symposium on Programming, Rome, Italy, March 22, 2013.
66. ACM SIGPLAN Workshop on Partial Evaluation and Semantics-Based Program Manipulation, Charleston, South Carolina, January 9, 2006.
67. ACM SIGPLAN Symposium on Languages, Compilers, and Tools for Embedded Systems, San Diego, California, June 12, 2003.
68. ACM SIGPLAN Joint Conference on Languages, Compilers, and Tools for Embedded Systems, and Software and Compilers for Embedded Systems, Berlin, Germany, June 20, 2002.
69. ACM SIGPLAN 2001 Workshop on Languages, Compilers, and Tools for Embedded Systems, Atlanta, Georgia, June 22, 2001.
70. The 5th International Conference for Young Computer Scientists, Nanjing, China, August 19, 1999.
71. ACM SIGPLAN 1999 Workshop on Languages, Compilers, and Tools for Embedded Systems, Atlanta, Georgia, May 5, 1999.
72. ACM SIGPLAN 1998 Workshop on Languages, Compilers, and Tools for Embedded Systems, Montreal, Canada, June 20, 1998.
73. IEEE 1998 International Conference on Computer Languages, Chicago, Illinois, May 15, 1998.

Invited Participant

74. Workshop on New Directions in Software Technology, Adversarial Computing: Active Engagement of Software in Adversarial Environments, St. John, U.S. Virgin Islands, December 7–10, 2015.
75. Microsoft Faculty Summit, Redmond, Washington, July 14–15, 2014.
76. Joint Mathematics Meetings AMS Special Session on Mathematical Underpinnings of Multivariate Complexity Theory and Algorithm Design, and Its Frontiers AND the Field of Incrementalization. San Diego, California, January 11–12, 2013.

77. The 14th Annual Workshop on New Directions in Software Technology, Personal Digital Manufacturing, St. John, U.S. Virgin Islands, December 5–8, 2011.
78. The Inaugural Meeting of Working Group on Language Design, Mountain View, California, June 1–3, 2011.
79. The 13th Annual Workshop on New Directions in Software Technology, Trustworthy Health Information Systems, St. John, U.S. Virgin Islands, December 6–9, 2010.
80. Dagstuhl Seminar on Software Synthesis, Leibniz Center for Informatics, Schloss Dagstuhl, Germany, December 7-11, 2009.
81. The CRA-W 2006 Grad Cohort Workshop, San Francisco, California, March 31–April 1, 2006.
82. Dagstuhl Seminar on Program Analysis for Object-Oriented Evolution, International Conference and Research Center for Computer Science, Schloss Dagstuhl, Germany, February 24–28, 2003.
83. The 38th Meeting of IFIP Working Group 2.4 on Software Implementation Technology, San Miniato, Tuscany, Italy, May 27–June 1, 2001.
84. Dagstuhl Seminar on Code Optimisation: Trends, Challenges and Perspectives, International Conference and Research Center for Computer Science, Schloss Dagstuhl, Germany, September 17–22, 2000.
85. The 51st Meeting of IFIP Working Group 2.1 on Algorithmic Languages and Calculi, Magdalen College, Oxford, U.K., January 5–9, 1998.
86. NSF Infrastructure 97, University of Kentucky, Lexington, Kentucky, May 11–12, 1997.
87. The 50th Meeting of IFIP Working Group 2.1 on Algorithmic Languages and Calculi, Le Bischenberg, Alsace, France, February 20, 1997.
88. The 49th Meeting of IFIP Working Group 2.1 on Algorithmic Languages and Calculi, Rancho Santa Fe, California, June 10–14, 1996.
89. Dagstuhl Seminar on Incremental Computation and Dynamic Algorithms, International Conference and Research Center for Computer Science, Schloss Dagstuhl, Germany, May 2–6, 1994.

Member of Review Panel

90. Korea Science and Engineering Foundation (KOSEF)
91. National Science Foundation (NSF)

Referee/Reviewer

92. National Science Foundation (NSF)
93. Netherlands Organisation for Scientific Research (NWO)
94. Research Council of Norway (RCN)
95. Singapore Agency for Science, Technology & Research (A*STAR)
96. Addison-Wesley
97. MIT Press

98. Morgan Kaufmann
99. Prentice Hall
100. Springer
101. *ACM Transactions on Architecture and Code Optimization*
102. *ACM Transactions on Embedded Computing Systems*
103. *ACM Transactions on Information and System Security*
104. *ACM Transactions on Programming Languages and Systems*
105. *ACM Transactions on Software Engineering and Methodology*
106. *Data and Knowledge Engineering*
107. *Distributed Computing*
108. *Formal Methods in System Design*
109. *Fundamenta Informaticae*
110. *Higher-Order and Symbolic Computation*
111. *IEEE Internet Computing*
112. *IEEE Transactions on Parallel and Distributed Systems*
113. *IEEE Transactions on Software Engineering*
114. *Information and Computation*
115. *Information Processing Letters*
116. *Journal of Computer Science and Technology*
117. *Journal of Functional Programming*
118. *Journal of Systems and Software*
119. *Science of Computer Programming*
120. *Software: Practice and Experience*
121. *Theoretical Computer Science*
122. *Theory and Practice of Logic Programming*
123. ACM Conference on Computer and Communications Security
124. ACM SIGPLAN Conference on Languages, Compilers, and Tools for Embedded Systems
125. ACM SIGPLAN Conference on Programming Languages Design and Implementation
126. ACM SIGPLAN International Conference on Principles and Practice of Declarative Programming
127. ACM SIGPLAN Symposium on Partial Evaluation & Semantics-Based Program Manipulation
128. ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages
129. ACM SIGPLAN-SIGSOFT Workshop on Program Analysis for Software Tools & Engineering
130. ACM Symposium on Access Control Models and Technologies
131. Annual Symposium on the Theory of Computing

132. Asian Symposium on Programming Languages and Systems
133. European Conference on Object-Oriented Programming
134. European Symposium on Programming
135. IEEE Computer Security Foundations Symposium
136. International Conference on Aspect-Oriented Software Development
137. International Conference on Compiler Construction
138. International Conference on Formal Engineering Methods
139. International Conference on Functional Programming
140. International Conference on Logic Programming
141. International Conference on Mathematics of Program Construction
142. International Conference on Verification, Model Checking and Abstract Interpretation
143. International Conference for Young Computer Scientists
144. International Static Analysis Symposium
145. International Symposium on Automated Technology for Verification and Analysis
146. International Symposium on Symbolic and Algebraic Computation
147. International Symposium on Theoretical Aspects of Computer Science
148. Workshop on Bytecode Semantics, Verification, Analysis and Transformation

UNIVERSITY ACTIVITIES

- Chair of SUNY Korea Computer Science Chair Search Committee. Computer Science Dept., Stony Brook Univ., 2015–2016.
- Search Committee for Dean of College of Engineering and Applied Sciences, Stony Brook Univ., 2015.
- Chair of SUNY Korea Faculty Search Committee. Computer Science Dept., Stony Brook Univ, 2011–2012.
- Excellence in Scholarship and Creative Activities Selection Committee. Stony Brook Univ., 2010–2013.
- Songdo Committee, the CS committee for the SBU campus at Songdo, S. Korea. Stony Brook Univ., 2009–2011.
- Discretionary Salary Increases Committee. Computer Science Dept, Stony Brook Univ., 2008, 2009, 2010, 2013, 2016.
- Graduate Information Director. Computer Science Dept., Stony Brook Univ., 2007–2008.
- Qualifying Exam Committee, Coordinator for Software Areas. Computer Science Dept., Stony Brook Univ., 2007.
- Student Careers Committee. Computer Science Dept., Stony Brook Univ., 2006–present.
- Chair of Instructor Search Committee. Computer Science Dept., Stony Brook Univ., 2001–present.

- Faculty Judge for Graduate Research Conference. Computer Science Dept., Stony Brook Univ., 2001, 2004.
- Graduate Admission Committee. Computer Science Dept., Stony Brook Univ., 2000–2011, 2013–2014.
- Chair of Research Environment Committee. Computer Science Dept., Indiana Univ., 1998–99.
- Faculty Affairs / Merit Review Committee. Computer Science Dept., Indiana Univ., 1997–98.
- Algorithms Qualifying Exam Committee. Computer Science Dept., Indiana Univ., 1996–2000.
- Admission and Awards Committee. Computer Science Dept., Indiana Univ., 1996–2000.
- Computer Science Department Picnic Czar. Cornell Univ., Fall 1994.
- Expanding Your Horizons Program. Cornell Univ., November 1991, 1992, 1993.
- Numerous activities and leaderships at Peking Univ. and Tsinghua Univ., 1983–89.

UNIVERSITY COURSES TAUGHT

- CSE 591: Programming Complex Algorithms, Stony Brook U., Fall 2016.
- CSE 645: Seminar in Languages, Stony Brook U., Spring 2008–present.
- CSE 535: Asynchronous Systems, Stony Brook U., Fall 2012, Fall 2013, Fall 2015.
- CSE 393: Concurrent and Distributed Algorithms, Stony Brook U., Spring 2012, Spring 2015.
- CSE 690: Concurrent and Distributed Algorithms, Stony Brook U., Spring 2011, Spring 2012, Spring 2015.
- CSE 392: Data Mining, Stony Brook U., Fall 2014.
- CSE 307: Principles of Programming Languages, Stony Brook U., Spring 2007, Spring 2009, Spring 2011, Spring 2012, Spring 2014.
- CSE 614: Advanced Programming Languages, Stony Brook U., Spring 2008, Spring 2010, Fall 2013.
- CSE 373: Analysis of Algorithms, Stony Brook U., Fall 2012.
- CSE/ISE 308: Software Engineering, Stony Brook U., Fall 2001, Fall 2004, Spring 2010.
- CSE 592: Protocol Design and Analysis, Stony Brook U., Spring 2009.
- ITS 102: How to Solve Them, Stony Brook U., Spring 2007, Spring 2008, Spring 2009.
- CSE/ISE 315: Database Transaction Processing Systems, Stony Brook U., Spring 2006.
- CSE 690: Advanced Program Design and Optimization, Stony Brook U., Spring 2006.
- CSE 526: Principles of Programming Languages, Stony Brook U., Spring 2001, Spring 2005.
- CSE 591: Security Policy Frameworks, Stony Brook U., Spring 2005.
- CSE 391: Web Queries: Methods and Tools, Stony Brook U., Spring 2004.
- CSE 626: Advanced Programming Languages, Stony Brook U., Spring 2003.
- CSE 352: Artificial Intelligence, Stony Brook U., Spring 2003.

- CSE 532: Advanced Database Systems, Stony Brook U., Fall 2002.
- CSE/ISE 305: Principles of Database Systems, Stony Brook U., Spring 2002.
- Organizer, CSE 667: Design and Analysis Research Seminar, Stony Brook U., Fall 2001.
- CSE 675: Program Transformation and Program Analysis, Stony Brook U., Fall 2000.
- CSCI B629: Language-Based Algorithm Design and Analysis, Indiana U., Fall 1999.
- CSCI P523: Programming Language Implementation, Indiana U., Spring 1999.
- CSCI B629: Program Analysis and Program Transformation, Indiana U., Spring 1999.
- Organizer, Computer Science Department Colloquium, Indiana U., Fall 1998, Spring 1999.
- CSCI C212/A592: Introduction to Software Systems (using Java), Indiana U., Fall 1997, Fall 1998.
- CSCI B629: Language-Based Performance Analysis and Improvement, Indiana U., Spring 1998.
- CSCI B403: Introduction to Algorithm Design and Analysis, Indiana U., Spring 1997.
- CSCI B629: Program Transformation and Programming Environments, Indiana U., Spring 1997.
- Organizer, Programming Languages Seminar, Indiana U., Fall 1996.

SUPERVISORY ACTIVITIES

Ph.D. Students Research Supervision

- Saksham Chand. Formal verification of distributed algorithms. Stony Brook U., Spring 2016–present.
- Christopher Kane. High-level language abstractions for security. Stony Brook U., Summer 2015–present.
- Xuetian (Kain) Weng. Model checking of distributed algorithms. Stony Brook U., Fall 2012–present.
- Bo Lin (Ph.D., expected 2016). Compilation of a very high-level language for distributed algorithms. Stony Brook U., Fall 2009–present.
- Jonathan Brandvein (Ph.D., May 2016). Demand-driven incremental computation of object queries. Stony Brook U., Summer 2009–Spring 2016. (Joined Google NYC in May 2016.)
- Yury Puzis. Object-oriented modeling of a national electronic health records policy. Stony Brook U., Fall 2007–Fall 2008. (Continued in the Ph.D. program under Prof. I.V. Ramakrishnan in Spring 2009.)
- K. Tuncay Tekle (Ph.D., December 2010). *Efficient Datalog Queries with Time and Space Complexity Guarantees.* Stony Brook U., Fall 2006–Fall 2010. (Joined LogicBlox in Fall 2010. Became Adjunct Assistant Professor at Stony Brook University in November 2014.)
- Michael Gorbovitski (Ph.D., May 2011). *A System for Invariant-Driven Transformations.* Stony Brook U., Fall 2005–Spring 2010. (Joined Morgan Stanley in Summer 2010.)
- Katia Hristova (Ph.D., December 2007). *From Rules to Efficient Algorithms for Cyber Trust Applications.* Stony Brook U., Summer 2003–Fall 2007. (Became Assistant Professor at New York City College of Technology in January 2008. Became Co-founder at ThraceCode in August 2014.)

- Tom Rothamel (Ph.D., December 2008). *Automatic Incrementalization of Queries in Object-Oriented Programs*. Stony Brook U., Spring 2003–Fall 2008. (Started his own software company in Summer 2007. Was a part-time Post-Doc. at Stony Brook June 2009–September 2010.)
- Fuxiang (Sean) Yu. Querying graphs via extended automata. Stony Brook U., Summer 2001–Spring 2004. (Continued in the Ph.D. program under Prof. Ker-I Ko in Summer 2004.)
- Leena Unnikrishnan (Ph.D., December 2008). *Automatic Live Memory Bound Analysis for High-Level Languages*. Stony Brook U., Spring 2001–Fall 2008. (Joint supervision with Prof. Scott Stoller. Joined IBM Watson Research Center in Summer 2009.)
- Ning Li. Implementation of loop optimization for aggregate array computations. Indiana U., Summer 1999–Summer 2000. (Became a Ph.D. student at University of Wisconsin, Madison, in Fall 2000. Joined IBM Almaden Research Center in August 2001. Joined Facebook in 2009.)
- Gustavo Gómez (Ph.D., February 2007). *Automatic Time-Bound Analysis for High-Level Languages*. Indiana U., Summer 1997–Summer 2006. (Became a research staff at University of Colorado Boulder in January 2005. Joined Gambro BCT in May 2007. Joined Amazon.com in January 2008. Joined Rosetta Stone in March 2009. Joined Oracle in October 2013.)
- Yuchen Zhang. Automating incrementalization. Indiana U., Summer 1997–Spring 2004. (Joined Motorola Software Technology Center with M.S. in Spring 2000.)
- Byron Long. An algorithm for comparing deterministic regular tree grammars. Indiana U., Summer 1997. (Was a Ph.D. student of Prof. Daniel Leivant before and continued afterwards.)

Post-Doctoral Associates Research Supervision

- Tom Rothamel. Stony Brook U., June 2009–September 2010.

External Dissertation Examination Committee Member

- Chin Soon Lee (Ph.D., 2001). *Program Termination Analysis and the Termination of Offline Partial Evaluation*. University of Western Australia, June 2001.
- Deepak Goyal (Ph.D., 2000). *A Language Theoretic Approach to Algorithms*. New York University, October 1999.

Dissertation Committee Chair or Member

- Eric Papenhausen (Ph.D., 2016 expected). *Manual, Automatic, and Semi-Automatic Performance Optimization*, Stony Brook U.
- Pramod Ganapathi (Ph.D., 2016 expected). *Automatic Discovery of Efficient Divide-and-Conquer Algorithms for Dynamic Programming Problems*, Stony Brook U.
- Reza Basseda (Ph.D., 2015). *Planning with Transaction Logic*, Stony Brook U.
- Spyros Hadjichristodoulou (Ph.D., 2014). *Mode-Sensitive Type Analysis for Prolog Programs*, Stony Brook U.
- Senlin Liang (Ph.D., 2013). *Non-Termination Analysis and Cost-Based Query Optimization of Logic Programs*, Stony Brook U.

- Paul Fodor (Ph.D., 2011). *Practical Reasoning with Transaction Logic Programming for Knowledge Base Dynamics*, Stony Brook U.
- Hui Wan (Ph.D., 2010). *Belief Logic Programs*. Stony Brook U.
- Leena Unnikrishnan (Ph.D., 2008). *Automatic Live Memory Bound Analysis for High-Level Languages*. Stony Brook U.
- Rahul Agarwal (Ph.D., 2006). *Combining Static Analysis and Run-Time Analysis for Verification and Testing of Multi-Threaded Programs*. Stony Brook U.
- Diptikalyan Saha (Ph.D., 2006). *Incremental Evaluation of Tabled Logic Programs*. Stony Brook U.
- V.N. Venkatakrishnan (Ph.D., 2004). *Enforcement Techniques for Expressive Security Policies*. Stony Brook U.
- Luis Castro (Ph.D., 2003). *Demand-Based Evaluation of Tabled Logic Programs*. Stony Brook U.
- Guizhen Yang (Ph.D., 2002). *Inheritance in Object-Oriented Knowledge Bases*. Stony Brook U.
- Andrew Kinley (Ph.D., 2001). *Learning to Improve Case Adaption*. Indiana U.

Ph.D. Students Research Proficiency Exam Committee Chair or Member

- Mohammad Mahdi Javanmard. Towards a general framework for cache-efficient demand-driven dynamic programming. Stony Brook U., August 2014.
- Alireza Saberi. Static binary analysis. Stony Brook U., September 2011.
- Zhiquan Gao. A survey on agent research. Stony Brook U., September 2006.
- Amit Sasturkar. A survey of trust management. Stony Brook U., June 2005.
- Shengying Li. A survey on tools for binary code analysis. Stony Brook U., August 2004.
- Diptikalyan Saha. Incremental maintenance of recursive views with applications to tabled logic programming. Stony Brook U., September 2003.
- Rahul Agarwal. Detecting race conditions in multithreaded programs. Stony Brook U., September 2003.
- Wei Xu. Making C programs memory safe. Stony Brook U., February 2003.
- Saikat Mukherjee. A survey of formal approach to workflow modeling. Stony Brook U., August 2002.
- Prem Uppuluri. Pattern-matching based intrusion detection systems. Stony Brook U., August 2001.
- Lap Chung Lam. A survey of data breakpoint and reverse execution. Stony Brook U., January 2001.
- Luis Castro. On the computational integration of well-founded and stable model semantics. Stony Brook U., November 2000.

Master's Thesis and Research Project Supervision

- Jieao Zhu. A repository of distributed algorithms. Fall 2015-present.
- Saksham Chand. Formal verification of multi-Paxos using TLAPS. Stony Brook U., Summer 2015–Fall 2015.
- Zhenjin Wang. Visualization for distributed algorithms. Stony Brook U., Fall 2015–present.
- Andre Hamilton. Supporting access control in programming languages. Stony Brook U., Spring 2011–Spring 2012.
- Anu Kulkarni. Graph query and data query applications. Stony Brook U., Fall 2009–Fall 2010.
- Hiep Nguyen. *A Study on Access Control, Trust Management, Constraints and Types of a EHR System*. Master's Thesis, Stony Brook U., December 2010.
- Yogesh Srihari. A survey of games for teaching. Stony Brook U., Spring 2009.
- Andrew Gaun. Analysis and modeling of a rule-based distributed access control policy. Stony Brook U., Fall 2008–Summer 2009.
- Jeongwon Seo. Optimization of game programs. Stony Brook U., Spring–Fall 2008.
- Vaishali Kshatriy. Data translation between relational and object-oriented databases. Stony Brook U., Spring 2007–Spring 2008.
- Krupa Jakkula. A study of security frameworks and access control policies. Stony Brook U., Fall 2006–Spring 2007.
- Krupa Jakkula. Enhancements of an interactive Web query tool. Stony Brook U., Spring 2006.
- Fatima Zarinni. A system for planning clinical trials in drug development programs. Stony Brook U., Spring 2006–Spring 2007.
- Sen (Boris) Wu. Overview for modeling a nuclear power plant digital feedwater control system. Stony Brook U., Spring 2006.
- Brian Black. Methods for generating user interfaces. Stony Brook U., Fall 2005–Spring 2006.
- Chenghua Yan. Security issues in an interactive Web query tool. Stony Brook U., Spring 2005.
- Yevgen Borodin. Generating Web-based data acquisition tools from RelaxNG XML schemas. Stony Brook U., Fall 2004–Spring 2005.
- Yevgen Borodin. A constraint-based random relation generator. Stony Brook U., Summer 2004.
- Michael Gorbovitski. Very high-level languages and security applications. Stony Brook U., Spring 2004–Summer 2005.
- Byungchan Yim. Enhanced interface and implementation of a Web link traverser. Spring–Fall 2004.
- Gayathri Priyalakshmi. *Generating Efficient Programs for Solving Relational Database Queries*. Master's Thesis, Stony Brook U., August 2004.
- Mariya Fazylova. Enhanced methods and techniques for interactive Web queries. Stony Brook U., Summer 2003–Spring 2004.

- Siddhartha Singh. Debugging performance and transforming XML documents for a molecular viewer. Stony Brook U., Spring–Fall 2003.
- Tom Rothamel. Analysis and optimization of object-oriented programs. Stony Brook U., Summer–Fall 2002.
- Tom Rothamel. Motion controller scripting language. Stony Brook U., Spring–Summer 2002.
- Zongming Pan. Performance of a molecular viewer parsing XML documents. Stony Brook U., Fall 2001–Fall 2002.
- Jingyu Yan. Visual class machine language and code generation. Stony Brook U., Summer–Fall 2001.

Master's Thesis Committee Chair or Member

- Daniel Dean. *The Visual Development of GCC Plug-ins with GDE.* Stony Brook U., May 2009.
- Aditi Pandit. *Design and Applications of a High-Level Transparent Java Interface for Flora-2.* Stony Brook U., May 2005.
- Vishal C. Chowdhary. *Encapsulation and Persistent Storage Mechanisms for Flora-2 Modules.* Stony Brook U., August 2004.

Graduate Students Practical Training Supervision

- Xuetian (Kain) Weng. Verification of distributed systems. Google, Mountain View, California, May–Aug. 2016.
- Xuetian (Kain) Weng. Program optimization and verification. Google, Mountain View, California, May–Aug. 2015.
- Yuchen Zhang. Program transformation. Motorola Labs, Illinois, Feb. 2000–Jan. 2001.
- Yuchen Zhang. An investigation of compiler optimizations and compiler development tools for transport-triggered architectures. NEC Research, New Jersey, Jan. 2000–Feb. 2000.
- Chandrashekhar Shetty. LAURE system development. Bellcore, New Jersey, Summer 1997.

Bachelor's Honor Thesis Supervision

- Zhenjin Wang. *Modeling, analysis, and visualization of privacy policies.* Stony Brook U., May 2015.
- Youngseo Son. *Privacy policy specification with programming languages.* Stony Brook U., May 2015.
- Matthew Gruen. *Automating Performance Evaluation of Logical Queries.* Stony Brook U., May 2012.
- David Mazza. *Leveraging Dynamic Indexing for Static Rules via Program Rewriting.* Stony Brook U., May 2012. (Joint supervision with Profs. Michael Kifer and David Warren).
- Jintae Jang. *Minesweeper Solver.* Stony Brook U., May 2004.
- Hong (Christine) Yu. *MultiQuery: An Assistant for Interactive Web Queries.* Stony Brook U., May 2003.

- Rishi R. Sinha and Steven Sehgal. *Soft Real-Time Concurrent Class Machines.* Stony Brook U., May 2002.
- Jia (Jay) Liu. *MultiRunner: An Intelligent Link Traversing Agent.* Stony Brook U., May 2002.

Undergraduate Research Project Supervision

- Youngseo Son. Data analysis and visualization for social networks. Stony Brook U., Summer 2014-Spring 2015.
- Arjun Menon. Automatic object-oriented modeling of a national electronic health records policy. Stony Brook U., Spring 2011–Spring 2012.
- Andre Hamilton. Exploring distributed programming models. Stony Brook U., Fall 2009.
- James Taliento. Software vulnerability analysis. Stony Brook U., Summer 2009.
- Dennis Mok. Building and navigating Web graphs. Stony Brook U., 2003–2004.
- Sijia (Amanda) Chen. Portable collaborative bibliography repository. CRA Distributed Mentoring Affiliates Program, Stony Brook U., Summer 2002.

Intern Supervision

- Ling-Ling Zhang. Security policy analysis for a national electronic health record service. Ward Melville High School student research project, Summer 2009–Spring 2010. (Early action admission into Stanford University.)
- Kenny Wang. Creating a heart simulation program from a CellML model. College of Engineering and Applied Sciences Summer Research Institute, Stony Brook U., Summer 2006.
- Sandra Tinta. Schema-based data acquisition for Web applications. Alliance for Graduate Education and the Professoriate, Stony Brook U., Summer 2004.

UNIVERSITY LECTURES

- The founded semantics of logic rules and its efficient computation. Computer Science Dept., Stony Brook U., February 11, 2016.
- High-level executable specifications of distributed algorithms. Computer Science Dept., Stony Brook U., September 12, 2013.
- Incrementalization: From clarity to efficiency. Computer Science Dept., Stony Brook U., August 30, 2013.
- Programming and optimizing distributed algorithms. Computer Science Dept., Stony Brook U., July 28, 2011.
- From clear specifications to efficient implementations. Computer Science Dept., Stony Brook U., December 8, 2006.
- Design by building up and breaking through abstractions. Computer Science Dept., Stony Brook U., January 28, 2005.
- Iterate, incrementalize, and implement: A systematic approach to efficiency improvement and guarantees. Computer Science Dept., Stony Brook U., November 7, 2003.

- From rules to analysis programs with time and space guarantees. Computer Science Dept., Stony Brook U., April 23, 2003.
- Methods and techniques for incremental computation. Computer Science Dept., Stony Brook U., October 17, 2002.
- Program optimization using indexed and recursive data structures. Computer Science Dept., Stony Brook U., January 30, 2002.
- Systematic and powerful program optimization by incrementalization. Computer Science Dept., Stony Brook U., October 5, 2001.
- From recursion to iteration: What are the optimizations? Computer Science Dept., Stony Brook U., September 29, 2000.
- Loop optimization for aggregate array computations. Computer Science Dept., Indiana U., March 3, 1998.
- Dependence analysis for recursive data. Computer Science Dept., Indiana U., February 27, 1998.
- Principled strength reduction. Computer Science Dept., Indiana U., November 21, 1997.
- Incremental computation: methods, techniques, and applications. Computer Science Dept., Indiana U., January 24, 1997.
- Incrementalization for efficiency improvement: some logical accounts. IU Logic Group, Indiana U., January 22, 1997.
- Cache as cache can. Computer Science Dept., Indiana U., October 4, 1996.
- Efficient computation via incremental computation. Integration Program, Computer Science Dept., Indiana U., September 13, 1996.