CURRICULUM VITAE

THOMAS REPS **August 2026**

J. Barkley Rosser Professor & Rajiv and Ritu Batra Chair Emeritus
Computer Sciences Department
University of Wisconsin
1205 University Avenue
Madison, Wisconsin 53706
Birth: May 28, 1956 (Ithaca, NY USA)

(608) 262-2091 (Office)
(608) 262-1204 (Secretary)
reps@cs.wisc.edu
http://pages.cs.wisc.edu/~reps

Citizenship: United States

EDUCATION

1982	Ph.D., Cornell University	Computer Science
1982	M.S., Cornell University	Computer Science
1977	B.A., *cum laude*, Harvard University	Applied Mathematics

POSITIONS

2024–	Professor Emeritus, CS Dept., Univ. of Wisconsin
2007–08	Guest Professor, University of Paris 7, Paris, France
2000–01	Visiting Researcher, Consiglio Nazionale delle Ricerche (CNR), Pisa, Italy
1993–94	Guest Professor, Datalogisk Institut, University of Copenhagen, Copenhagen, Denmark
1990–93	Associate Chairman, Computer Sciences Department, University of Wisconsin
1988–2019	Co-founder and President, GrammaTech, Inc.
1985–2024	Professor, CS Dept., Univ. of Wisconsin (Asst.: 1985–88; Assoc.: 1988–94; Full: 1994–2024)
1984–85	Research Associate, Department of Computer Science, Cornell University
1982–83	Visiting Researcher, INRIA, Rocquencourt, France
1982–84	Post-Doctoral Associate, Department of Computer Science, Cornell University

AWARDS AND HONORS

2023	CAV Award (with A. Lal, M. Musuvathi, S. Qadeer, and J. Rehof)
2017	ACM SIGPLAN Programming Languages Achievement Award
2015	WARF Named Professorship, University of Wisconsin
2015	Rajiv and Ritu Batra Chair, Computer Sciences Department, University of Wisconsin
2014	#4 (field rating) and #7 (citations) on Microsoft Academic Search's list of most-highly-cited authors in Programming Languages (as of 8/27/2014)
2014	#13 (field rating) and #19 (citations) on Microsoft Academic Search's list of most-highly-cited authors in Software Engineering (as of 8/27/2014)
2013	Foreign member, Academia Europaea
2005	ACM Fellow
2003	Recognized as a "Highly Cited Researcher" in the field of Comp. Sci., Inst. for Scientific Inf.
2000	Guggenheim Fellowship, John Simon Guggenheim Memorial Foundation
2000	Humboldt Research Award, Alexander von Humboldt Foundation
1997	Vilas Associate Award, University of Wisconsin
1988	David and Lucile Packard Fellowship for Science and Engineering
1986	IBM Faculty Development Award
1986	NSF Presidential Young Investigator Award
1983	ACM Doctoral Dissertation Award

AWARD PAPERS

2020	PLDI Distinguished Paper Award (for Li, Y., Zhang, Q., and Reps, T., "Fast graph simplification for interleaved Dyck-reachability")
2011	ACM SIGSOFT Retrospective Impact Paper Award (for Reps, T., Horwitz, S., Sagiv, M., and Rosay, G., "Speeding up slicing," FSE 1994)

2010 ACM SIGSOFT Retrospective Impact Paper Award (for Reps, T. and Teitelbaum, T., "The Synthesizer Generator," Practical Software Development Environments, 1984)
2008 EAPLS Best-Paper Award at ETAPS ′08 (for Lim, J. and Reps, T., "A system for generating static analyzers for machine instructions," CC 2008)
2004 EAPLS Best-Paper Award at ETAPS ′04 (for Balakrishnan, G. and Reps, T., "Analyzing memory accesses in x86 executables," CC 2004)
2002 Horwitz, S., Reps, T., and Binkley, D., "Interprocedural slicing using dependence graphs" (PLDI 1988) selected as one of the 50 most influential papers from the ACM SIGPLAN Conference on Programming Language Design and Implementation (PLDI), 1979–99.
1982– 16 conference papers invited for special submission to journals

RESEARCH INTERESTS

Abstract interpretation (its applications to model checking; dataflow analysis; shape analysis; and the analysis of machine code, concurrent programs, probabilistic programs, and quantum programs; as well as its relationship to logic), program slicing, partial evaluation, computer security, program synthesis, quantum-circuit simulation and verification, code instrumentation and profiling, divided-difference arithmetic, incremental algorithms, and software-development environments.

For further information, see http://pages.cs.wisc.edu/~reps/past-research.html.

PUBLICATIONS

Books

Datta, A., Jha, S., Li, N., Melski, D., and Reps, T., *Analysis Techniques for Information Security*. Synthesis Lectures on Information Security, Privacy, and Trust, Morgan & Claypool Publishers, 2010.

Reps, T. and Teitelbaum, T., *The Synthesizer Generator: A System for Constructing Language-Based Editors*. Springer-Verlag, New York, NY, 1988.

Reps, T. and Teitelbaum, T., *The Synthesizer Generator Reference Manual: Third Edition*. Springer-Verlag, New York, NY, 1988.

Chinese reprint published by the World Publishing Corporation, Beijing, China, 1991.

Reps, T., *Generating Language-Based Environments*. The M.I.T. Press, Cambridge, MA, 1984.

Received the 1983 ACM Doctoral Dissertation Award.

Edited Books

Program Analysis and Compilation, Theory and Practice: Essays Dedicated to Reinhard Wilhelm, Reps, T., Sagiv, M., and Bauer, J. (eds.), Lecture Notes in Computer Science, Vol. 4444, Springer-Verlag, 2007.

Proceedings of the 27th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages (POPL) (Boston, Massachusetts, USA, January 19-21, 2000), M.N. Wegman and T.W. Reps (eds.) ACM, 2000.

Journal Publications

Yang, S., Ma, W., Reps, T., Tan, T., and Li, Y., Cut-shortcut whole-program pointer analysis: Re-imagining context-sensitivity without contexts. In *ACM Trans. on Program. Lang. and Syst. (TOPLAS)* (2026). DOI: 10.1145/3818615.

Yu, N., Palsberg, J. and Reps, T., SAQR-QC: A logic for scalable but approximate quantitative reasoning about quantum circuits. In *PACMPL 10(PLDI)*, 6 (2026), Article 206. DOI: 10.1145/3808284.

Pimpalkhare, N., Kincaid, Z., and Reps, T., Context-free-language reachability for almost-commuting transition systems. In *PACMPL 10(POPL)*, (2026). DOI: 10.1145/3776686.

Yu, N., Trinh, X.D., and Reps, T., Scalable equivalence checking and verification of shallow quantum circuits. In *PACMPL 9(OOPSLA2)* (2025).

Fang, R., Kincaid, Z., and Reps, T. Software model checking via summary-guided search. In *PACMPL 9(OOPSLA2)* (2025).

Kalita, P.K., Reps, T., and Roy, S., Automated abstract transformer synthesis for reduced product domains. In *ACM Trans. on Softw. Eng. and Methodology (TOSEM)* (2025), DOI: 10.1145/3733716.

Murphy, C., Johnson, K., Reps, T., and D'Antoni, L., Verifying solutions to semantics-guided synthesis problems. In *PACMPL 9(PLDI)* (2025).

Kim, J., Nagy, S., Reps, T., and D'Antoni, L., Semantics of sets of programs. In *PACMPL 9(OOPSLA1)* (2025).

Zhi, X., and Reps, T.W., Polynomial bounds of CFLOBDDs against BDDs. In *ACM Trans. on Program. Lang. and Syst. (TOPLAS) 47*(2): 4:1–4:36 (2025). DOI: 10.1145/3716313.

Also presented as a journal-first paper at the Int. Conf. on Object-Oriented Programming Systems, Languages, and Application (OOPSLA24-25).

Sistla, M., Chaudhuri, S., and Reps, T., Weighted context-free-language ordered binary decision diagrams. In *PACMPL 8(OOPSLA2)*, 1390–1419 (2024).

Johnson, K.J.C., Krishnan, R., Reps, T., and D'Antoni, L., Automating pruning in top-down enumeration for program synthesis problems with monotonic semantics. In *PACMPL 8(OOPSLA2)*, 935–961 (2024).

Liu, J., Murphy, C., Grover, A., Johnson, K., Reps, T., D'Antoni, L., Synthesizing formal semantics from executable interpreters. In *PACMPL 8(OOPSLA2)*, 362–388 (2024).

Nagy, S., Kim, J., Reps, T., and D'Antoni, L., Automating unrealizability logic: Hoare-style proof synthesis for infinite sets of programs. In *PACMPL 8(OOPSLA2)*, 113–139 (2024).

Wang, D. and Reps, T., Newtonian program analysis of probabilistic programs. In *PACMPL (OOPSLA1)* 8, 305–333, 2024. DOI: 10.1145/3649822.

Sistla, M., Chaudhuri, S., and Reps, T., CFLOBDDs: Context-free-language ordered binary decision diagrams. In *ACM Trans. on Program. Lang. and Syst. (TOPLAS) 46*(2): 7:1–7:82 (2024). DOI: 10.1145/3651157. (Extended version: arXiv:2211.06818, Nov. 2022; revised May 2024.)

Also presented as a journal-first paper at the ACM Conference on Programming Language Design and Implementation (PLDI), 2024.

Park, K., D'Antoni, L., and Reps, T., Synthesizing specifications. In *PACMPL 7(OOPSLA2)*, 1787–1816 (2023).

Li, Y., Zhang, Q., and Reps, T., Single-source-single-target interleaved-Dyck reachability via integer linear programming. In *PACMPL 7(POPL)*, 1003–1026 (2023).

Kim, J., D'Antoni, L., and Reps, T., Unrealizability logic. In *PACMPL 7(POPL)*, 659–688 (2023).

Kalita, P., Muduli, S., D'Antoni, L., Reps, T., and Roy, S., Synthesizing abstract transformers. In *PACMPL (OOPSLA)*, Dec. 2022.

Li, Y., Zhang, Q., and Reps, T., Fast graph simplification for interleaved Dyck-reachability. In *ACM Transactions on Programming Languages and Systems (TOPLAS) 44*(2): 11:1–11:28 (2022).

Kim, J., Hu, Q., D'Antoni, L., and Reps, T., Semantics-guided synthesis. In *PACMPL 5(POPL)* (2021).

Li, Y., Zhang, Q., and Reps, T., On the complexity of bidirected interleaved Dyck-reachability. In *PACMPL 5(POPL)* (2021).

Liu, P., Wahl, T., and Reps, T., Interprocedural context-unbounded program analysis using observation sequences. In *ACM Trans. on Program. Lang. and Syst. (TOPLAS) 42*(4): 16:1-16:34 (2021).

Sharma, T. and Reps, T., A new abstraction framework for affine transformers. In *Formal Methods in System Design (FMSD) 54*(1): 110-143 (2019).

Cyphert, J., Breck, J., Kincaid, Z., and Reps, T., Refinement of path expressions for static analysis. In *PACMPL 3(POPL)* (2019).

Kincaid, Z., Breck, J., Cyphert, J., and Reps, T., Closed forms for numerical loops. In *PACMPL 3(POPL)* (2019).

Kincaid, Z., Cyphert, J., Breck, J., and Reps, T., Non-linear reasoning for invariant synthesis. In *PACMPL 2(POPL)*: 54:1-54:33 (2018).

Harris, W., Jha, S., Reps, T., and Seshia, S., Program synthesis for interactive-security systems. In *Formal Methods in System Design (FMSD) 51*(2): 362-394 (2017).

Srinivasan, V., Vartanian, A., and Reps, T., Model-assisted machine-code synthesis. In *PACMPL 1(OOPSLA)*: 61:1-61:26 (2017).

Reps, T., Turetsky, E., and Prabhu, P., Newtonian program analysis via tensor product. In *ACM Trans. on Program. Lang. and Syst. 39*(2): 9:1-9:72 (2017).

Thakur, A., Lal, A., Lim, J., and Reps, T., PostHat and all that: Automating abstract interpretation. In *Electr. Notes Theor. Comput. Sci. 311*, 15–32 (2015).

Elder, M., Lim, J., Sharma, T., Andersen, T., and Reps, T., Abstract domains of affine relations. In *ACM Trans. on Program. Lang. and Syst. (TOPLAS) 36*(4): 11:1–11:73 (2014).

Aung, M., Horwitz, S., Joiner, R., and Reps, T., Specialization slicing. In *ACM Trans. on Program. Lang. and Syst. (TOPLAS) 36*(2): 5:1–5:67, (2014).

Also presented as a journal-first paper at the ACM Conference on Programming Language Design and Implementation (PLDI), 2014.

Lim, J. and Reps. T., TSL: A system for generating abstract interpreters and its application to machine-code analysis. In *ACM Trans. on Program. Lang. and Syst. (TOPLAS) 35*(1): 4:1–4:59 (2013).

Zhang, W., Sun, C., Lim, J., Lu, S., and Reps, T., ConMem: Detecting crash-triggering concurrency bugs through an effect-oriented approach. In *ACM Trans. on Softw. Eng. and Methodology (TOSEM) 22*(2): 10:1–10:33 (2013).

Kidd, N., Reps, T., Dolby, J., and Vaziri, M., Finding concurrency-related bugs using random isolation. In *Int. Journal on Software Tools for Technology Transfer (STTT) 13*, 6 (2011), 495–518.

Kidd, N., Lammich, P., Touili, T., and Reps, T., A decision procedure for detecting atomicity violations for communicating processes with locks. In *Int. Journal on Software Tools for Technology Transfer (STTT) 13*, 1 (2011), 37–60.

Lim, J., Lal, A., and Reps, T., Symbolic analysis via semantic reinterpretation. In *Int. Journal on Software Tools for Technology Transfer (STTT) 13*, 1 (2011), 61–87.

Elder, M., Gopan, D., and Reps, T., View-augmented abstractions. In *Electr. Notes Theor. Comput. Sci. 267*, 1 (2010), 43-57.

Reps, T., Sagiv, M., and Loginov, A., Finite differencing of logical formulas for static analysis. In *ACM Trans. on Program. Lang. and Syst. (TOPLAS) 32*(6): 24:1–24:55 (2010).

Balakrishnan, G. and Reps, T., WYSINWYX: What You See Is Not What You eXecute. In *ACM Trans. on Program. Lang. and Syst. (TOPLAS) 32*(6): 23:1–23:84 (2010).

Jeannet, B., Loginov, A., Reps, T., and Sagiv, M., A relational approach to interprocedural shape analysis. In *ACM Trans. on Program. Lang. and Syst. (TOPLAS) 32*(2): 5:1–5:52 (2010).

Lal, A. and Reps, T., Reducing concurrent analysis under a context bound to sequential analysis. In *Formal Methods in System Design 35*, 1 (2009), 73–97.

Lev-Ami, T., Immerman, N., Reps, T., Sagiv, M., Srivastava, S., and Yorsh, G., Simulating reachability using first-order logic with applications to verification of linked data structures. In *Logical Methods in Computer Science 5*, 2 (2009).

Yorsh, G., Reps, T., Sagiv, M., and Wilhelm, R., Logical characterizations of heap abstractions. In *ACM Transactions on Computational Logic 8*(1): 5:1–5:27 (2007).

Yahav, E., Reps, T., Sagiv, M., and Wilhelm, R., Verifying temporal heap properties specified via evolution logic. *Logic Journal of the IGPL 14*, 5 (Oct. 2006), 755-784.

Reps, T., Schwoon, S., Jha, S., and Melski, D., Weighted pushdown systems and their application to interprocedural dataflow analysis. *Science of Computer Programming 58*, 1-2 (Oct. 2005), 206-263.

Alur, R., Benedikt, M., Etessami, K., Godefroid, P., Reps, T., and Yannakakis, M., Analysis of recursive state machines. In *ACM Trans. on Prog. Lang. and Syst. (TOPLAS) 27*, 4 (2005), 786-818.

Yorsh, G., Skidanov, A., Reps, T., and Sagiv, M., Assume/guarantee reasoning for heap-manipulating programs. In *Electr. Notes Theor. Comput. Sci. 131*, 24 (May 2005), 125-138.

Jha, S. and Reps, T., Model checking SPKI/SDSI. In *Journal of Computer Security 12*, 3-4 (2004), 317-353.

Anderson, P., Reps, T., and Teitelbaum, T., Design and implementation of a fine-grained software inspection tool. In *IEEE Trans. on Softw. Eng. (TSE) 29*, 8 (Aug. 2003), 721-733.

Reps, T.W. and Rall, L.B., Computational divided differencing and divided-difference arithmetics. In *Higher-Order and Symbolic Computation 16*, 1-2 (2003), 93-149.

Clarke, E.M., Fujita, M., Rajan, P.S., Reps, T., Shankar, S., and Teitelbaum, T., Program slicing for VHDL. In *Software Tools for Technology Transfer (STTT) 4*(1), Oct. 2002, 125-137.

Sagiv, M., Reps, T., and Wilhelm, R., Parametric shape analysis via 3-valued logic. In *ACM Trans. on Program. Lang. and Syst. (TOPLAS) 24*(3), 2002, 217-298.

Reps, T., Undecidability of context-sensitive data-dependence analysis. In *ACM Trans. on Program. Lang. and Syst. (TOPLAS) 22*, 1 (Jan. 2000), pp. 162-186.

Melski, D. and Reps, T., Interconvertibility of a class of set constraints and context-free language reachability. In *Theoretical Computer Science 248*, 1-2 (Nov. 2000), pp. 29-98.

Siff, M. and Reps, T., Identifying modules via concept analysis. In *IEEE Trans. on Softw. Eng. (TSE) 25*, 6 (Nov./Dec. 1999), pp. 749-768.

Reps, T., Program analysis via graph reachability. *Information and Software Technology 40*, 11-12 (November/December 1998), pp. 701-726.

Reps, T., "Maximal-munch" tokenization in linear time. *ACM Trans. on Program. Lang. and Syst. (TOPLAS) 20*, 2 (March 1998), pp. 259-273.

Sagiv, M., Reps, T. and Wilhelm, R., Solving shape-analysis problems in languages with destructive updating. *ACM Trans. on Program. Lang. and Syst. (TOPLAS) 20*, 1 (January 1998), pp. 1-50.

Sagiv, M., Reps, T., and Horwitz, S., Precise interprocedural dataflow analysis with applications to constant propagation. *Theoretical Computer Science 167* (1996), pp. 131-170.

Reps, T., On the sequential nature of interprocedural program-analysis problems. *Acta Informatica 33* (1996), pp. 739-757.

Ramalingam, G. and Reps, T., An incremental algorithm for a generalization of the shortest-path problem. *Journal of Algorithms 21* (1996), pp. 267-305.

Ramalingam, G. and Reps, T., On the computational complexity of dynamic graph problems. *Theoretical Computer Science A 158* (May 1996), pp. 233-277.

Binkley, D., Horwitz, S., and Reps, T., Program integration for languages with procedure calls. *ACM Trans. on Softw. Eng. and Methodology (TOSEM) 4*, 1 (January 1995), pp. 3-35.

Ramalingam, G. and Reps, T., On competitive on-line algorithms for the dynamic priority-ordering problem. *Information Processing Letters 51* (1994), 155-161.

Yang, W., Horwitz, S., and Reps, T., A program integration algorithm that accommodates semantics-preserving transformations. *ACM Trans. on Softw. Eng. and Methodology (TOSEM) 1*, 3 (July 1992), 310-354.

Reps, T., Algebraic properties of program integration. *Science of Computer Programming 17* (1991), 139-215.

Horwitz, S. and Reps, T., Efficient comparison of program slices. *Acta Informatica 28* (1991), 713-732.

Horwitz, S., Reps, T., and Binkley, D., Interprocedural slicing using dependence graphs. *ACM Trans. on Program. Lang. and Syst. (TOPLAS) 12*, 1 (January 1990), 26-60.

Horwitz, S., Prins, J., and Reps, T., Integrating non-interfering versions of programs. *ACM Trans. on Program. Lang. and Syst. (TOPLAS) 11*, 3 (July 1989), 345-387.

Reps, T. Incremental evaluation for attribute grammars with unrestricted movement between tree modifications. *Acta Informatica 25* (1988), 155-178.

Reps, T. and Demers, A., Sublinear-space evaluation algorithms for attribute grammars. *ACM Trans. on Program. Lang. and Syst. (TOPLAS) 9*, 3 (July 1987), 408-440.

Reps, T., Teitelbaum, T., and Demers, A., Incremental context-dependent analysis for language-based editors. *ACM Trans. on Program. Lang. and Syst. (TOPLAS) 5*, 3 (July 1983), 449-477.

Teitelbaum, T. and Reps, T., The Cornell Program Synthesizer: A syntax-directed programming environment. *Communications of the ACM 24*, 9 (September 1981), 563-573.

Invited Papers

Kincaid, Z., Reps, T.W., and Cyphert, J., Algebraic program analysis (Invited tutorial). In *Computer Aided Verification (CAV), Part I*, LNCS 12759, Springer, 2021.

D'Antoni, L., Hu, Q., Kim, J., and Reps, T.W., Programmable program synthesis. In *Computer Aided Verification (CAV), Part I*, LNCS 12759, Springer, 2021.

Reps, T., Program analyses using Newton's method. In *Proc. Int. Conf. on NETworked sYStems (NETYS)*, May 2018.

Reps, T. and Thakur, A., Automating abstract interpretation. In *Proc. Verification, Model Checking, and Abstract Interpretation (VMCAI)*, Jan. 2016.

McCloskey, B., Reps, T., and Sagiv, M., Statically inferring complex heap, array, and numeric invariants. In *Proc. Static Analysis Symposium (SAS)*, Sept. 2010.

Reps, T., Lim, J., Thakur, A., Balakrishnan, G., and Lal, A., There's plenty of room at the bottom: Analyzing and verifying machine code (Invited tutorial). In *Proc. Computer Aided Verification (CAV)*, July 2010.

Reps, T. and Balakrishnan, G., Improved memory-access analysis for x86 executables. (Paper accompanying a ''unifying'' invited talk at ETAPS 2008.) In *Proc. Int. Conf. on Compiler Construction (CC)*), April 2008.

Reps, T., Lal, A., and Kidd, N., Program analysis using weighted pushdown systems. In *Proc. 27th Conf. on Foundations of Software Technology and Theoretical Computer Science (FSTTCS)*, (New Delhi, India, Dec. 12–14, 2007).

Balakrishnan, G. and Reps, T., DIVINE: DIscovering Variables IN Executables. In *Proc. Int. Conf. on Verification, Model Checking and Abstract Interpretation (VMCAI)*, (Nice, France, Jan. 14–16, 2007).

Jha, S., Schwoon, S., Wang, H., and Reps, T., Weighted pushdown systems and trust-management systems. In *Proc. Int. Conf. on Tools and Algorithms for the Construction and Analysis of Systems (TACAS)*, (Vienna, Austria, Mar. 25 – Apr. 2, 2006).

Reps, T., Balakrishnan, G., and Lim, J., Intermediate-representation recovery from low-level code. In *Proc. Workshop on Partial Evaluation and Program Manipulation (PEPM)*, (Charleston, SC, Jan. 9-10, 2006).

Reps, T., Balakrishnan, G., Lim, J., and Teitelbaum, T., A next-generation platform for analyzing executables. In *Proc. 3rd Asian Symposium on Programming Languages and Systems (APLAS)*, (Tsukuba, Japan, Nov. 3-5, 2005), 212-229.

Reps, T., Sagiv, M., and Wilhelm, R., Static program analysis via 3-valued logic. In *Proc. Int. Conf. on Computer-Aided Verification (CAV)*, 2004, 15-30.

Rall, L.B. and Reps, T.W., Algorithmic differencing. In *Perspectives on Enclosure Methods*, U. Kulisch, R. Lohner, and A. Facius (eds.), Springer-Verlag, Vienna, 2001, 133-147. (Invited paper presented at SCAN 2000: 9th GAMM-IMACS Int. Symp. on Sci. Comput., Comp. Arith., and Validated Numerics, (Karlsruhe, Ger., Sept. 19-22, 2000).)

Wilhelm, R., Sagiv, M., and Reps, T., Shape analysis. In *Proc. 9th Int. Conf. on Compiler Construction* (CC), (Berlin, Ger., Mar. 27 − Apr. 2, 2000).

Reps, T., Program analysis via graph reachability. In *Proc. Int. Logic Programming Symposium (ILPS)*, (Port Jefferson, NY, Oct. 12-16, 1997), J. Maluszynski (ed.), The M.I.T. Press, Cambridge, MA, 1997, pp. 5-19.

Reps, T., The use of program profiling in software testing. In *Proc. of Informatik '97* (Aachen, Germany, Sept. 24-27, 1997), M. Jarke, K. Pasedach, and K. Pohl (eds.), Springer-Verlag, Berlin, Ger., 1997, pp. 4-16.

Horwitz, S. and Reps, T., The use of program dependence graphs in software engineering. In *Proc. 14th Int. Conference on Software Engineering (ICSE)*, (Melbourne, Australia, May 11-15, 1992), ACM, New York, NY, 1992, pp. 392-411.

Reps, T. and Horwitz, S., Semantics-based program integration. In *Proc. 2nd European Symposium on Programming (ESOP)*, (Nancy, France, March 21-25, 1988), *Lecture Notes in Computer Science*, Vol. 300, H. Ganzinger (ed.), Springer-Verlag, New York, NY, 1988, pp. 1-20.

Refereed Conference and Workshop Publications

Kalita, P.K., Reps, T., and Roy, S., Synthesizing abstract transformers for reduced-product domains. In *Proc. Static Analysis Symposium* (SAS), 2024.

Jain, A., Chaudhuri, S., Reps, T.W., and Jermaine, C.M., Prompt tuning strikes back: Customizing foundation models with low-rank prompt adaptation. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2024.

Johnson, K.J.C., Reynolds, A., Reps, T., and D'Antoni, L., The SemGuS Toolkit. In *Computer Aided Verification (CAV)*, 2024.

Park, K., Johnson, K.J.C., D'Antoni, L. and Reps, T., Modular system synthesis. In *Formal Methods in Computer-Aided Design (FMCAD)*, 2023.

Sistla, M., Chaudhuri, S., and Reps, T., Symbolic quantum simulation with Quasimodo. In *Computer Aided Verification (CAV)*, 2023.

Alhanahnah, M., Jain, R., Rastogi, V. Jha, S., and Reps, T., Lightweight, multi-stage, compiler-assisted application specialization. In *7th IEEE European Symposium on Security and Privacy*, June 2022.

Henkel, J.,* Ramakrishnan, G.,* Wang, Z., Albarghouthi, A., Jha, S., and Reps, T., Semantic robustness of models of source code. In *29th IEEE Int. Conf. on Software Analysis, Evolution and Reengineering (SANER)*, March 2022. ("*" denotes equal contributions.)

Mukherjee, R., Wen, Y., Chaudhari, D., Reps, T., Chaudhuri, S., and Jermaine, C., Neural program generation modulo static analysis. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2021 (**spotlight presentation**).

Hu, Q., Cyphert, J., D'Antoni, L., and Reps, T.W., Synthesis with asymptotic resource bounds. In *Computer Aided Verification (CAV), Part I*, LNCS 12759, Springer, 2021.

Henkel, J., Silva, D., Teixeira, L., d'Amorim, M., and Reps, T.W., Shipwright: A human-in-the-loop system for Dockerfile repair. In *Proc. Int. Conf. on Software Engineering (ICSE)*, 2021.

Wang, D., Hoffmann, J., and Reps, T., Central moment analysis for cost accumulators in probabilistic programs. In *ACM Conference on Programming Language Design and Implementation (PLDI)*, 2021.

Wang, D., Hoffmann, J., and Reps, T., Sound probabilistic inference via guide types. In *ACM Conference on Programming Language Design and Implementation (PLDI)*, 2021.

Henkel, J.,* Silva, D.,* Teixeira, L, d'Amorim, M., and Reps, T., Shipwright: An approach to building, clustering, analyzing, and fixing broken Dockerfiles. In *Int. Conf. on Software Engineering (ICSE 2021)*. ("*" denotes equal contributions.)

Henkel, J., Bird, C., Lahiri, S.K., and Reps, T., A dataset of Dockerfiles. In *Mining Software Repositories (Data Showcase Track)*, 2020.

Henkel, J., Bird, C., Lahiri, S.K., and Reps, T., Learning from, understanding, and supporting DevOps artifacts for Docker. In *Proc. Int. Conf. on Software Engineering (ICSE)*, 2020.

Hu, Q., Cyphert, J., D'Antoni, L., and Reps, T., Exact and approximate methods for proving unrealizability of syntax-guided synthesis problems. In *Proc. ACM Conf. on Programming Language Design and Implementation*, (PLDI), 2020.

Li, Y., Zhang, Q., and Reps, T., Fast graph simplification for interleaved Dyck-reachability. In *Proc. ACM Conf. on Programming Language Design and Implementation*, (PLDI), 2020.

Received a 2020 PLDI Distinguished Paper Award

Invited for special submission to *ACM Trans. on Program. Lang. and Syst. (TOPLAS)*.

Breck, J., Cyphert, J., Kincaid, Z., and Reps, T., Templates and recurrences: Better together. In *Proc. ACM Conf. on Programming Language Design and Implementation*, (PLDI), 2020.

Hu, Q., Breck, J., Cyphert, J., D'Antoni, L., and Reps, T., Proving unrealizability for syntax-guided synthesis. In *Computer Aided Verification (CAV)*, 2019.

Wang, D., Hoffmann, J., and Reps, T., A denotational semantics for low-level probabilistic programs with nondeterminism. In *Mathematical Foundations of Programming Semantics (MFPS)*, 2019.

Henkel, J., Lahiri, S., Liblit, B., and Reps, T., Code vectors: Understanding programs through embedded abstracted symbolic traces. In *ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering (ESEC/FSE)*, 2018.

Wang, D., Hoffmann, J., and Reps, T., PMAF: An algebraic framework for static analysis of probabilistic programs. In *Proc. ACM Conference on Programming Language Design and Implementation* (PLDI), 2018.

Sharma, T. and Reps, T., A new abstraction framework for affine transformers. In *Proc. Static Analysis Symposium* (SAS), 2017.

Invited for special submission to the *Int. Journal on Formal Methods in System Design (FMSD).*

Brown, D.B., Vaughn, M., Liblit, B., and Reps, T., The care and feeding of wild-caught mutants. In *Proc. Joint Meeting of the European Software Eng. Conf. and the ACM SIGSOFT Symp. on the Foundations of Software Engineering (ESEC/FSE)*, 2017.

Carbonneaux, Q., Hoffmann, J., Reps, T., and Shao, Z., Automated resource analysis with Coq proof objects. In *Proc. Int. Conf. on Computer Aided Verifications (CAV)*, 2017.

Kincaid, Z., Breck, J., Forouhi Boroujeni, A., and Reps, T., Compositional recurrence analysis revisited. In *Proc. ACM Conference on Programming Language Design and Implementation* (PLDI), 2017.

Sharma, T. and Reps, T., Sound bit-precise numerical domains. In *Proc. Verification, Model Checking, and Abstract Interpretation* (VMCAI), 2017.

Feng, Y., Martins, R., Wang, Y., Dillig, I., and Reps, T.W., Component-based synthesis for complex APIs. In *Proc. ACM Symposium on Principles of Programming Languages (POPL)*, 2017.

Srinivasan, V., Sharma, T., and Reps, T., Speeding up machine-code synthesis. In *Proc. ACM SIGPLAN Conf. on Systems, Programming, Languages and Applications: Software for Humanity* (SPLASH/OOPSLA), 2016.

Srinivasan, V. and Reps, T., An improved algorithm for slicing machine code. In *Proc. ACM SIGPLAN Conf. on Systems, Programming, Languages and Applications: Software for Humanity* (SPLASH/OOPSLA), 2016.

Mine, A., Breck, J., and Reps, T., An algorithm inspired by constraint solvers to infer inductive invariants in numeric programs. In *Proc. European Symp. on Programming (ESOP)*, Apr. 2016.

Nominated for the EASST Best-Paper Award at ETAPS 2016.

Selected for *Notable Books and Articles***, Best of Computing–2016,** *ACM Computing Reviews***, July 2017. (See www.computingreviews.com/recommend/bestof/notableitems.cfm?bestYear=2016.)**

Reps, T., Turetsky, E., and Prabhu, P., Newtonian program analysis via tensor product. In *Conference Record of the ACM Symposium on Principles of Programming Languages* (POPL), 2016.

Ohmann, P., Brown, D.B., Liblit, B., and Reps, T., Recovering execution data from incomplete observations. In *Proc. 13th Int. Workshop on Dynamic Analysis* (WODA), 2015.

Srinivasan, V. and Reps, T., Partial evaluation of machine code. In *Proc. ACM SIGPLAN Conference on Systems, Programming, Languages and Applications: Software for Humanity* (SPLASH/OOPSLA), 2015.

Srinivasan, V. and Reps, T., Synthesis of machine code from semantics. In *Proc. ACM Conference on Programming Language Design and Implementation* (PLDI), 2015.

Joiner, R., Reps, T., Jha, S., Dhawan, M., and Ganapathy, V., Efficient runtime enforcement techniques for policy weaving. In *Proc. Found. of Software Engineering* (FSE), 2014.

Thakur, A., Breck, J., and Reps, T., Satisfiability modulo abstraction for separation logic with linked lists. In *Proc. Int. SPIN Symposium on Model Checking of Software* (SPIN), 2014.

Itzhaky, S., N. Bjørner, Reps, T., Sagiv, M., and Thakur, A., Property-directed shape analysis. In *Proc. Computer-Aided Verification* (CAV), 2014.

Srinivasan, V. and Reps, T., Recovery of class hierarchies and composition relationships from machine code. In *Proc. Int. Conf. on Compiler Construction* (CC), 2014.

Kreiker, J., Reps, T., Rinetzky, N., Sagiv, M., Wilhelm, R., and Yahav, E., Interprocedural shape analysis for effectively cutpoint-free programs. In *Programming Logics: Essays in Memory of Harald Ganzinger*, LNCS 7797, Springer, 2013.

Harris, W.R., Jha, S., Reps, T., Anderson, J., and Watson, R.N.M., Declarative, temporal, and practical programming with capabilities. In *Proc. IEEE Symposium on Security and Privacy* (SP), 2013.

Thakur, A., Elder, M., and Reps, T., Bilateral algorithms for symbolic abstraction. In *Proc. Static Analysis Symposium* (SAS), 2012.

Thakur, A. and Reps, T., A generalization of Staalmarck's method. In *Proc. Static Analysis Symposium* (SAS), 2012.

Harris, W.R., Jha, S., and Reps, T., Secure programming via visibly pushdown safety games. In *Proc. Computer-Aided Verification* (CAV), 2012.

Fredrikson, M., Joiner, R., Jha, S., Reps, T., Porras, P., Saidi, H., and Yegneswaran, V., Efficient runtime policy enforcement using counterexample-guided abstraction refinement. In *Proc. Computer-Aided Verification* (CAV), 2012.

Thakur, A. and Reps, T., A method for symbolic computation of abstract operations. In *Proc. Computer-Aided Verification* (CAV), 2012.

Driscoll, E., Thakur, A., and Reps, T., OpenNWA: A nested-word-automaton library (tool paper). In *Proc. Computer-Aided Verification* (CAV), 2012.

Elder, M., Lim, J., Sharma, T., Andersen, T., and Reps, T., Abstract domains of affine relations. In *Proc. Static Analysis Symposium* (SAS), 2011.

Invited for special submission to *ACM Trans. on Program. Lang. and Syst. (TOPLAS)*.

Driscoll, E., Burton, A., and Reps, T., Checking conformance of a producer and a consumer. In *Proc. Found. of Software Engineering* (FSE), 2011.

Zhang, W., Lim, J., Olichandran, R., Scherpelz, J., Jin, G., Lu, S., and Reps, T., ConSeq: Detecting concurrency bugs through sequential errors. In *Proc. Architectural Support for Programming Languages and Operating Systems (ASPLOS)*, 2011.

Harris, W., Jha, S., and Reps, T., DIFC programs by automatic instrumentation. In *Proc. ACM Conf. on Computer and Communications Security (CCS)*, 2010.

Thakur, A., Lim, J., Lal, A., Burton, A., Driscoll, E., Elder, M., Andersen, T., and Reps, T., Directed proof generation for machine code. In *Proc. Computer-Aided Verification (CAV)*, 2010.

Harris, W., Kidd, N., Chaki, S., Jha, S., and Reps, T., Verifying information flow over unbounded processes. In *Proc. Int. Symp. on Formal Methods (FM)*, 2009.

Kidd, N., Lammich, P., Touili, T., and Reps, T., A decision procedure for detecting atomicity violations for communicating processes with locks. In *Proc. SPIN Workshop*, 2009, 125–142.

Invited for special submission to the *Int. Journal on Software Tools for Tech. Transfer (STTT)***.**

Lim, J., Lal, A., and Reps, T., Symbolic analysis via semantic reinterpretation. In *Proc. SPIN Workshop*, 2009, 148–168.

Invited for special submission to the *Int. Journal on Software Tools for Tech. Transfer (STTT)***.**

Kidd, N., Reps, T., Dolby, J., and Vaziri, M., Finding concurrency-related bugs using random isolation. In *Proc. Verification, Model Checking, and Abstract Interpretation* (VMCAI), 2009.

Invited for special submission to the *Int. Journal on Software Tools for Tech. Transfer (STTT)***.**

Lal, A. and Reps, T., Reducing concurrent analysis under a context bound to sequential analysis. In *Proc. Computer-Aided Verification (CAV)*, 2008.

Invited for special submission to the *Int. Journal on Formal Methods in System Design (FMSD)***.**

Lal, A. and Reps, T., Solving multiple dataflow queries using WPDSs. In *Proc. Static Analysis Symposium* (SAS), July 2008.

Kidd, N., Lal A., and Reps, T., Language strength reduction. In *Proc. Static Analysis Symposium* (SAS), July 2008.

Lim, J. and Reps, T., A system for generating static analyzers for machine instructions. In *Proc. Int. Conf. on Compiler Construction* (CC), April 2008.

Received the ETAPS Best-Paper Award for 2008 from the European Association for Programming Languages and Systems (EAPLS).

Balakrishnan, G. and Reps, T., Analyzing stripped device-driver executables. In *Proc. Int. Conf. on Tools and Algorithms for the Construction and Analysis of Systems (TACAS)*, Springer-Verlag, New York, NY, April 2008.

Lal, A., Touili, T., Kidd, N., and Reps, T., Interprocedural analysis of concurrent programs under a context bound. In *Proc. Int. Conf. on Tools and Algorithms for the Construction and Analysis of Systems (TACAS)*, Springer-Verlag, New York, NY, April 2008.

Lal, A., Kidd, N., Reps, T., and Touili, T., Abstract error projection. In *Proc. Static Analysis Symposium* (SAS), August 2007.

Gopan, D. and Reps, T., Guided static analysis. In *Proc. Static Analysis Symposium* (SAS), August 2007.

Lev-Ami, T., Weidenbach, C., Reps, T., and Sagiv, M., Labelled clauses. In *Proc. Conference on Automated Deduction* (CADE), 2007.

Gopan, D. and Reps, T., Low-level library analysis and summarization. In *Proc. Computer-Aided Verification (CAV)*, 2007.

Amit, D., Rinetzky, N., Reps, T., Sagiv, M., and Yahav, E., Comparison under abstraction for verifying linearizability. In *Proc. Computer-Aided Verification (CAV)*, 2007.

Bogudlov, I., Lev-Ami, T., Reps, T, and Sagiv, M., Revamping TVLA: Making parametric shape analysis competitive (tool paper). In *Proc. Computer-Aided Verification (CAV)*, 2007.

Loginov, A., Reps, T., and Sagiv, M., Refinement-based verification for possibly-cyclic lists. In *Program Analysis and Compilation, Theory and Practice: Essays Dedicated to Reinhard Wilhelm*, Springer-Verlag, 2007.

Balakrishnan, G., Reps, T., Melski, D., and Teitelbaum, T., WYSINWYX: What You See Is Not What You eXecute. In *Verified Software: Theories, Tools, Experiments* (VSTTE), Springer-Verlag, 2007.

Dor, N., Field, J., Gopan, D., Lev-Ami, T., Loginov, A., Manevich, R., Ramalingam, G., Reps, T., Rinetzky, N., Sagiv, M., Wilhelm, R., Yahav, E., and Yorsh, G., Automatic verification of strongly dynamic software systems. In *Verified Software: Theories, Tools, Experiments* (VSTTE), Springer-Verlag, 2007.

Lev-Ami, T., Sagiv, M., Immerman, N., and Reps, T., Shape analysis of uniform change. In *Proc. Int. Conf. on Verification, Model Checking and Abstract Interpretation*, (VMCAI) (Nice, France, Jan. 14–16, 2007).

Lim, J., Reps, T., and Liblit, B., Extracting output formats from executables. In *Proc. IEEE Working Conference on Reverse Engineering* (WCRE), (Benevento, Italy, Oct. 23-27, 2006).

Wang, H., Jha, S., Reps, T., Schwoon, S., and Stubblebine, S., Reducing the dependence of SPKI/SDSI on PKI. In *Proc. European Symp. on Research in Computer Security* (ESORICS), 2006.

Balakrishnan, G. and Reps, T., Recency-abstraction for heap-allocated storage. In *Proc. Static Analysis Symposium* (SAS), Springer-Verlag, New York, NY, 2006.

Loginov, A., Reps, T., and Sagiv, M., Automated verification of the Deutsch-Schorr-Waite tree-traversal algorithm. In *Proc. Static Analysis Symposium* (SAS), Springer-Verlag, New York, NY, 2006.

Lal, A. and Reps, T., Improving pushdown system model checking. In *Proc. Computer-Aided Verification* (CAV), Springer-Verlag, New York, NY, 2006.

Gopan, D. and Reps, T., Lookahead widening. In *Proc. Computer-Aided Verification* (CAV), Springer-Verlag, New York, NY, 2006.

Chaki, S., Clarke, E., Kidd, N., Reps, T., and Touili, T., Verifying concurrent message-passing C programs with recursive calls. In *Proc. Int. Conf. on Tools and Algorithms for the Construction and Analysis of Systems (TACAS)*, Springer-Verlag, New York, NY, 2006.

Jeannet, B., Gopan, D., and Reps, T., A relational abstraction for functions. In *Proc. 12th Int. Static Analysis Symp.*, 2005.

Lev-Ami, T., Immerman, N., Reps, T., Sagiv, M., Srivastava, S., and Yorsh, G., Simulating reachability using first-order logic with applications to verification of linked data structures. In *Proc. Conf. on Automated Deduction*, 2005.

Balakrishnan, G., Reps, T., Kidd, N., Lal, A., Lim, J., Melski, D., Gruian, R., Yong, S., Chen, C.-H., and Teitelbaum, T., Model checking x86 executables with CodeSurfer/x86 and WPDS++, (tool-demonstration paper). In *Proc. Computer-Aided Verification* (CAV), 2005.

Lal, A., Balakrishnan, G., and Reps, T., Extended weighted pushdown systems. In *Proc. Computer-Aided Verification* (CAV), 2005.

Loginov, A., Reps, T, and Sagiv, M., Abstraction refinement via inductive learning. In *Proc. Computer-Aided Verification* (CAV), 2005.

Ganapathy, V., Seshia, S.A., Jha, S., Reps, T.W., and Bryant, R.E., Automatic discovery of API-level exploits. In *Proc. Int. Conf. on Software Engineering (ICSE)*, (St. Louis, Missouri, May 15-21, 2005).

Balakrishnan, G., Gruian, R., Reps, T., and Teitelbaum, T., CodeSurfer/x86–A platform for analyzing x86 executables, (tool demonstration paper). In *Int. Conf. on Compiler Construction* (CC), April 2005.

Jeannet, B., Gopan, D., and Reps, T., A relational abstraction for functions. In *Int. Workshop on Numerical and Symbolic Abstract Domains*, Jan. 2005.

Gopan, D., Reps, T., and Sagiv, M., Numeric analysis of array operations. In *Conference Record of the Thirty-Second ACM Symposium on Principles of Programming Languages* (POPL), (Long Beach, CA, Jan. 12-14, 2005), 338-350.

Rinetzky, N., Bauer, J., Reps, T., Sagiv, M., and Wilhelm, R., A semantics for procedure local heaps and its abstractions. In *Conference Record of the Thirty-Second ACM Symposium on Principles of Programming Languages* (POPL), (Long Beach, CA, Jan. 12-14, 2005), 296-309.

Immerman, N., Rabinovich, A., Reps, T., Sagiv, M., and Yorsh, G., The boundary between decidability and undecidability for transitive closure logics. In *Proc. Computer Science Logic*, 2004. Springer-Verlag, New York, NY, 2004, 160-174.

Jeannet, B., Loginov, A., Reps, T., and Sagiv, M., A relational approach to interprocedural shape analysis. In *Proc. 11th Int. Static Analysis Symp.*, Springer-Verlag, New York, NY, 2004.

Immerman, N., Rabinovich, A., Reps, T., Sagiv, M., and Yorsh, G., Verification via structure simulation. In *Proc. Int. Conf. on Computer-Aided Verification* (CAV), 2004, 281-294.

Balakrishnan, G. and Reps, T., Analyzing memory accesses in x86 executables. In *Proc. Int. Conf. on Compiler Construction* (CC), Springer-Verlag, New York, NY, 2004, 5-23.

Received the ETAPS Best-Paper Award for 2004 from the European Association for Programming Languages and Systems (EAPLS).

Yorsh, G., Reps, T., and Sagiv, M., Symbolically computing most-precise abstract operations for shape analysis. In *Proc. Int. Conf. on Tools and Algorithms for the Construction and Analysis of Systems (TACAS)*, Springer-Verlag, New York, NY, 2004, 530-545.

Gopan, D., DiMaio, F., Dor, N., Reps, T., and Sagiv, M., Numeric domains with summarized dimensions. In *Proc. Int. Conf. on Tools and Algorithms for the Construction and Analysis of Systems (TACAS)*, Springer-Verlag, New York, NY, 2004, 512-529.

Reps, T., Sagiv, M., and Yorsh, G., Symbolic implementation of the best transformer. In *Proc. Verification, Model Checking, and Abstract Interpretation (VMCAI)*, 2004, 252-266.

Reps, T., Schwoon, S., and Jha, S., Weighted pushdown systems and their application to interprocedural dataflow analysis. In *Proc. 10th Int. Static Analysis Symp. (SAS)*, (June 11–13, 2003, San Diego, CA), *Lecture Notes in Computer Science*, Vol. 2694, Springer-Verlag, New York, NY, 2003, pp. 189-213.

Invited for special submission to the *Science of Computer Programming (SCP)***.**

Schwoon, S., Jha, S., Reps, T., and Stubblebine, S., On generalized authorization problems. In *Proc. 16th IEEE Computer Security Foundations Workshop (CSFW)*, (June 30–July 2, 2003, Asilomar, Pacific Grove, CA), pp. 202-218.

Reps, T., Sagiv, M., and Loginov, A., Finite differencing of logical formulas for static analysis. In *Proc. European Symp. on Programming (ESOP)*, *Lecture Notes in Computer Science*, Vol. 2618, Springer-Verlag, New York, NY, 2003, pp. 380-398.

Yahav, E., Reps, T., Sagiv, M., and Wilhelm, R. Verifying temporal heap properties specified via evolution logic. In *Proc. European Symp. on Programming (ESOP)*, *Lecture Notes in Computer Science*, Vol. 2618, Springer-Verlag, New York, NY, 2003, pp. 204-222.

Melski, D and Reps, T., The interprocedural express-lane transformation. In *Proc. Int. Conf. on Compiler Construction* (CC), *Lecture Notes in Computer Science*, Vol. 2622, Springer-Verlag, New York, NY, 2003, pp. 200-216.

Reps, T., Loginov, A., and Sagiv, M., Semantic minimization of 3-valued propositional formulae. In *Proc. IEEE Symp. on Logic in Computer Science (LICS)*, (Copenhagen, Denmark, July 22–25, 2002), pp. 40-54.

Jha, S. and Reps, T., Analysis of SPKI/SDSI certificates using model checking. In *Proc. 15th IEEE Computer Security Foundations Workshop (CSFW)*, (Cape Breton, Nova Scotia, June 24–26, 2002), pp. 129-146.

Invited for special submission to the *Journal of Computer Security (JCS)***.**

Benedikt, M., Godefroid, P., and Reps, T., Model checking of unrestricted hierarchical state machines. In *Proc. 28th Int. Colloq. on Automata, Languages, and Programming (ICALP)*, (Crete, Greece, July 8–12, 2001), *Lecture Notes in Computer Science*, Vol. 2076, Springer-Verlag, New York, NY, 2001, pp. 652-666.

Loginov, A., Yong, S.H., Horwitz, S., and Reps, T., Debugging via run-time type checking. In *Proc. of FASE 2001: Fundamental Approaches to Softw. Eng.*, (Genoa, Italy, April 2–6, 2001).

Xu, Z., Reps, T., and Miller, B., Typestate checking of machine code. In *Proc. of ESOP 2001: European Symp. on Programming*, (Genoa, Italy, April 2–6, 2001).

Lev-Ami, T., Reps, T., Sagiv, M., and Wilhelm, R., Putting static analysis to work for verification: A case study. In *ISSTA 2000: Proc. of the Int. Symp. on Software Testing and Analysis*, (Portland, OR, Aug. 21-25, 2000).

Xu, Z., Miller, B., and Reps, T., Safety checking of machine code. In *Proc. ACM Conf. on Programming Language Design and Implementation*, (PLDI) (Vancouver B.C., Canada, June 18-21, 2000), pp. 70-82.

Chandra, S. and Reps, T., Physical type checking for C. In *Proc. Workshop on Program Analysis for Software Tools and Engineering (PASTE)*, (Toulouse, France, Sept. 6, 1999). *ACM SIGSOFT Software Engineering Notes 24*, 5 (Sept. 1999), pp. 66-75.

Siff, M., Chandra, S., Ball, T., Kunchithapadam, K., and Reps, T., Coping with type casts in C. In *Proceedings of ESEC/FSE '99: Seventh European Software Engineering Conference and Seventh ACM SIGSOFT Symposium on the Foundations of Software Engineering*, (Toulouse, France, Sept. 6-10, 1999), pp. 180-198.

Clarke, E.M., Fujita, M., Rajan, P.S., Reps, T., Shankar, S., and Teitelbaum, T., Program slicing of hardware description languages. In *Proc. of Charme '99*, (Bad Herrenalb, Ger., Sept. 27-29, 1999).

Invited for special submission to the *Int. Journal on Software Tools for Tech. Transfer (STTT)***.**

Yong, S.H., Horwitz, S., and Reps, T., Pointer analysis for programs with structures and casting. In *Proc. ACM Conf. on Programming Language Design and Implementation*, (PLDI) (Atlanta, GA, May 1-4, 1999), *ACM SIGPLAN Notices 34*, 5 (May 1999), pp. 91-103.

Melski, D. and Reps, T., Interprocedural path profiling. In *Proc. 8th Int. Conf. on Compiler Construction* (CC), (Amsterdam, The Netherlands, Mar. 22-26, 1999), *Lecture Notes in Computer Science*, Vol. 1575, S. Jaehnichen (ed.), Springer-Verlag, New York, NY, 1999, pp. 47-62.

Benedikt, M., Reps, T., and Sagiv, M., A decidable logic for describing linked data structures. In *Proc. of ESOP '99: European Symposium on Programming*, (Amsterdam, The Netherlands, Mar. 22-26, 1999), *Lecture Notes in Computer Science*, Vol. 1576, S.D. Swierstra (ed.), Springer-Verlag, New York, NY, 1999, pp. 2-19.

Sagiv, M., Reps, T., and Wilhelm, R., Parametric shape analysis via 3-valued logic. In *Conference Record of the Twenty-Sixth ACM Symposium on Principles of Programming Languages (POPL)*, (San Antonio, TX, Jan. 20-22, 1999), ACM, New York, NY, 1999, pp. 105-118.

Siff, M. and Reps, T., Identifying modules via concept analysis. In *ICSM '97: IEEE International Conference on Software Maintenance* (Bari, Italy, Oct. 1-3, 1997), M.J. Harrold and G. Visaggio (eds.), IEEE Computer Society, Washington, DC, 1997, pp. 170-179.

Invited for special submission to *IEEE Trans. on Software Engineering (TSE)***.**

Reps, T., Ball, T., Das, M., and Larus, J., The use of program profiling for software maintenance with applications to the Year 2000 Problem. In *Proceedings of ESEC/FSE '97: Sixth European Software Engineering Conference and Fifth ACM SIGSOFT Symposium on the Foundations of Software Engineering*, (Zurich, Switzerland, Sept. 22-25, 1997), *Lecture Notes in Computer Science*, Vol. 1301, M. Jazayeri and H. Schauer (eds.), Springer-Verlag, New York, NY, 1997, pp. 432-449.

Melski, D. and Reps, T., Interconvertibility of set constraints and context-free language reachability. In *PEPM '97: Proceedings of the ACM SIGPLAN Symposium on Partial Evaluation and Semantics-Based Program Manipulation*, (Amsterdam, The Netherlands, June 12-13, 1997), ACM, New York, NY, 1997, pp. 74-89.

Invited for special submission to *Theoretical Computer Science (TCS)***.**

Siff, M. and Reps, T., Program generalization for software reuse: From C to C++. In *SIGSOFT 96: Proceedings of the Fourth ACM SIGSOFT Symposium on the Foundations of Software Engineering*, (San Francisco, CA, October 16-18, 1996), ACM, New York, NY, 1996, pp. 135-146.

Reps, T. and Turnidge, T., Program specialization via program slicing. In *Proceedings of the Dagstuhl Seminar on Partial Evaluation*, (Schloss Dagstuhl, Wadern, Germany, Feb. 12-16, 1996), *Lecture Notes in Computer Science*, Vol. 1110, O. Danvy, R. Glueck, and P. Thiemann (eds.), Springer-Verlag, New York, NY, 1996, pp. 409-429.

Sagiv, M., Reps, T., and Wilhelm, R., Solving shape-analysis problems in languages with destructive updating. In *Conference Record of the Twenty-Third ACM Symposium on Principles of Programming Languages*, (St. Petersburg, FL, Jan. 22-24, 1996), ACM, New York, NY, 1996, pp. 16-31.

Horwitz, S., Reps, T., and Sagiv, M., Demand interprocedural dataflow analysis. In *SIGSOFT '95: Proceedings of the Third ACM SIGSOFT Symposium on the Foundations of Software Engineering*, (Washington, DC, October 10-13, 1995), *ACM SIGSOFT Software Engineering Notes 20*, 4 (1995), pp. 104-115.

Reps, T. and Rosay, G., Precise interprocedural chopping. In *SIGSOFT '95: Proceedings of the Third ACM SIGSOFT Symposium on the Foundations of Software Engineering*, (Washington, DC, October 10-13, 1995), *ACM SIGSOFT Software Engineering Notes 20*, 4 (1995), pp. 41-52.

Reps, T., Shape analysis as a generalized path problem. In *PEPM '95: Proceedings of the ACM SIGPLAN Symposium on Partial Evaluation and Semantics-Based Program Manipulation*, (La Jolla, California, June 21-23, 1995), ACM, New York, NY, 1995, pp. 1-11.

Das, M., Reps, T., and Van Hentenryck, P., Semantic foundations of binding-time analysis for imperative programs. In *PEPM '95: Proceedings of the ACM SIGPLAN Symposium on Partial Evaluation and Semantics-Based Program Manipulation*, (La Jolla, California, June 21-23, 1995), ACM, New York, NY, 1995, pp. 100-110.

Sagiv, M., Reps, T., and Horwitz, S., Precise interprocedural dataflow analysis with applications to constant propagation. In *Proceedings of FASE '95: Colloquium on Formal Approaches in Software Engineering*, (Aarhus, Denmark, May 22-26, 1995), *Lecture Notes in Computer Science*, Vol. 915, P.D. Mosses, M. Nielsen, and M.I. Schwartzbach (eds.), Springer-Verlag, New York, NY, 1995, pp. 651-665.

Invited for special submission to *Theoretical Computer Science (TCS)***.**

Reps, T., Horwitz, S., and Sagiv, M., Precise interprocedural dataflow analysis via graph reachability. In *Conference Record of the Twenty-Second ACM Symposium on Principles of Programming Languages*, (San Francisco, CA, Jan. 23-25, 1995), ACM, New York, NY, 1995, pp. 49-61.

Reps, T., Horwitz, S., Sagiv, M., and Rosay, G., Speeding up slicing. In *SIGSOFT '94: Proceedings of the Second ACM SIGSOFT Symposium on the Foundations of Software Engineering*, (New Orleans, LA, December 7-9, 1994), *ACM SIGSOFT Software Engineering Notes 19*, 5 (December 1994), pp. 11-20.

Received an ACM SIGSOFT Retrospective Impact Paper Award (2011).

Reps, T., Solving demand versions of interprocedural analysis problems. In *Proceedings of the Fifth International Conference on Compiler Construction*, (Edinburgh, Scotland, April 7-9, 1994), *Lecture Notes in Computer Science*, Vol. 786, P. Fritzson (ed.), Springer-Verlag, New York, NY, 1994, pp. 389-403.

Ramalingam, G. and Reps, T., An incremental algorithm for maintaining the dominator tree of a reducible flowgraph. In *Conference Record of the Twenty-First ACM Symposium on Principles of Programming Languages*, (Portland, OR, Jan. 16-19, 1994), pp. 287-296.

Reps, T., Scan grammars: Parallel attribute evaluation via data-parallelism. In *Proceedings of the Fifth ACM Symposium on Parallel Algorithms and Architectures*, (Velen, Germany, June 30 - July 2, 1993), pp. 367-376.

Ramalingam, G. and Reps, T., Modification algebras. In *Proceedings of the Second International Conference on Algebraic Methodology and Software Technology (AMAST)*, (Iowa City, Iowa, May 22-24, 1991).

Ramalingam, G. and Reps, T., A theory of program modifications. In *Proceedings of the Colloquium on Combining Paradigms for Software Development*, (Brighton, UK, April 8-12, 1991), *Lecture Notes in Computer Science*, Vol. 494, S. Abramsky and T.S.E. Maibaum (eds.), Springer-Verlag, New York, NY, 1991, pp. 137-152.

Yang, W., Horwitz, S., and Reps, T., A program integration algorithm that accommodates semantics-preserving transformations. In *SIGSOFT '90: Proceedings of the Fourth ACM SIGSOFT Symposium on Software Development Environments*, (Irvine, CA, December 3-5, 1990), *ACM Software Engineering Notes 15*, 6 (December 1990), pp. 133-143.

Invited for special submission to *ACM Trans. on Softw. Eng. and Methodology (TOSEM)***.**

Reps, T., Algebraic properties of program integration. In *Proceedings of the Third European Symposium on Programming*, (Copenhagen, Denmark, May 15-18, 1990), *Lecture Notes in Computer Science*, Vol. 432, N. Jones (ed.), Springer-Verlag, New York, NY, 1990, pp. 326-340.

Invited for special submission to *Science of Computer Programming (SCP)*.

Reps, T. and Bricker, T., Illustrating interference in interfering versions of programs. In *Proceedings of the Second International Workshop on Software Configuration Management*, (Princeton, NJ, October 24-27, 1989), *ACM Software Engineering Notes 17*, 7 (November 1989), pp. 46-55.

Horwitz, S., Pfeiffer, P., and Reps, T., Dependence analysis for pointer variables. In *SIGPLAN '89: Proceedings of the ACM Conference on Programming Language Design and Implementation* (PLDI),

(Portland, OR, June 21-23, 1989), *ACM SIGPLAN Notices 24*, 7 (July 1989), pp. 28-40.

Reps, T. and Yang, W., The semantics of program slicing and program integration. In *Proceedings of the Colloquium on Current Issues in Programming Languages*, (Barcelona, Spain, March 13-17, 1989), *Lecture Notes in Computer Science*, Vol. 352, J. Diaz and F. Orejas (eds.), Springer-Verlag, New York, NY, 1989, pp. 360-374.

Horwitz, S., Reps, T., and Binkley, D., Interprocedural slicing using dependence graphs. In *SIGPLAN '88: Proceedings of the ACM Conference on Programming Language Design and Implementation* (PLDI), (Atlanta, GA, June 22-24, 1988), *ACM SIGPLAN Notices 23*, 7 (July 1988), pp. 35-46.

Selected for inclusion in a special SIGPLAN collection of the 50 most influential papers from the SIGPLAN Conference on Programming Language Design and Implementation from 1979 to 1999.

Reps, T., Horwitz, S., and Prins, J., Support for integrating program variants in an environment for programming in the large. In *Proceedings of the International Workshop on Software Version and Configuration Control*, (Grassau, W. Germany, Jan. 27-29, 1988), *Berichte des German Chapter of the ACM*, Vol. 30, J.F.H. Winkler (ed.), B.G. Teubner, Stuttgart, W. Germany, 1988, pp. 197-216.

Horwitz, S., Prins, J., and Reps, T., Integrating non-interfering versions of programs. In *Conference Record of the Fifteenth ACM Symposium on Principles of Programming Languages*, (San Diego, CA, January 13-15, 1988), ACM, New York, NY, 1988, pp. 133-145.

Horwitz, S., Prins, J., and Reps, T., On the adequacy of program dependence graphs for representing programs. In *Conference Record of the Fifteenth ACM Symposium on Principles of Programming Languages*, (San Diego, CA, January 13-15, 1988), ACM, New York, NY, 1988, pp. 146-157.

Reps, T., Marceau, C., and Teitelbaum, T., Remote attribute updating for language-based editors. In *Conference Record of the Thirteenth ACM Symposium on Principles of Programming Languages*, (St. Petersburg, FL, January 13-15, 1986), ACM, New York, NY, 1986, pp. 1-13.

Reps, T. and Teitelbaum, T., The Synthesizer Generator. In *Proceedings of the ACM SIGSOFT/SIGPLAN Software Engineering Symposium on Practical Software Development Environments*, (Pittsburgh, PA, April 23-25, 1984), *ACM SIGPLAN Notices 19*, 5 (May 1984), pp. 42-48.

Received an ACM SIGSOFT Retrospective Impact Paper Award (2010).

Reps, T. and Alpern, B., Interactive proof checking. In *Conference Record of the Eleventh ACM Symposium on Principles of Programming Languages*, (Salt Lake City, Utah, January 15-18, 1984), ACM, New York, NY, 1984, pp. 36-45.

Reps, T., Optimal-time incremental semantic analysis for syntax-directed editors. In *Conference Record of the Ninth ACM Symposium on Principles of Programming Languages*, (Albuquerque, NM, January 25-27, 1982), ACM, New York, NY, 1982, pp. 169-176.

Invited for special submission to *ACM Trans. on Program. Lang. and Syst. (TOPLAS)*.

Teitelbaum, T., Reps, T., and Horwitz, S., The why and wherefore of the Cornell Program Synthesizer. In *Proceedings of the ACM SIGPLAN/SIGOA Symposium on Text Manipulation*, (Portland, OR, June 8-10, 1981), *ACM SIGPLAN Notices 16*, 6 (June 1981), pp. 8-16.

Demers, A., Reps, T., and Teitelbaum, T., Incremental evaluation for attribute grammars with application to syntax-directed editors. In *Conference Record of the Eighth ACM Symposium on Principles of Programming Languages*, (Williamsburg, VA, January 26-28, 1981), ACM, New York, NY, 1981, pp. 105-116.

Other Conference Publications

Henkel, J., Silva, D., Teixeira, L., d'Amorim, M., and Reps, T.W., Shipwright: A human-in-the-loop system for Dockerfile repair. In *Proc. Int. Conf. on Software Engineering (ICSE, Companion Volume)*, 2021, pp. 198-199.

Reps, T. and Thakur, A., Through the lens of abstraction. In *Proc. of the High Confidence Software and Systems Conference* (HCSS), May 2014.

Horwitz, S. and Reps, T., Language-based tools for software engineering. In *Proceedings of the DARPA/ISTO Software Technology Conference*, (Santa Monica, CA, April 28-30, 1992), Defense Advanced Research Projects Agency, Arlington, VA, 1992, pp. 106-110.

Ramalingam, G. and Reps, T., Modification algebras (Extended abstract). In *Preliminary Proceedings of the Second International Conference on Algebraic Methodology and Software Technology (AMAST)*, (Iowa City, Iowa, May 22-24, 1991), Department of Computer Science, University of Iowa, Iowa City, IA, pp. 323-327.

Reps, T., Demonstration of a prototype tool for program integration. In *Proceedings of the DARPA/ISTO Software Principal Investigators' Meeting*, (Warwick, RI, Feb./March 1991), Defense Advanced Research Projects Agency, Arlington, VA, 1991.

Reps, T., Static-semantic analysis in language-based editors. In *Digest of Papers of the IEEE Spring CompCon 83*, (San Francisco, CA, March 1-3, 1983), IEEE Computer Society, Washington, DC, 1983, pp. 411-414.

Book Chapters

Ramalingam, G. and Reps, T., Semantics of program representation graphs. In *Semantics to Computer Science: Essays in Honour of Gilles Kahn*, Y. Bertot, G. Huet, J.-J. Levy, and G. Plotkin (eds.), Cambridge University Press, 2009.

Reps, T., Sagiv, M., and Wilhelm, R., Shape analysis and applications. Chapter 12 in *The Compiler Design Handbook: Optimizations and Machine Code Generation*, 2nd Edition, CRC Press, 2007.

Clarke, E., Kroening, D., and Reps, T., Static analysis to enhance the power of model checking for concurrent software. In *Department of Defense Sponsored Information Security Research: New Methods for Protecting Against Cyber Threats*, Wiley, 2007, 349-360.

Balakrishnan, G., Christodorescu, M., Ganapathy, V., Giffin, J.T., Rubin, S., Wang, H., Jha, S., Miller, B.P., and Reps, T., Analysis of COTS for Security Vulnerability Remediation In *Department of Defense Sponsored Information Security Research: New Methods for Protecting Against Cyber Threats*, Wiley, 2007, 375-380.

Sagiv, M., Reps, T.W., Wilhelm, R., and Yahav, E., On the utility of canonical abstraction. In *Engineering Theories of Software Intensive Systems*, M. Broy, J. Gruenbauer, T. Hoare, D. Harel (eds.), Kluwer Academic Publishers, Dordrecht, The Netherlands, 2005, 215-253.

Reps, T., Sagiv, M., and Wilhelm, R., Shape analysis and applications. In *The Compiler Design Handbook: Optimizations and Machine Code Generation*, CRC Press, 2002, 175-217.

Reps, T., Demand interprocedural program analysis using logic databases, In *Applications of Logic Databases*, R. Ramakrishnan (ed.), Kluwer Academic Publishers, Boston, MA, 1994, pp. 163-196.

Reprinted in Collections

Reps, T.W. and Rall, L.B., Computational divided differencing and divided-difference arithmetics, In *Automatic Program Development: A Tribute to Robert Paige*, Danvy, O., Mairson, H., Henglein, F., and Pettorossi, A. (Eds.), Springer Verlag, 2008.

Revised version of Reps, T.W. and Rall, L.B., Computational divided differencing and divided-difference arithmetics. *Higher-Order and Symbolic Computation 16*, 1-2 (June 2003).

Loginov, A., Reps, T., and Sagiv, M., Abstraction refinement via inductive learning, In *Department of Defense Sponsored Information Security Research: New Methods for Protecting Against Cyber Threats*, Wiley, 2007, 361-374.

Reprinted from *Proc. Computer-Aided Verification*, 2005.

Reps, T., Balakrishnan, G., Lim, J., and Teitelbaum, T., A next-generation platform for analyzing executables. In *Malware Detection*, Advances in Information Security Series, Vol. 27, M. Christodorescu, S. Jha, D. Maughan, D. Song, and C. Wang (eds.), Springer-Verlag, 2006, pp. 43-61.

Reprinted from *Proc. 3rd Asian Symposium on Programming Languages and Systems*, (Tsukuba, Japan, Nov. 3-5, 2005).

Horwitz, S., Reps, T., and Binkley, D., Interprocedural slicing using dependence graphs. *20 Years of the ACM SIGPLAN Conference on Programming Language Design and Implementation (1979 - 1999): A Selection*, K.S. McKinley, ed., *ACM SIGPLAN Notices 39*, 4 (April 2004), 232-243.

Reprinted from *SIGPLAN '88: Proceedings of the ACM Conference on Programming Language Design and Implementation* (PLDI), (Atlanta, GA, June 22-24, 1988), *ACM SIGPLAN Notices 23*, 7 (July 1988), pp. 35-46.

A retrospective on the paper appeared as

Horwitz, S., Reps, T., and Binkley, D., Retrospective: Interprocedural slicing using dependence graphs. *20 Years of the ACM SIGPLAN Conference on Programming Language Design and Implementation (1979 - 1999): A Selection*, K.S. McKinley, ed. *ACM SIGPLAN Notices 39*, 4 (April 2004), 229-231.

Horwitz, S., Reps, T., and Binkley, D., Interprocedural slicing using dependence graphs. In *Software Change Impact Analysis*, S.A. Bohner and R.S. Arnold (eds.), IEEE Computer Society, Los Alamitos, CA, 1996.

Reprinted from *ACM Trans. on Program. Lang. and Syst. (TOPLAS) 12*, 1 (January 1990), 26-60.

Horwitz, S., Reps, T., and Binkley, D., Interprocedural slicing using dependence graphs. In *Software Merging and Slicing*, V. Berzins (ed.), IEEE Computer Society, Los Alamitos, CA, pp. 10-44.

Reprinted from *ACM Trans. on Program. Lang. and Syst. (TOPLAS) 12*, 1 (January 1990), 26-60.

Horwitz, S., Prins, J., and Reps, T., Integrating non-interfering versions of programs. In *Software Merging and Slicing*, V. Berzins (ed.), IEEE Computer Society, Los Alamitos, CA, pp. 137-179.

Reprinted from *ACM Trans. on Program. Lang. and Syst. (TOPLAS) 11*, 3 (July 1989), 345-387.

Ramalingam, G. and Reps, T., A theory of program modifications. In *Software Merging and Slicing*, V. Berzins (ed.), IEEE Computer Society, Los Alamitos, CA, 90-105.

Reprinted from *Proceedings of the Colloquium on Combining Paradigms for Software Development*, (Brighton, UK, April 8-12, 1991), *Lecture Notes in Computer Science*, Vol. 494, S. Abramsky and T.S.E. Maibaum (eds.), Springer-Verlag, New York, NY, 1991, pp. 137-152.

Reps, T. and Teitelbaum, T., Language processing in program editors. In *Language Architectures and Programming Environments*, T. Ichikawa and H. Tsubotani (eds.), The World Scientific Publishing Company, Singapore, 1992, pp. 146-169.

Reprinted from *IEEE Computer 20*, 11 (November 1987), 29-40.

Teitelbaum, T. and Reps, T., The Cornell Program Synthesizer: A syntax-directed programming environment. In *Interactive Programming Environments*, D. Barstow, E. Sandewall, and H. Shrobe (eds.), McGraw-Hill, 1984, pp. 97-116.

Reprinted from *Communications of the ACM 24*, 9 (September 1981), 563-573.

Teitelbaum, T., Reps, T., and Horwitz, S., The why and wherefore of the Cornell Program Synthesizer. In *Software Development Environments*, A.I. Wasserman (ed.), IEEE Computer Society, Washington, DC, 1981, 64-72.

Reprinted from *Proceedings of the ACM SIGPLAN/SIGOA Symposium on Text Manipulation*, (Portland, OR, June 8-10, 1981), *ACM SIGPLAN Notices 16*, 6 (June 1981), pp. 8-16.

Magazine Articles

Anderson, P. and Reps, T., When good compilers go bad, or What you see is not what you execute. In *Embedded Systems Design*, Feb. 2010.

Anderson, P., Reps, T., Teitelbaum, T., and Zarins, M., Tool support for fine-grained software inspection. *IEEE Software 20*, 4 (2003), 42-50.

Reps, T. and Teitelbaum, T., Language processing in program editors. *IEEE Computer 20*, 11 (November 1987), 29-40.

Other Publications and Reports

Zhou, C., Wang, D., and Reps, T., The best of times, the worst of times: Moment-based analysis of probabilistic cost structures. arXiv:2607.25361, July 2026.

Liu, H., Sala, F., Reps, T., and Murali, A., Counterexample guided learning in the large using reasoning agents. arXiv:2606.11521, June 2026.

Sistla, M., Chaudhuri, S., and Reps, T., Do CFLOBDDs actually make use of linear structure? arXiv:2605.15552, May 2026.

Trinh, X.D., Sistla, M., Yu, N., and Reps, T., Manjushri: A tool for equivalence checking of quantum circuits. arXiv:2601.22372, January 2026.

Jain, A., Yao, X., Reps, T., and Jermaine, C., MHA-RAG: Improving efficiency, accuracy, and consistency by encoding exemplars as soft prompts. arXiv:2510.05363, October 2025.

Dramko, E., Xiong, Y., Zhu, Y., Hautier, G., Reps, T., Jermaine, C., and Kyrillidis, A., ADAPT: Lightweight, long-range machine learning force fields without graphs. arXiv:2509.24115, September 2025.

Kahn, D.M., Hoffmann, J., Reps, T., and Grosen, J., Efficient cost bounds with linear maps. arXiv:2509.22982, September 2025.

Fang, R., Kincaid, Z., and Reps, T. Software model checking via summary-guided search (Extended version). arxiv:2508.15137, August 2025.

Wang, Z., Weng, S., Alhanahnah, M., Jha, S., and Reps, T., PEA: Enhancing LLM performance on computational-reasoning tasks. arXiv:2502.10938, February 2025.

Jain, A., Adiole, C., Chaudhuri, S., Reps, T., and Jermaine, C., Coarse-tuning models of code with reinforcement learning feedback. arXiv:2305.18341, May 2023.

Wang, Z., Liblit, L., and Reps, T., TOFU: Target-oriented fuzzer. arXiv:2004.14375, April 2020.

Henkel, J., Lahiri, S.K., Liblit, B., and Reps, T.W., Enabling open-world specification mining via unsupervised learning. arXiv:1904.12098, Apr. 2019.

Kashyap, B., Brown, D.B., Liblit, B., Melski, D., and Reps, T., Source Forager: A search engine for similar source code. arXiv:1706.02769, June 2017.

Amodio, M., Chaudhuri, S., and Reps, T., Neural attribute machines for program generation. arXiv:1705.09231v2, May 2017.

Henry, J., Thakur, A., Kidd, N., and Reps, T., Dissolve: A distributed SAT solver based on Staalmarck's method. TR-1839, Computer Sciences Department, University of Wisconsin, Madison, WI, Aug. 2016.

Srinivasan, V. and Reps, T., Slicing machine code. TR-1824, Computer Sciences Department, University of Wisconsin, Madison, WI, Oct. 2015.

Srinivasan, V. and Reps, T., Synthesis of machine code from semantics. TR-1814, Computer Sciences Department, University of Wisconsin, Madison, WI, Feb. 2015.

Kroening, D., Reps, T.W., Seshia, S.A., and Thakur, A. (eds.), Decision procedures and abstract interpretation (Dagstuhl Seminar 14351), Dagstuhl Seminar Report 4, 8 (Dec. 2014) International Conference and Research Center for Computer Science (IBFI), Schloss Dagstuhl, Wadern, Germany, 2014.

Aung, M., Horwitz, S., Joiner, R., and Reps, T., Specialization slicing. In *Proc. ACM Conf. on Programming Language Design and Implementation* (PLDI), (Edinburgh, UK, June 9-11, 2014). (Journal-first conference paper; see the TOPLAS 2014 paper of the same title.)

Sharma, T., Thakur, A., and Reps, T., An abstract domain for bit-vector inequalities. TR-1789, Computer Sciences Department, University of Wisconsin, Madison, WI, April 2013.

Srinivasan, V.K. and Reps, T., Software-architecture recovery from machine code. TR-1781, Computer Sciences Department, University of Wisconsin, Madison, WI, March 2013.

King, A.M., Mycroft, A., Reps, T.W., and Simon, A. (eds.), Analysis of executables: Benefits and challenges (Dagstuhl Seminar 12051) Dagstuhl Seminar Report 2, 1 (May 2012) International Conference and Research Center for Computer Science (IBFI), Schloss Dagstuhl, Wadern, Germany, 2012.

Harris, W.R., Farley, B., Jha, S., and Reps, T., Programming for a capability system via safety games, TR-1705, Computer Sciences Department, University of Wisconsin, Madison, WI, March 2012.

Prabhu, P., Reps, T., Lal, A., and Kidd, N., Verifying concurrent programs via bounded context-switching and induction. TR-1701, Computer Sciences Department, University of Wisconsin, Madison, WI, November 2011.

Driscoll, E., Thakur, A., Burton, A., and Reps, T., WALi: Nested-word automata. TR-1675r, Computer Sciences Department, University of Wisconsin, Madison, WI, July 2010; revised Sept. 2011.

Lim, J. and Reps, T., BCE: Extracting botnet commands from bot executables. TR-1668, Computer Sciences Department, University of Wisconsin, Madison, WI, February 2010.

Lal, A., Lim, J., and Reps, T., McDash: Refinement-based property verification for machine code. TR-1659, Computer Sciences Department, University of Wisconsin, Madison, WI, June 2009.

Lev-Ami, T., Sagiv, M., Reps, T., and Gulwani, S., Backward analysis for inferring quantified preconditions. Tech. Rep. TR-2007-12-01, Tel Aviv University, Dec. 2007.

Kidd, N., Moore, K., Wood, D., and Reps, T., Towards the analysis of transactional software. TR-1624, Computer Sciences Department, University of Wisconsin, Madison, WI, October 2007.

Kidd, N., Reps, T., Dolby, J., and Vaziri, M., Static detection of atomic-set serializability violations. TR-1623, Computer Sciences Department, University of Wisconsin, Madison, WI, October 2007.

Kidd, N., Lal, A., and Reps, T., Advanced querying for property checking. TR-1621, Computer Sciences Department, University of Wisconsin, Madison, WI, October 2007.

Reps, T., Sagiv, M., and Bauer, J., An appreciation of the work of Reinhard Wilhelm. In *Program Analysis and Compilation, Theory and Practice: Essays Dedicated to Reinhard Wilhelm*, Springer-Verlag, 2007.

Lal, A., Touili, T., Kidd, N., and Reps, T. Weighted pushdown systems and weighted transducers. TR-1581, Computer Sciences Department, University of Wisconsin, Madison, WI, Oct. 2006.

Lal, A., Kidd, N., Reps, T., and Touili, T., Abstract error projection. TR-1579, Computer Sciences Department, University of Wisconsin, Madison, WI, Sept. 2006.

Reps, T.W., Teitelbaum, T., Anderson, P., and Melski, D., Static analysis of binary executable code. Proceedings of *Defining the State of the Art in Software Security Tools Workshop*, (Gaithersburg, MD, Aug. 10-11, 2005), NIST Special Publication 500-264, Nat. Inst. of Standards and Technology, Gaithersburg, MD, Sept. 2005.

Balakrishnan, G., Reps, T., Kidd, N., Lal, A., Lim, J., Melski, D., Gruian, R., Yong, S., Chen, C.-H., and Teitelbaum, T., Model checking x86 executables with CodeSurfer/x86 and WPDS++. In *Proc. Workshop on the Evaluation of Software Defect Detection Tools*, June 2005.

Yahav, E., Reps, T., and Sagiv, M., LTL model checking for systems with unbounded number of dynamically created threads and objects. TR-1424, Computer Sciences Department, University of Wisconsin, Madison, WI, March 2001.

Chandra, S. and Reps, T., Physical type checking for C. Bell Labs. Tech. Rep. BL0113590-990302-04, Lucent Technologies, Inc., Naperville, IL, Mar. 1999.

Siff, M., Chandra, S., Ball, T., Kunchithapadam, K., and Reps, T., Coping with type casts in C. Bell Labs. Tech. Rep. BL0113590-990202-03, Lucent Technologies, Inc., Naperville, IL, Feb. 1999.

Clarke, E.M., Fujita, M., Rajan, P.S., Reps, T., Shankar, S., and Teitelbaum, T., Program slicing for design automation: An automatic technique for speeding-up hardware design, simulation, testing, and verification. Unpublished report, October 1998.

Mueller, H., Reps, T., and Snelting, G. (eds.), Program comprehension and software reengineering. Dagstuhl Seminar Report, International Conference and Research Center for Computer Science (IBFI), Schloss Dagstuhl, Wadern, Germany, 1998.

The Wisconsin Program-Slicing Tool 1.0, Reference Manual. Computer Sciences Department, University of Wisconsin-Madison, August 1997.

Das, M. and Reps, T. BTA termination using CFL-reachability. TR-1329, Computer Sciences Department, University of Wisconsin, Madison, WI, November 1996.

Horwitz, S., Reps, T., and Sagiv, M., Demand interprocedural dataflow analysis. TR-1283, Computer Sciences Department, University of Wisconsin, Madison, WI, August 1995.

van Leeuwen, J., Mehlhorn, K., and Reps, T. (eds.), Incremental computation and dynamic algorithms. Dagstuhl Seminar Report 88, International Conference and Research Center for Computer Science (IBFI), Schloss Dagstuhl, Wadern, Germany, 1994.

Reps, T., Sagiv, M., and Horwitz S., Interprocedural dataflow analysis via graph reachability. TR 94-14, Datalogisk Institut, University of Copenhagen, Copenhagen, Denmark, April 1994.

Reps, T., *The Wisconsin Program-Integration System Reference Manual: Release 2.0*. Computer Sciences Department, University of Wisconsin, Madison, WI, July 1993.

Ramalingam, G. and Reps, T., A categorized bibliography on incremental computation. In *Conference Record of the Twentieth ACM Symposium on Principles of Programming Languages*, (Charleston, SC, Jan. 11-13, 1993), ACM, New York, NY, 1993, pp. 502-510.

Reps, T., Incremental computation. Unpublished tutorial notes, 1993. (Presented at the Twentieth ACM Symposium on Principles of Programming Languages, (Charleston, SC, Jan. 11-13, 1993).)

Klint, P., Reps, T., and Snelting, G. (eds.), Programming environments. Dagstuhl Seminar Report 34, International Conference and Research Center for Computer Science (IBFI), Schloss Dagstuhl, Wadern, Germany, 1992.

Ramalingam, G. and Reps, T., New programs from old. TR-1057, Computer Sciences Department, University of Wisconsin Madison, WI, November 1991.

Binkley, D., Horwitz, S., and Reps, T., Identifying semantic differences in programs with procedures (Extended abstract). Computer Sciences Department, University of Wisconsin, Madison, WI, September 1991.

Ball, T., Horwitz, S., and Reps, T., Correctness of an algorithm for reconstituting a program from a dependence graph. TR-947, Computer Sciences Department, University of Wisconsin, Madison, WI, July 1990.

Binkley, D., Horwitz, S., and Reps, T., The multi-procedure equivalence theorem. TR-890, Computer Sciences Department, University of Wisconsin, Madison, WI, November 1989.

Yang, W., Horwitz, S., and Reps, T., Detecting program components with equivalent behaviors. TR-840, Computer Sciences Department, University of Wisconsin, Madison, WI, April 1989.

Horwitz, S., Prins, J., and Reps, T., On the suitability of dependence graphs for representing programs. Computer Sciences Department, University of Wisconsin, Madison, WI, August 1988.

Reps, T., Guest editor's introduction. *ACM Trans. on Program. Lang. and Syst. (TOPLAS) 8*, 4 (October 1986), 417-418.

Teitelbaum, T. and Reps, T., Release of the Synthesizer Generator. *ACM Software Engineering Notes 11*, 1 (January 1986), 109-110.

Teitelbaum, T. and Reps, T., Immediate computation. *Engineering: Cornell Quarterly*, College of Engineering, Cornell University, Ithaca, NY, November 1985.

Teitelbaum, T. and Reps, T., Correspondence: On the value of syntax-directed editors. *Communications of the ACM 25*, 5 (May 1982), 351-352.

SOFTWARE

Kidd, N., Lal, A., and Reps, T., *WALi: The Weighted Automaton Library*, December 2007. (Available for download from http://www.cs.wisc.edu/wpis/wpds/wali/.)

Kidd, N., Reps, T., Melski, D., and Lal, A., *WPDS++: A C++ Library for Weighted Pushdown Systems*, 2004. (Available for download from http://www.cs.wisc.edu/wpis/wpds/wpds++/.)

Reps, T., Rosay, G., and Horwitz, S., *The Wisconsin Program-Slicing Tool*. Release 1.0, August 1997; Release 1.1, January 1998.

Reps, T., Bricker, T., Rosay, G., *et al.*, *The Wisconsin Program-Integration System*. Release 0.5, April 1990; Release 1.0, April 1992. Release 2.0, July 1993. Licensed to 9 sites.

Reps, T., Teitelbaum, T., *et al.*, *The Synthesizer Generator*. Release 1.0, December 1985; Release 2.0, July 1987; Release 3.0, April 1989. Licensed to approximately 325 sites.

Teitelbaum, T., Reps, T., *et al.*, *The Cornell Program Synthesizer*. Version 1, June 1979; Version 1.02, September 1980; Version 1.03, September 1981. Licensed to approximately 110 sites.

PATENTS

Reps, T., Horwitz, S., and Binkley, D., Interprocedural slicing of computer programs using dependence graphs. U.S. Patent Number 5,161,216, issued November 3, 1992. (Assignee: Wisconsin Alumni Research Foundation, Madison, WI.)

Alhanahnah, M., Rastogi, V., Jha, S., and Reps, T., Computer implemented program specialization. U.S. Patent Number 12,079,607, issued September 3, 2024. (Assignee: Wisconsin Alumni Research Foundation, Madison, WI.)

LECTURES

Invited Addresses

1. Semantics-based program integration. Second European Symposium on Programming (ESOP ′88), Nancy, France, March 24, 1988.
2. The use of program dependence graphs in software engineering. Fourteenth International Conference on Software Engineering, Melbourne, Australia, May 15, 1992.
3. The use of program profiling in software testing. Informatik ′97, Aachen, Germany, Sept. 24, 1997.
4. Program analysis via graph reachability. Fourteenth International Logic Programming Symposium, Port Jefferson, NY, Oct. 14, 1997.
5. Program analysis via graph reachability. Distinguished Lecture Series, Department of Computer and Information Science, Ohio State University, Columbus, OH, April 1, 1998.
6. Static analysis for software-engineering tools: Issues and Opportunities. University of Washington/Microsoft Research Workshop on Technologies to Improve Software Development, Seattle, WA, August 2, 1999.

7. Static analysis and linked data structures. Distinguished Lecture Series, University of Louisiana at Lafayette, Lafayette, LA, April 26, 2002.
8. Static program analysis via 3-valued logic. Static Analysis Symposium. Madrid, Spain, Sept. 18, 2002.
9. Finite differencing of logical formulas for static analysis, Workshop on Dynamic Algorithms and Applications. New Orleans, LA, Jan. 14, 2004.
10. Weighted pushdown systems and their applications. 3rd Int. Workshop on Automated Verification of Infinite-State Systems. Barcelona, Spain, April 3, 2004.
11. CodeSurfer/x86: A platform for analyzing x86 executables. Infosec Research Council, Arlington, VA, July 8, 2004.
12. Static program analysis via three-valued logic. Int. Conf. on Computer-Aided Verification, Boston, MA, July 16, 2004.
13. Static analysis of executables. Distinguished Lecture Series, U. Minn., Nov. 8, 2004.
14. Abstraction and analysis for dynamically changing resources. IFIP Working Conference on Verified Software: Theories, Tools, Experiments, Zurich, Switzerland, Oct. 11, 2005.
15. A next-generation platform for analyzing executables. 3rd Asian Symposium on Programming Languages and Systems, Tsukuba, Japan, Nov. 5, 2005.
16. ----------. ACM SIGPLAN Workshop on Partial Evaluation and Program Manipulation (PEPM) Charleston, SC, Jan. 10, 2006.
17. WYSINWYX: What You See Is Not What You eXecute. Infosec Research Council, Washington, DC, March 9, 2006.
18. DIVINE: DIscovering Variables IN Executables. Eighth Int. Conf. on Verification, Model Checking and Abstract Interpretation (VMCAI), (Nice, France, Jan. 2007).
19. WYSINWYX: What You See Is Not What You eXecute. Workshop on Compiler Techniques and Applications, New Delhi, India, Dec. 11, 2007.
20. Program analysis using weighted pushdown systems. 27th Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS), New Delhi, India, Dec. 14, 2007.
21. WYSINWYX: What You See Is Not What You eXecute. “Unifying” invited talk at the European Joint Conferences on Theory and Practice of Software (ETAPS), Budapest, Hungary, Apr. 2, 2008.
22. ----------. Distinguished Lecture Series, Max Planck Institute for Software Systems, Saarbruecken, Germany, May 5, 2008.
23. ----------. Distinguished Lecture Series, Department of Computer and Information Science, Univ. of Pennsylvania, April 2, 2009.
24. ----------. Keynote talk, Int. Symp. on Software Testing and Analysis (ISSTA), Chicago, IL, July 21, 2009.
25. ----------. Distinguished Lecture Series, Computer Science Department, Stony Brook University. Sept. 11, 2009.
26. There’s plenty of room at the bottom: Analyzing and verifying machine code. Systems Distinguished Lecture Series, Department of Computer Science, Purdue University, Dec. 6, 2010.
27. Automating abstract interpretation. Triangle CS Distinguished Lecture Series, Department of Computer Science, University of North Carolina, Apr. 6, 2015.
29. ----------. Keynote talk, IEEE/ACM Conference on Automated Software Engineering (ASE), Nov. 2015
29. ----------. IAS Distinguished Lecture, HKUST Jockey Club Institute for Advanced Study, Hong Kong, Jan. 5, 2016.

30. ----------. Int. Conference on Verification, Model Checking, and Abstract Interpretation (VMCAI), Jan. 17, 2016.
31. Program analyses using Newton's method. Invited talk, International Conference on Networked Systems (NETYS). Essaouira, Morocco, May 11, 2018.
32. CFLOBDDs: Context-Free-Language Ordered Binary Decision Diagrams. Distinguished Lecture, School of Computer Science, Peking University, April 27, 2023.
32. We will publish no algorithm before its time. Keynote talk, 40th IEEE/ACM International Conference on Automated Software Engineering (ASE), Seoul, South Korea, Nov. 17, 2025.

Tutorials

1. Incremental computation. Twentieth ACM Symposium on Principles of Programming Languages, Charleston, SC, January 10, 1993.
2. Program analysis via graph reachability. ACM Conference on Programming Language Design and Implementation (PLDI), Vancouver B.C., Canada, June 18, 2000.
3. There's plenty of room at the bottom: Analyzing and verifying machine code. 22nd Int. Conf. on Computer-Aided Verification (CAV), Edinburgh, Scotland. July 2010.
4. Introduction to algebraic program analysis. ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages (POPL), Los Angeles, CA, Jan. 2018 (with Zachary Kincaid).
5. Tips on writing a research paper, PL Mentoring Workshop (PLMW@PLDI 2021), June 2021.
6. Algebraic program analysis. Int. Conf. on Computer Aided Verification (CAV), July 2021 (with Zachary Kincaid).
7. Program analysis via graph reachability: Past, present, and future. ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages (POPL), 2022. (with Qirun Zhang).
8. Roses are blue, violets are red; With future AI, is research dead? PL Mentoring Workshop (PLMW@PLDI 2026), June 2026.

Presentations at Symposia, Workshops, etc.

1. The why and wherefore of the Cornell Program Synthesizer. ACM SIGPLAN/SIGOA Symposium on Text Manipulation, Portland, OR, June 8, 1981.
2. Optimal-time incremental semantic analysis for syntax-directed editors. Ninth ACM Symposium on Principles of Programming Languages, Albuquerque, NM, January 26, 1982.
3. Static-semantic analysis in language-based editors. IEEE Spring CompCon 83, San Francisco, CA, March 3, 1983.
4. ----------. International course on syntax-directed editing, Aussois, France, April 20, 1983.
5. Optimal-time incremental semantic analysis for syntax-directed editors. International course on syntax-directed editing, Aussois, France, April 21, 1983.
6. Interactive proof checking. Eleventh ACM Symposium on Principles of Programming Languages, Salt Lake City, Utah, January 16, 1984.
7. The Synthesizer Generator. ACM SIGSOFT/SIGPLAN Software Engineering Symposium on Practical Software Development Environments, Pittsburgh, PA, April 23, 1984.
8. ----------. Workshop on Programs as Data Objects, Copenhagen, Denmark, October 18, 1985.
9. Remote attribute updating for language-based editors. Thirteenth ACM Symposium on Principles of Programming Languages, St. Petersburg, FL, January 13, 1986.
10. The Synthesizer Generator. IBM Academic Information Systems 1986 University Study Conference, Ft. Lauderdale, FL, November 18, 1986.

11. Integrating non-interfering versions of programs. IFIP Working Group 2.4, Niagara-on-the-Lake, Ontario, May 5, 1987.
12. ----------. First Meeting of the Midwest Society for Programming Languages and Systems, Chicago, IL, October 24, 1987.
13. Semantics-based program integration. NSF–Industry–University Symposium, IBM T. J. Watson Research Center, Yorktown Heights, NY, April 25, 1989.
14. ----------. First Annual Meeting of the David and Lucile Packard Fellowships Program, Monterey, CA, September 8, 1989.
15. Algebraic properties of program integration. Fifth Meeting of the Midwest Society for Programming Languages and Systems, Chicago, IL, October 21, 1989.
16. Illustrating interference in interfering versions of programs. Second International Workshop on Software Configuration Management, Princeton, NJ, October 25, 1989.
17. Algebraic properties of program integration. Third European Symposium on Programming (ESOP ′90), Copenhagen, Denmark, May 18, 1990.
18. Semantics-based program integration. DARPA/ISTO Software Principal Investigators' Meeting, Warwick, R.I., February 27, 1991.
19. The computational complexity of incremental algorithms. Seminar on Programming Environments, International Conference and Research Center for Computer Science (IBFI) at Dagstuhl Castle, Wadern, Germany, March 9, 1992.
20. Semantics-based program integration. International Workshop on Feature Interactions in Telecommunications Software Systems, St. Petersburg, FL, December 4, 1992.
21. Scan grammars: Parallel attribute evaluation via data-parallelism. Twelfth Meeting of the Midwest Society for Programming Languages and Systems, Iowa City, IA, April 10, 1993.
22. ----------. Fifth ACM Symposium on Parallel Algorithms and Architectures, Velen, Germany, July 2, 1993.
23. Bounded incremental computation. Fifth Annual Meeting of the David and Lucile Packard Fellowships Program, Monterey, CA, September 11, 1993.
24. Solving demand versions of interprocedural analysis problems. Fifth International Conference on Compiler Construction, Edinburgh, Scotland, April 9, 1994.
25. From "incremental attribute evaluation" to "bounded incremental computation". Seminar on Incremental Computation and Dynamic Algorithms, International Conference and Research Center for Computer Science (IBFI) at Dagstuhl Castle, Wadern, Germany, May 4, 1994.
26. Incremental interprocedural dataflow analysis. Seminar on Incremental Computation and Dynamic Algorithms, International Conference and Research Center for Computer Science (IBFI) at Dagstuhl Castle, Wadern, Germany, May 5, 1994.
27. Speeding up slicing. SIGSOFT 94: Second ACM SIGSOFT Symposium on the Foundations of Software Engineering, New Orleans, LA, December 7, 1994.
28. Precise interprocedural dataflow analysis via graph reachability. Twenty-Second ACM Symposium on Principles of Programming Languages, San Francisco, CA, January 23, 1995.
29. Shape analysis as a generalized path problem. PEPM ′95: ACM SIGPLAN Symposium on Partial Evaluation and Semantics-Based Program Manipulation, La Jolla, California, June 21, 1995.
30. Precise interprocedural chopping. Third ACM SIGSOFT Symposium on the Foundations of Software Engineering, Washington, DC, October 11, 1995.
31. Program specialization via program slicing. Seminar on Partial Evaluation, International Conference and Research Center for Computer Science (IBFI) at Dagstuhl Castle, Wadern, Germany, February 15, 1996.

32. Program profiling in debugging and testing. Quest Annual Technology Awareness Symposium, Motorola Museum, Schaumburg, IL, April 29, 1997.
33. The use of program profiling for software maintenance with applications to the Year 2000 Problem. Fourteenth Int. Conf. on Testing Computer Software, Tysons Corner, VA, June 18, 1997.
34. ----------. Sixth European Software Engineering Conference and Fifth ACM SIGSOFT Symposium on the Foundations of Software Engineering, Zurich, Switzerland, Sept. 25, 1997.
35. Program analysis via graph reachability. Seminar on Program Comprehension and Software Reengineering, International Conference and Research Center for Computer Science (IBFI) at Dagstuhl Castle, Wadern, Germany, March 10, 1998.
36. Parametric shape analysis via 3-valued logic. Seminar on Programs with Recursively Defined Data Structures, International Conference and Research Center for Computer Science (IBFI) at Dagstuhl Castle, Wadern, Germany, April 23, 1998.
37. Program analysis via graph reachability. Seminar on Programs with Recursively Defined Data Structures, International Conference and Research Center for Computer Science (IBFI) at Dagstuhl Castle, Wadern, Germany, April 24, 1998.
38. Parametric shape analysis via 3-valued logic. Seminar on Program Analysis, International Conference and Research Center for Computer Science (IBFI) at Dagstuhl Castle, Wadern, Germany, April 13, 1999.
39. Program analysis via graph reachability. Seminar on Model Checking and Program Analysis, Schloss Ringberg, Tegernsee, Germany, Feb. 22, 2000.
40. Semantic minimization of 3-valued propositional formulae. IEEE Symp. on Logic in Computer Science, Copenhagen, Denmark, July 22, 2002.
41. Static program analysis via 3-valued logic. Seminar on Reasoning about Shape, International Conference and Research Center for Computer Science (IBFI) at Dagstuhl Castle, Wadern, Germany, March 3, 2003.
42. Symbolic implementation of the best transformer. Seminar on Reasoning about Shape, International Conference and Research Center for Computer Science (IBFI) at Dagstuhl Castle, Wadern, Germany, March 5, 2003.
43. Weighted pushdown systems and their application to interprocedural dataflow analysis. 10th Int. Static Analysis Symp., San Diego, CA, June 12, 2003.
44. On generalized authorization problems. 16th IEEE Computer Security Foundations Workshop, Pacific Grove, CA, July 1, 2003.
45. Weighted pushdown systems, Seminar on Language-Based Security, International Conference and Research Center for Computer Science (IBFI) at Dagstuhl Castle, Wadern, Germany, Oct. 2003.
46. Numeric analysis of array operations. Thirty-Second ACM Symposium on Principles of Programming Languages, Long Beach, CA, Jan. 2005.
47. CodeSurfer/x86 -- A next-generation platform for analyzing executables. National Security Agency's 5th Conference on High Confidence Software and Systems, Linthicum Heights, MD, March 2005.
48. A next-generation platform for analyzing executables. Software Assurance Forum, Alexandria, VA, April 2005.
49. Model checking x86 executables with CodeSurfer/x86 and WPDS++. Workshop on the Evaluation of Software Defect Detection Tools, June 12, 2005.
50. A next-generation platform for analyzing executables. ARO-DHS Workshop on Malware Detection, Aug. 10, 2005.
51. WYSINWYX: What You See Is Not What You eXecute. IFIP Working Conference on Verified Software: Theories, Tools, Experiments, Zurich, Switzerland, Oct. 12, 2005.

52. ----------. Dagstuhl Seminar on ‘‘Software Verification: Infinite-State Model Checking and Static Program Analysis’’, Feb. 19, 2006.
53. A system for generating static analyzers for machine instructions. Dagstuhl Seminar on ‘‘Types, Logics and Semantics for State’’, Feb. 5, 2008.
53. WYSINWYX: What You See Is Not What You eXecute. Summer School on MOdelling and VErification of parallel Processes (MOVEP), Orleans, France, June 23-28, 2008.
54. There’s plenty of room at the bottom: Analyzing and verifying machine code, Dagstuhl Seminar on Analysis of Executables: Benefits and Challenges, Jan. 30, 2012.
55. Setting the Scene: Decision Procedures and Abstract Interpretation, Dagstuhl Seminar on Decision Procedures and Abstract Interpretation, Aug. 25, 2014.
56. Solving recurrences using operational calculus, Dagstuhl Seminar on Resource Bound Analysis, July 2017.
57. Introduction to program analysis, Tenth Summer School on Formal Techniques, SRI, Menlo Park, CA, May 23-28, 2021.
58. Automating abstract interpretation, Tenth Summer School on Formal Techniques, SRI, Menlo Park, CA, May 23-28, 2021.
59. Algebraic program analysis, Tenth Summer School on Formal Techniques, SRI, Menlo Park, CA, May 23-28, 2021.
60. Newtonian program analysis, Tenth Summer School on Formal Techniques, SRI, Menlo Park, CA, May 23-28, 2021.

Demonstrations of Software at Symposia, Workshops, etc.

1. The Synthesizer Generator. ACM SIGSOFT/SIGPLAN Software Engineering Symposium on Practical Software Development Environments, Pittsburgh, PA, April 24, 1984.
2. The Wisconsin Program-Integration System. Third European Symposium on Programming (ESOP ′90), Copenhagen, Denmark, May 15-18, 1990.
3. ----------. ACM SIGSOFT ′90: Fourth Symposium on Software Development Environments, Irvine, CA, December 4-5, 1990.
4. ----------. Second International Conference on Algebraic Methodology and Software Technology (AMAST), Iowa City, Iowa, May 22-24, 1991.
5. ----------. Third Annual Meeting of the David and Lucile Packard Fellowships Program, Monterey, CA, September 6, 1991.
6. ----------. Seminar on Programming Environments, International Conference and Research Center for Computer Science (IBFI) at Dagstuhl Castle, Wadern, Germany, March 11, 1992.
7. The Synthesizer Generator. Fifteenth International Conference on Software Engineering, Baltimore, MD, May 19, 1993.
8. The Wisconsin Program-Integration System. Fifteenth International Conference on Software Engineering, Baltimore, MD, May 19, 1993.
9. ----------. Seminar on Partial Evaluation, International Conference and Research Center for Computer Science (IBFI) at Dagstuhl Castle, Wadern, Germany, February 15, 1996.

Lectures at Universities and Research Institutes

Approximately 160 seminar talks at universities and research institutes in North America, Europe, Asia, and Australia.

SUPERVISORY ACTIVITIES

Ph.D. Students Supervised as Major Dissertation Advisor

1. Wuu Yang, 1990. (Supervised jointly with S. Horwitz.) *Dissertation:* A new algorithm for semantics-based program integration. Tech. Rep. TR-962, Computer Sciences Department, University of Wisconsin, Madison, WI, August 1990.
2. Phillip Pfeiffer, 1991. *Dissertation:* Dependence-based representations for programs with reference variables. Tech. Rep. TR-1037, Computer Sciences Department, University of Wisconsin, Madison, WI, August 1991.
3. David Binkley, 1991. *Dissertation:* Multi-procedure program integration. Tech. Rep. TR-1038, Computer Sciences Department, University of Wisconsin, Madison, WI, August 1991.
4. G. Ramalingam, 1993. *Dissertation:* Bounded incremental computation. Tech. Rep. TR-1172, Computer Sciences Department, University of Wisconsin, Madison, WI, August 1993.

   Dissertation published as: Ramalingam, G., *Bounded Incremental Computation*. Lecture Notes in Computer Science, Vol. 1089, Springer-Verlag, New York, NY, 1996.
5. Manuvir Das, 1998. *Dissertation:* Partial evaluation using dependence graphs. Tech. Rep. TR-1362, Computer Sciences Department, University of Wisconsin, Madison, WI, February 1998.
6. Michael Siff, 1998. *Dissertation:* Techniques for software renovation. Tech. Rep. TR-1384, Computer Sciences Department, University of Wisconsin, Madison, WI, August 1998.
7. Zhichen Xu, 2000. (Supervised jointly with B. Miller.) *Dissertation:* Safety checking of machine code.
8. David Melski, 2002. *Dissertation:* Interprocedural path profiling and the interprocedural express-lane transformation. TR-1435, Computer Sciences Department, University of Wisconsin, Madison, WI, Feb. 2002.
9. Alexey Loginov, 2006. *Dissertation:* Refinement-based program verification via three-valued-logic analysis. Ph.D. dissertation and Tech. Rep. TR-1574, Computer Sciences Department, University of Wisconsin, Madison, WI, August 2006.
10. Denis Gopan, 2007. *Dissertation:* Numeric program analysis techniques with applications to array analysis and library summarization. Ph.D. dissertation and Tech. Rep. TR-1602, Computer Sciences Department, University of Wisconsin, Madison, WI, August 2007.
11. Gogul Balakrishnan, 2007. *Dissertation:* WYSINWYX: What You See Is Not What You eXecute. Ph.D. dissertation and Tech. Rep. TR-1603, Computer Sciences Department, University of Wisconsin, Madison, WI, August 2007.

    For this work, Balakrishnan received the UW Computer Sciences Department's Outstanding Graduate Student Research Award for 2007–2008.
12. Akash Lal, 2009. *Dissertation:* Interprocedural analysis and the verification of concurrent programs. Ph.D. dissertation, Computer Sciences Department, University of Wisconsin, Madison, WI, August 2009.

    For this work, Lal was a co-recipient of the 2009 SIGPLAN Outstanding Doctoral Dissertation Award. He was also a co-recipient of the UW Computer Sciences Department's Outstanding Graduate Student Research Award for 2008–2009. Lal was also named as one of the 18 awardees selected for the 2011 India TR-35 list (top innovators under 35).
13. Nicholas Kidd, 2009. *Dissertation:* Static verification of data-consistency properties. Ph.D. dissertation and Tech. Rep. TR-1665, Computer Sciences Department, University of Wisconsin, Madison, WI, August 2009.
14. Junghee Lim, 2011. *Dissertation:* Transformer Specification Language: A system for generating analyzers and its applications. Ph.D. dissertation and Tech. Rep. TR-1689, Computer Sciences Depart-

ment, University of Wisconsin, Madison, WI, May 2011.

For this work, Lim received the UW Computer Sciences Department's Outstanding Graduate Student Research Award for 2010–2011.

15. Evan Driscoll, 2013. *Dissertation:* Checking format compatibility of programs using automata. Ph.D. dissertation and Tech. Rep. TR-1799, Computer Sciences Department, University of Wisconsin, Madison, WI, Aug. 2013.
16. Aditya V. Thakur, 2014. *Dissertation:* Symbolic abstraction: Algorithms and applications. Ph.D. dissertation and Tech. Rep. TR-1812, Computer Sciences Department, University of Wisconsin, Madison, WI, Aug. 2014.

    For this work, Thakur was a co-recipient of the UW Computer Sciences Department's Outstanding Graduate Student Research Award for 2013–2014.
17. William Harris, 2014. (Supervised jointly with S. Jha.) *Dissertation:* Secure programming via game-based synthesis. Ph.D. dissertation and Tech. Rep. TR-1814, Computer Sciences Department, University of Wisconsin, Madison, WI, Dec. 2014.
18. Venkatesh Srinivasan, 2017. *Dissertation:* Synthesis of Machine Code: Algorithms and Applications. Ph.D. dissertation and Tech. Rep. TR-1844, Computer Sciences Department, University of Wisconsin, Madison, WI, Mar. 2017.

    For this work, Srinivasan received the UW Computer Sciences Department's Outstanding Graduate Student Research Award for 2016–2017.
19. Tushar Sharma, 2017. *Dissertation:* Abstract Interpretation Over Bitvectors. Ph.D. dissertation and Tech. Rep. TR-1849, Computer Sciences Department, University of Wisconsin, Madison, WI, Aug. 2017.
20. David Bingham Brown, 2020. (Supervised jointly with B. Liblit.) *Dissertation:* Mutation Testing: Algorithms and Applications. Computer Sciences Department, University of Wisconsin, Madison, WI, Aug. 2020.
21. Jason Breck, 2020. *Dissertation:* Enhancing Algebraic Program Analysis. Ph.D. dissertation and Tech. Rep. TR-1863, Computer Sciences Department, University of Wisconsin, Madison, WI, Aug. 2020.
22. Henkel, J., 2022. *Dissertation:* Learning from Code and Non-Code Artifacts, Computer Sciences Department, University of Wisconsin, Madison, WI, July 2022.
23. Cyphert, J., 2023. *Dissertation:* Compositional, Monotone, and Non-linear Program Analysis. Computer Sciences Department, University of Wisconsin, Madison, WI, December 2023.
24. Vaughn, M., 2024. *Dissertation:* Specializing C and x86 Machine-Code Software with OS Assistance. Computer Sciences Department, University of Wisconsin, Madison, WI, January 2024.

Post-Doctoral Associates and Visitors

1. Jan Prins (Cornell Univ.), 1986–87. (Currently Professor, Department of Computer Science, University of North Carolina, Chapel Hill.)
2. Wuu Yang (Univ. of Wisconsin), 1990–91. (Currently Professor, Department of Computer and Information Science, National Chiao-Tung University, Taiwan.)
3. Robert Paige (NYU), 1990–91.
4. Jiazhen Cai (NYU), 1990–91.
5. Mooly Sagiv (Technion), 1994–95. (Currently Professor, Computer Science Department, Tel-Aviv University, Israel.)
6. David Melski, (Ph.D, Univ. of Wisconsin) 2002.

7. Stefan Schwoon (Univ. of Stuttgart), Feb.–Mar. 2003.
8. Bertrand Jeannet (IRISA), Mar.–June 2003.
9. Neil Immerman (U. Mass.), 2004–05.
10. Xin Li (Japan Advanced Institute of Science and Technology), Summer 2010.
11. Anders Moeller (Aarhus University), Oct.–Nov. 2010.
12. Junghee Lim (Univ. of Wisconsin) 2011–2013.
13. Julien Henry (VERIMAG) Nov. 2014–Aug. 2016.
14. Di Wang (Peking University) June–August 2016.
15. Meghana Aparna Sistla (Indian Institute of Technology Madras) May–July 2018.
16. Xusheng Zhi (Peking University), January–July 2024.
17. Zirui Zhou (Peking University), July–August 2024.
18. Chengxi Yang (Shanghai Jiao Tong University), June–December 2025.

External Ph.D. Examiner

1. Emma van der Meulen, Centrum voor Wiskunde en Informatica (The Mathematical Center), Amsterdam, The Netherlands, Jan. 1994.
2. Frank Tip, Centrum voor Wiskunde en Informatica (The Mathematical Center), Amsterdam, The Netherlands, March 1995.
3. Laure Gonnord, University Joseph Fourier -- Grenoble I, Grenoble, France, Oct. 2007.
4. Xiao Xiao, Hong Kong University of Science and Technology (HKUST), Jan. 2016.
5. Ulrik Terp Rasmussen, University of Copenhagen, Denmark, Jan. 2017.
6. Quentin Carbonneaux, Yale University, Aug. 2017.
7. Di Wang, Carnegie Mellon University, May 2022.
8. David Kahn, Carnegie Mellon University, June 2024.
9. Meghana Aparna Sistla, University of Texas at Austin, March 2026.
10. Nikhil Pimpalkhare, Princeton University, June 2026.

GRANTS

PI, *Semantics-Aware Neural Models of Code*, NSF, 2022–26.

Co-PI, *Compositional Semantics-Guided Synthesis*, (Loris D'Antoni, PI), NSF, 2022–25.

Co-PI, *Correct-By-Construction IAM Policies*, (Loris D'Antoni, PI), Amazon, 2021–22.

PI, *Attribution-Based Approaches to Learning from Code and Non-Code Artifacts*, Facebook, 2020–21.

PI, *Programs as Differentiable Data Objects*, Facebook, 2019–20.

PI (for Wisconsin), *Algebraic Program-Analysis Techniques for Invariant Synthesis*, (Zachary Kincaid, Princeton University, PI), ONR, 4/15/2019–4/14/2022.

Co-PI, *Techniques and Tools for De-bloating Containers*, (Somesh Jha, PI), ONR, 2017–2020, plus two-year option.

Co-PI, *Scaling Static Analysis via Large-Scale Distribution*, (Aws Albarghouthi, PI), Google, 2015–16.

PI, *CURB: Calculating and Understanding Resource Bounds to Detect Space/Time Vulnerabilities* (UW Subcontract from GrammaTech, Inc.), DARPA STAC, 2015–2019.

PI, *A Google-Scale SAT Solver*, Google Research Cloud Credits Award, 2014–2015.

Co-PI, *Collaborative Research: Transforming Natural Language to Programming Languages*, Ben Snyder, PI. NSF, 2014–2017.

PI, *PLINY: An End-to-End Framework for Big Code Analytics* (UW Subcontract from Rice Univ.), DARPA MUSE, 2014–2018.

PI, *Recovering Components from Executables*, DARPA BET, 2011–2015.

PI, *Policy Weaving for System Security*, DARPA CRASH, 2010–2015.

PI, *Mathematically Rigorous Methods for Determining Software Quality*, ONR STTR Phase I and Phase II (with GrammaTech, Inc.), 2010–2014,

PI, *MACANTOK – a MAchine-Code-ANalysis TOol Kit – and its Applications*, NSF, 2009–14.

PI, *Automated Vulnerability Detection in Executables*, ARL, 2009–12.

PI, *Mathematically Rigorous Methods for Determining Software Quality*, ONR STTR (with GrammaTech, Inc.), 2010–2011.

PI (for UW subcontract from Purdue University), *A Systematic Defensive Framework for Combating Botnets*, ONR, 2009–10.

PI, *Techniques to Address the Application/Library-Compatibility Problem*, ONR, 2009–12.

PI, *Cross-Boundary Security Analysis*, AFOSR, 2009–12.

PI, *Advanced Methods for Performing Static Analysis of Machine Code*, NSF, 2008–11.

PI, *Advanced Static-Analysis Techniques for Ensuring Reliable Software*, NSF, 2006–2009.

PI, *Reverse Engineering Kernel-Mode Rootkits*, GrammaTech, Inc., 2006–2007.

PI, *Deobfuscating Tools for Validation and Verification of Tamper-Proofed Software*, GrammaTech, Inc., 2006–2007.

PI (for UW subcontract from GrammaTech, Inc.), *Deep Static Analysis of Binaries*, IARPA, 2006–2008.

PI (for UW subcontract from GrammaTech, Inc.), *Model Checking Software Binaries*, HSARPA, 2005–2007.

PI, *Advanced Methods for Checking Information-Security Properties*, NSF Cybertrust, 2005–2009.

PI (for UW subcontract from CMU), *Static Analysis to Enhance the Power of Model Checking for Concurrent Software*, ONR University Research Initiative, 2001–2006.

Co-PI, *Vulnerability and Information Flow Analysis for COTS*, ONR University Research Initiative, 2001–2006.

PI, *Investigation of a New Compressed Representation of Boolean Functions*, NSF, 2000–04.

PI, *Investigation of a New Compressed Representation of Boolean Functions*, ONR, 2000–01.

Co-PI, *Run-Time Type Checking of C Programs*, IBM Partnership Award, IBM Research, 2000–01.

Co-PI, *Run-Time Type Checking of C Programs*, IBM Partnership Award, IBM Research, 1999–00.

PI, *Software Diagnostics Using Path-Spectrum Information*, IBM Partnership Award, IBM Research, 1998–99.

PI, *Shape-Analysis for Languages with Destructive Updating*, NSF, 1997–00.

Co-PI, *Slicing Programs with Recursive Data Structures*, United States-Israel Binational Science Foundation, 1997–00.

PI, *Improved Techniques for Program Analysis and Program Specialization*, Rockwell Foundation.

PI, *Software Diagnostics Using Path-Spectrum Information*, IBM Partnership Award, IBM Research, 1997–98.

PI, *A Preprocessor for Activity Analysis of ANSI-C Programs*, Argonne National Laboratories, 1997.

PI, *Semantics-Based Program Manipulation*, NSF, 1996–98.

PI, *Technical Opportunities to Help with the Year 2000 Problem*, Advanced Research Projects Agency, 1996–97.

PI, *Improved Techniques for Program Analysis and Program Slicing*, Advanced Research Projects Agency, 1995–98.

PI, *Software Support for Programming in the Large*, Advanced Research Projects Agency, 1992–97.

PI, *Travel Support for U.S. Participants at an International Workshop*, National Science Foundation, 1992–93.

PI, *Semantics-Based Program Integration*, National Science Foundation, 1991–96.

David and Lucile Packard Fellowship for Science and Engineering, David and Lucile Packard Foundation, 1988–93.

PI, *Software Support for Programming in the Large*, Defense Advanced Research Projects Agency, 1988–91.

PI, *Software Support for Programming in the Large*, Digital Equipment Corporation, 1988–91.

Presidential Young Investigator Award, NSF, 1986–91.

PI, *Language-Based Program Development Environments*, Xerox Corporation, 1986–89.

Faculty Development Award, IBM Corporation, 1986–88.

PI, *Language-Based Program Development Environments*, Digital Equipment Corporation, 1986–88.,

PI, *Language-Based Program Development Environments*, Siemens Corporate Research, 1986–87.

Participating Faculty Member, *PRISM: A Laboratory for Research in Future High-Performance Parallel Computing*, NSF, 1991–96. (M.K. Vernon, M.J. Carey, C.R. Dyer, M.D. Hill, R.R. Meyer, and B.P. Miller, PIs.)

Participating Faculty Member, *TOPAZ: A Laboratory for Research in Distributed Computing*, NSF, 1986–91. (D. Dewitt, R. Finkel, C. Fischer, L. Landweber, O. Mangasarian, and M. Solomon, PIs.)

PROFESSIONAL ACTIVITIES

Government

Member, F/A-22 Avionics Advisory Team, Office of the Deputy Under Secretary for Defense (Science and Technology), 2003.

Consultant to the Defense Advanced Research Projects Agency (DARPA) to plan a project aimed at reducing the impact of the Year 2000 Problem on the Department of Defense, 1996–97.

Editorial Activities

Co-editor, *Program Analysis and Compilation, Theory and Practice: Essays Dedicated to Reinhard Wilhelm*. (Festschrift for Prof. Reinhard Wilhelm, Univ. des Saarlandes), Springer-Verlag.

Member, Advisory Editorial Board, *Software–Practice & Experience*, 2003–present.

Associate Editor, *IEEE Trans. on Softw. Eng. (TSE)*, 2001–2003.

Guest editor, *ACM Trans. on Program. Lang. and Syst. (TOPLAS) 8*, 4 (October 1986). Special issue consisting of papers from the ACM SIGPLAN 85 Symposium on Languages Issues in Programming Environments (Seattle, WA, June 1985).

Steering Committees

Member of Steering Committee, International Conference on Verification, Model Checking, and Abstract Interpretation (VMCAI), 2002–present.

Organizer/Co-organizer

Seminar on “Decision Procedures and Abstract Interpretation”, International Conference and Research Center for Computer Science (IBFI) at Dagstuhl Castle, Wadern, Germany, Aug. 25–29, 2014. (With D. Kroening, S. Seshia, and A. Thakur.)

Seminar on “Analysis of Executables: Benefits and Challenges”, International Conference and Research Center for Computer Science (IBFI) at Dagstuhl Castle, Wadern, Germany, Jan. 30 – Feb. 3, 2012. (With A. Flexeder, A. King, A. Mycroft, and A. Simon.)

Seminar on “Program Comprehension and Software Reengineering”, International Conference and Research Center for Computer Science (IBFI) at Dagstuhl Castle, Wadern, Germany, March 9-13, 1998. (With G. Snelting and H. Mueller.)

NSF/DARPA/ARO Workshop on Software Engineering and Programming Languages, Cambridge, MA, June 1996. (With H. Gill, C. Gunter, I. Lee, P. Lee, J. Mitchell, D. Notkin, and V. Tannen.)

Seminar on “Incremental Computation and Dynamic Algorithms”, International Conference and Research Center for Computer Science (IBFI) at Dagstuhl Castle, Wadern, Germany, May 2-6, 1994. (With J. van Leeuwen and K. Mehlhorn.)

Seminar on “Programming Environments”, International Conference and Research Center for Computer Science (IBFI) at Dagstuhl Castle, Wadern, Germany, March 9-13, 1992. (With P. Klint and G. Snelting.)

Chairman of Program Committee

Twenty-Seventh ACM Symposium on Principles of Programming Languages, January 2000.

Member of Program Committee

ACM SIGPLAN Symposium on Principles of Programming Languages (POPL), 2024

ACM SIGPLAN Conf. on Systems, Programming, Languages and Applications: Software for Humanity (SPLASH/OOPSLA), 2021 (External Review Committee).

Static Analysis Symposium (SAS), 2021.

ACM SIGPLAN ′21: Conference on Programming Language Design and Implementation (PLDI), 2021.

Workshop on Forming an Ecosystem Around Software Transformation (FEAST), 2018.

Workshop on Partial Evaluation and Program Manipulation (PEPM), 2018.

7th Workshop on Numerical and Symbolic Abstract Domains (NSAD), 2017.

Static Analysis Symposium (SAS), 2011.

17th ACM Conference on Computer and Communications Security (CCS), 2010.

Static Analysis Symposium (SAS), 2009.

18th Conference on Computer-Aided Verification (CAV), 2006.

7th International Workshop on Verification, Model Checking, and Abstract Interpretation (VMCAI), 2006.

European Symposium on Programming (ESOP), 2005.

ACM SIGPLAN ′04: Conference on Programming Language Design and Implementation (PLDI), 2004.

ACM Workshop on Partial Evaluation and Semantics-Based Program Manipulation (PEPM), 2003.

3rd International Workshop on Verification, Model Checking, and Abstract Interpretation (VMCAI), 2002.

Eleventh International Conference on Compiler Construction, 2001.

First International Workshop on Automated Program Analysis, Testing and Verification, 2000.

Static Analysis Symposium, 1998.

IEEE International Conference on Computer Languages, 1998.

ACM PEPM ′97: SIGPLAN Symposium on Partial Evaluation and Semantics-Based Program Manipulation, 1997.

ACM SIGPLAN ′96: Conference on Programming Language Design and Implementation, 1996.

Sixth International Conference on Compiler Construction, 1996.

First ACM SIGSOFT Symposium on Foundations of Software Engineering, 1993.

ACM SIGSOFT ′90: Fourth Symposium on Software Development Environments, Irvine, CA, 1990.

Sixteenth ACM Symposium on Principles of Programming Languages, Austin, TX, 1989.

Second European Symposium on Programming (ESOP ′88), Nancy, France, 1988.

Fourteenth ACM Symposium on Principles of Programming Languages, Munich, West Germany, 1987.

ACM SIGPLAN 85 Symposium on Languages Issues in Programming Environments, Seattle, WA, 1985.

Member of Selection Committee

Ninth ACM George E. Forsythe Student Paper Competition, 1982.

Student Research Competition Selection Committee for POPL, Jan. 2021.

ACM SIGPLAN John C. Reynolds Doctoral Dissertation Award, 2022–24.

ACM-IEEE CS Ken Kennedy Award, 2026–28.

Review Committees

Member, Review Committee for the U.S. Army Research Office Computing and Information Sciences Division, 2006.

Search Committees

Chair, Search Committee for Editor-in-Chief of ACM Trans. on Program. Lang. and Syst. (TOPLAS), 2016.

Proposal-Review Panels

Member, National Science Foundation proposal-review panels in 1987, 1989, 2007, and 2018.

Instructor

International course on syntax-directed editing, Aussois, France, April 18-22, 1983.

Summer School on MOdelling and VErification of parallel Processes (MOVEP), Orleans, France, June 23-28, 2008.

Tenth Summer School on Formal Techniques, SRI, Menlo Park, CA, May 23-28, 2021.

Panelist

“Panel on Software Development, Quality, and Metrics”, CIP/SW URI Workshop, Annapolis, MD, Aug. 18, 2004.

“Formal methods and software verification”, CAV 2000: 12th Int. Conf. on Computer Aided Verification, July 18, 2000.

“Programming environments and tools”, IFIP 12th World Computer Congress. Madrid, Spain, September 1992.

“Programming environments and tools”, IBM Academic Information Systems 1986 University Study Conference, Ft. Lauderdale, FL, November 1986.

Participant

NSF Workshop on Symbolic Computation for Constraint Satisfaction Problems, Washington, DC, November 2008.

DARPA/ITO Workshop on Software-Enabled Control of Systems, Atlanta, GA, December 1997.

"Models of Programming Languages" Working Group, ACM Workshop on Strategic Directions in Computing Research, Cambridge, MA, June 1996.

Workshop on Foundational Studies for Software Engineering, NSF and ARPA, Palo Alto, CA, September 1995.

Workshop on the Future of Research in Programming Languages and Compilers, NSF, Charleston, SC, January 1993.

Workshop on Critical Research Directions in Programming Languages, Office of Naval Research, Miami Beach, FL, October 1988.

Referee/Reviewer

ACM Computing Surveys

ACM Trans. on Program. Lang. and Syst. (TOPLAS)

ACM Trans. on Softw. Eng. and Methodology (TOSEM)

Acta Informatica

IEEE Transactions on Computers

IEEE Trans. on Softw. Eng. (TSE)

Information Processing Letters

Lecture Notes in Computer Science

Mathematical Systems Theory

Science of Computer Programming

Software—Practice & Experience

The M.I.T. Press

National Research Council of the National Academy of Sciences

National Science Foundation

Netherlands Computer Science Research Foundation

Swedish Research Council for Engineering Sciences

Professional Societies

Association for Computing Machinery (ACM)