

Curriculum Vita

Paul R. Hudak
Professor

Department of Computer Science
Yale University
P.O. Box 208285
New Haven, CT 06520-8285
(203) 432-1235 paul.hudak@yale.edu

Born: July 15, 1952
U.S. Citizen, Married, two children

Education

- PhD in Computer Science, University of Utah, 1982. Dissertation title: *Object and Task Reclamation in Distributed Applicative Processing Systems.*
- MS in Computer Science, Massachusetts Institute of Technology, 1974. Thesis title: *A Model of Music Comprehension.*
- BS in Electrical Engineering, Vanderbilt University, 1973, *magna cum laude.*

Professional Appointments

- Master, Saybrook College, Yale University, New Haven, CT, July 2009 to present (on medical leave Nov 1, 2010 to July 1, 2011).
- Professor, Department of Computer Science, Yale University, New Haven, CT, July 1992 to present.
- Adjunct Professor, University of Utah, School of Computing, Salt Lake City, UT, September 2008 to September 2011.
- Professor and Chair, Department of Computer Science, Yale University, New Haven, CT, July 1999 to July 2005.
- Associate Professor with tenure, Department of Computer Science, Yale University, 1988-1992.
- Associate Professor, Department of Computer Science, Yale University, 1986-88.
- Assistant Professor, Department of Computer Science, Yale University, 1982-86.
- Collaborator, IBM T.J. Watson Research Laboratory, Hawthorn, NY, 1989-90.
- Collaborator, Los Alamos National Laboratory, 1983-1993.
- Teaching Assistant 1979-1982, University Research Fellow 1981-1982, University of Utah, Department of Computer Science, Salt Lake City, UT.

- Engineer and Manager, Intelligence Acquisition Systems, Watkins-Johnson Company, Gaithersburg, MD, 1974-1979. Held top-secret DOD clearance.
- Teaching Assistant, Digital Systems Lab, Massachusetts Institute of Technology, Cambridge, MA, 1973-74.

Honors

- Most Influential ICFP Paper Award, 2007, for the paper “Functional Reactive Animation” (written with Conal Elliott), published in ICFP ’97.
- ACM Fellow, 2004.
- Co-author of one of fifty papers chosen for inclusion in “Twenty Years of PLDI (1979-1999): A Selection” in 2002. (“ORBIT: An Optimizing Compiler for Scheme,” by Kranz, Kelsey, Rees, Hudak, Philbin, and Adams.)
- Research on *functional reactive programming* chosen as one of seven “Great Talks” by WG2.8 for nomination to IFIP TC-2 in 2002.
- Presidential Young Investigator Award, 1985.
- IBM Faculty Development Award, 1984.
- University of Utah Research Fellow, 1981-82.
- Eta Kappa Nu; Sigma Xi.

University Service

- Member, Faculty Committee on Athletics, 2005-present.
- Member, Dean’s Advisory Committee on the Arts, 2009-2010.
- Chair, Quantitative Reasoning Council, 2004-2009.
- Member, Committee on Technology and Music, 2007-2008.
- Member, Committee on Cooperative Research, 2006-2008.
- Member, Digital Landscape Committee, 2005-2007.
- Member, Engineering and Physical Sciences Faculty Fellowship Selection Committee, 2005-2007.
- Chair, Department of Computer Science, 1999-2005.
- Member, Dean’s Committee, Faculty of Engineering, 1999-2005.
- Chair, Information Technology Committee, 1995-1997.
- Member, Instructional Facilities Oversight Committee, 1997.
- Member, Committee on Math Instruction, 1994-1995.
- Member, Executive Committee of the Graduate School, 1994-1995.

Professional Activities

- General Chair, 2010 ACM International Conference on Functional Programming (ICFP).
- Member, Program Committee, 2009 International Symposium on Implementation and Application of Functional Languages.
- co-Editor-in-Chief, *Journal of Functional Programming* (2004-2008).
- Editor, *Journal of Functional Programming* (1989-2003, 2008-2010).
- Charter Member, Editorial Board, *Journal of Higher-Order and Symbolic Computation* (1988-present). (Previously *Lisp and Symbolic Computation: An Int'l Journal*.)
- Charter Member, IFIP WG2.8, Working Group on Functional Programming (1988-present).
- Member, Microsoft Research University Relations Faculty Advisory Board, 2002-03.
- Associate Editor, *ACM Transactions on Programming Languages and Systems* (1989-1992).
- Charter Member, Editorial Board, *International Journal of Parallel Programming* (1987-1992).
- co-Chairman, Program Committee for the 2008 International Symposium on Practical Aspects of Declarative Languages (PADL).
- Co-Chairman, Program Committee of 1998 ACM International Conference on Functional Programming (ICFP).
- Chairman, Program Committee of 1995 Haskell Workshop.
- Chairman, 1993 ACM Workshop on State in Programming Languages (SIPL).
- Co-Chairman, Program Committee for the 1991 ACM Symposium on Partial Evaluation and Semantics-Based Program Manipulation (PEPM).
- Chairman, Program Committee for the 1990 ACM Symposium on Principles of Programming Languages (POPL).
- Member (and Editor), *Haskell Committee*, an international functional programming language standardization effort, 1988-1991.
- Member, Program Committees for the 1988, 1989, and 2007 ACM Symposium on Principles of Programming Languages (POPL).
- Member, Program Committee for the 2001 ACM SIGPLAN Conference on Programming Language Design and Implementation (PLDI).
- Member, Program Committee for the 2000, 2005, and 2006 Haskell Workshops.
- Member, Program Committee for the 2000 and 2006 International Symposium on Practical Aspects of Declarative Languages (PADL).

- Member, Program Committee for the 2000 Journées d'Informatique Musicale (JIM), a computer music conference.
- Member, Program Committee for the 1999 and 1997 Usenix Conference on Domain Specific Languages (DSL).
- Member, Program Committee for the 1997 and 1995 International Conference on Declarative Programming Languages in Education (DPLE).
- Member, Program Committee for the 1996 International Conference on Functional Programming (ICFP).
- Member, Program Committee for the 1995 ACM Symposium on Partial Evaluation and Semantics-Based Program Manipulation (PEPM).
- Member, Program Committee for the 1988 and 2001 ACM SIGPLAN Conference on Programming Language Design and Implementation (PLDI).
- Member, Program Committees for the 1987, 1989, and 1991 Conferences on Functional Programming Languages and Computer Architecture (FPCA).
- Member, DARPA Working Group on a Common Prototyping Language, (1987-88).
- Member, "Formal Methods Team" in NSF/DARPA-ESPRIT Negotiations for Collaboration, Brussels (July 1990).
- Member, Association for Computing Machinery and ACM SIGPLAN.
- Member, Institute of Electrical and Electronic Engineers and IEEE Computer Society.

Seminars and Colloquia

- "Euterpea: From Signals to Symphonies," invited talk given at Rutgers University (April 2010), Vanderbilt University (April 2010), University of Utah (March 2010), and Williams College (May 2010).
- "An Ode to Arrows," Practical Aspects of Declarative Languages, Madrid, Spain, January 2010.
- "Haskell and the Arts: How Functional Programmers can Help, Inspire, or even Be Artists," QCon, San Francisco, November 2008.
- "Sound Synthesis in Haskell," New York Functional Programmer's Meet-Up, February 2008.
- "Arrows and Space Leaks," Workshop on Foundational Methods in Computer Science (FMCS), Colgate University, June, 2007.
- "Plugging a Space Leak with an Arrow," IFIP WG2.8 Working Group on Functional Programming, Nesjavellir, Iceland, July 2007.
- "HPorter: Using Arrows to Compose Parallel Processes," Practical Aspects of Declarative Languages, Nice, France, January 2007.

- “Why Programming Languages Matter,” invited talk given at Haverford University (February 2005), Central CT State University (February 2005), and the University of Bordeaux (January 2007).
- “HasSound: Generating Musical Instrument Sounds in Haskell,” New England Programming Language Seminar, October 2005.
- “An Algebraic Theory of Polymorphic Temporal Media,” Keynote Address, IBM Programming Languages Day, April 2003 (invited talk); also presented at the PADL Symposium, June 2004; Harvard University, February 2004 (invited talk); and Cornell University, February 2004 (invited talk).
- “A Sound and Complete Algebraic Semantics for Music,” Drexel University MCS Society, Distinguished Lecture, February 2003 (invited talk); also presented at the IFIP WG2.8 Working Group on Functional Programming, Crans-Montana, Switzerland, January 2003.
- “Describing and Interpreting Music in Haskell,” University of Vermont, November 2002 (invited talk).
- “Arrows, Robots, and Functional Reactive Programming,” Lecture Series, Summer School on Advanced Functional Programming, Oxford, England, August 2002.
- “Functional Reactive Programming in Real-Time Applications,” University of Pennsylvania’s Systems Seminar Series, April 2002 (invited talk).
- “Little Languages for Big Applications,” Haverford College Computer Science Department, April 2002 (invited talk).
- “A Brief History of Time (in the Design of FRP),” IFIP WG2.8 Working Group on Functional Programming, Las Vegas, March 2002.
- “APL, FP, and Yale,” Keynote Address, APL 2001 Conference, Yale University, June 2001.
- “Real-Time FRP,” IFIP WG2.8 Working Group on Functional Programming, Are, Sweden, April 2001, and New England Programming Language Seminar, Williams College, Department of Computer Science, May 2001 (invited talk).
- “Issues in Domain Specific Language Design and Implementation,” Invited Lecture, 2000 ACM Symposium on Principles of Programming Languages, January 2000.
- “Functional Multimedia,” Invited Lecture, IEEE Computer Society International Conference on Computer Languages, May 1998.
- “The Promise of Domain Specific Languages,” Keynote Address, USENIX Conference on Domain Specific Languages, October 1997.
- “A Timely Semantics for Fran,” Lucent Technologies, May 1997 (invited talk).
- “Domain-Specific Embedded-Language Design,” Joint (NSF and CNPq) Workshop on Formal Foundations of Software Systems, May 1997 (invited talk).

- “An Algebraic Approach to Multimedia Programming,” Distinguished Lecture Series, University of Washington, October 1996; and Harvey Mudd College, Dept. of Computer Science, May 1997 (invited talks).
- “Haskore Music Tutorial,” Lecture Series, Summer School on Advanced Functional Programming, Olympia, WA, August 1996.
- “What Do Music, Graphics, and Animation Have in Common: A Formal Methods Approach to Multimedia,” IBM T.J. Watson Research, Yorktown Heights NY; Wesleyan University, Middletown CT, Fall 1994; Microsoft Research, June 1995; ACM Hartford Chapter Colloquium, April 1996 (invited talks).
- “A Model of Performance, Interaction, and Improvisation,” International Conference on Computer Music, Banff Canada, September 1995.
- “Linear Monadic MADT’s,” SCILL Workshop, Univ. of Glasgow, July 1995.
- “Using Types to Parse Natural Language,” Glasgow Functional Programming Workshop, July 1995.
- “Reflections on Program Optimization,” Workshop on Static Analysis, Padova Italy, September 1993 (invited talk).
- “How to Add *State* to a Functional Language,” tutorial presented at the ACM SIPL Workshop, Copenhagen, June 1993.
- Wheaton College, Norton MA, “Resolving the Discrepancy Between Programming Languages and Formal Specifications,” NSF-sponsored Summer Course, June 1992 (invited talk).
- University of Pennsylvania, Philadelphia PA, “How to Have Your State and Munge It Too,” January 1992 (invited talk).
- ACM Conference on Functional Programming Languages and Computer Architecture, Cambridge MA, August 1991, “Manipulating State in Functional Languages” (invited tutorial).
- ACM Symposium on Programming Language Design and Implementation, Toronto, June 1991, “The State of Functional Programming” (invited tutorial).
- Johns Hopkins University’s IBM Distinguished Lecture Series, February 1991, “Haskell: A Common Functional Language” (invited talk).
- ACM Symposium on Principles of Programming Languages, Orlando FL, January 1991, “Incremental Computation via Partial Evaluation.”
- Brandeis Universty (November 1990) and University of Utah (January 1991), “Innovative Design Features of Haskell” (invited talks).
- IEEE Computer Languages Conference, March 1990, “Tutorial on the Functional Programming Language Haskell” (invited tutorial).

- IBM T.J. Watson Research Center (December 1989), New York University (December 1989), Princeton University (February 1990), Pennsylvania State University (February 1990), and Glasgow University (June 1990), “Single-Threaded Polymorphic Lambda Calculus” (all invited talks).
- Williams College, NECUSE Workshop, September 1988, “The Functional Programming Paradigm” (invited talk).
- IBM T.J. Watson Research Center (June 1988), and Brown University, Department of Computer Science (July 1988), “On the Design of Haskell” (invited talks).
- MIT and Indiana University, January 1988, “Graphinators and the Duality of SIMD and MIMD” (invited talks).
- ACM Conference on Functional Programming Languages and Computer Architecture, Portland OR, September 1987, “Implementing Functional Languages” (invited tutorial).
- ACM Conference on Functional Programming Languages and Computer Architecture, Portland OR, September 1987, “Pomset Interpretations of Parallel Functional Programs.”
- IBM T.J. Watson Research Center (May 1987) and Virginia Polytechnic Institute, Department of Computer Science (October 1987), “Arrays, Non-Determinism, Side-Effects, and Parallelism: A (Pseudo-)Functional Perspective” (invited talks).
- IEEE COMPCON '87, February 1987, “Para-functional Languages for Parallel and Distributed Computing” (invited talk).
- Cornell University, University of Chicago, and University of Arizona, November 1986, “Para-functional Programming: A Status Report” (invited talks).
- ACM Symposium on Lisp and Functional Programming, Cambridge, August 1986, “A Semantic Model of Reference Counting and its Abstraction.” Extended version presented at AT&T Bell Labs, Murray Hill, July 1986 (invited talk).
- ACM Symposium on Principles of Programming Languages, St. Petersburg, FL, January 1986, “Higher-Order Strictness Analysis in Untyped Lambda Calculus.” Extended version presented at Mass. Institute of Technology, February 1986 (invited talk).
- ACM Symposium on Principles of Programming Languages, St. Petersburg, January 1986, “Para-functional Programming: A Paradigm for Programming Multiprocessor Systems.” Extended versions presented at the University of Rochester (November 1985), University of Utah (March 1986), and Schlumberger/Doll Research (April 1986).
- Workshop on Programs as Data Objects, University of Copenhagen, Denmark, October 1985, “Inferencing Strategies for Copy Avoidance in Applicative Languages” (invited talk).

- IFIP International Conference on Functional Programming Languages and Computer Architecture, Nancy France, September 1985, “Serial Combinators: Optimal Grains of Parallelism.” Extended version presented at MCC (Microelectronics and Computer Technology Corporation), Austin, December 1985 (invited talk).
- Workshop on Abstract Interpretation, University of Kent, Canterbury England, August 1985, “Higher-Order Strictness Analysis,” invited talk. Also presented at University of Utah, August 1985 (invited talk).
- Workshop on Implementation Issues in Functional Programming Systems, Chalmers University, Aspenas Sweden, February 1985, “A Set-Theoretic Characterization of Function Strictness in the Lambda Calculus” (invited talk).
- ACM Symposium on Principles of Programming Languages, New Orleans, January 1985, “The Aggregate Update Problem in Functional Programming Systems.”
- ACM Symposium on Lisp and Functional Programming, Austin, August 1984, “Experiments in Diffused Combinator Reduction.”
- Los Alamos National Laboratories, May 1984, “Distributed Combinator Reduction” (invited talk).
- New York University, Department of Computer Science, April 1984, “An Overview of Functional Programming” (invited talk).
- ACM Symposium on Principles of Programming Languages, Salt Lake City, Utah, January 1984, “A Combinator-based Compiler for a Functional Language.”
- ACM Symposium on Principles of Distributed Computing, Montreal, August 1983, “Distributed Task and Memory Management.”
- DEC External Research Workshop, San Francisco, June 1983, “Current Systems Research at Yale” (invited talk).
- University of Utah, Department of Computer Science, March 1983, “The Myth of the Simple Dataflow Compiler” (invited talk).
- ACM Symposium on Lisp and Functional Programming, Pittsburg, August 1982, “Garbage Collection and Task Deletion in DAPS.”
- Presented dissertation research at Univ. of Mass., Univ. of Maryland, Penn. State Univ., Univ. of Penn., Princeton Univ., Yale Univ., Univ. of Delaware, SUNY at Stony Brook, Univ. of Pittsburgh and Univ. of Vermont, April 1982.

Publications in Journals and Books

- [1] Paul Liu, Eric Cheng, and Paul Hudak. Causal commutative arrows. *Journal of Functional Programming*, 21(Special Issue 4-5):467–495, September 2011.
- [2] Paul Hudak. *The Haskell School of Music – From Signals to Symphonies*. draft Version [2.0], January 2011.

- [3] Henrik Nilsson, John Peterson, and Paul Hudak. Functional hybrid modeling from an object-oriented perspective. *Simulation News Europe*, 17(2):29–38, September 2007.
- [4] Paul Hudak. A sound and complete axiomatization of polymorphic temporal media. *Journal of Logic and Algebraic Programming*, (submitted), 2008.
- [5] Paul Liu and Paul Hudak. Plugging a space leak with an arrow. *Electronic Notes in Theoretical Computer Science*, 193:29–45, November 2007.
- [6] Rajiv Mirani and Paul Hudak. First-class monadic schedules. *ACM Transactions on Programming Languages and Systems*, 26(4):609–651, July 2004.
- [7] Paul Hudak. Describing and interpreting music in Haskell. In *The Fun of Programming*, chapter 4. Palgrave, 2003.
- [8] Paul Hudak. *The Haskell School of Expression – Learning Functional Programming through Multimedia*. Cambridge University Press, New York, 2000.
- [9] Paul Hudak. Functional programming. In *Encyclopedia of Computer Science and Technology, Vol. 38*, chapter 7, pages 149–158. Marcel Dekker, New York, 1998.
- [10] Paul Hudak. Domain specific languages. In *Handbook of Programming Languages, Vol. III: Little Languages and Tools*, chapter 3, pages 39–60. MacMillan, Indianapolis, 1998.
- [11] P. Hudak. Building domain specific embedded languages. *ACM Computing Surveys*, 28A:(electronic), December 1996.
- [12] Paul Hudak, Tom Makucevich, Syam Gadde, and Bo Whong. Haskore music notation – an algebra of music. *Journal of Functional Programming*, 6(3):465–483, May 1996.
- [13] A. Kishon and P. Hudak. Semantics-directed program execution monitoring. *Journal of Functional Programming*, 5(4), October 1995.
- [14] P. Hudak, S. Peyton Jones, and P. Wadler (editors). Report on the Programming Language Haskell, A Non-strict Purely Functional Language (Version 1.2). *ACM SIGPLAN Notices*, 27(5), May 1992.
- [15] P. Hudak and J. Fasel. A gentle introduction to Haskell. *ACM SIGPLAN Notices*, 27(5), May 1992.
- [16] P. Hudak, S. Peyton Jones, and P. Wadler (editors). Report on the Programming Language Haskell, A Non-strict Purely Functional Language (Version 1.2). *ACM SIGPLAN Notices*, 27(5), May 1992.
- [17] P. Hudak and S. Anderson. Haskell solutions to the language session problems at the 1988 Salishan High-Speed Computing Conference. In J.T. Feo, editor, *A Comparative Study of Parallel Programming Languages: The Salishan Problems*. North-Holland, Amsterdam, 1992.

- [18] P. Hudak. Para-functional programming in Haskell. In *Parallel Functional Languages and Compilers*, pages 159–196. ACM Press (New York) and Addison-Wesley (Reading), 1991.
- [19] P. Hudak and J. Young. Collecting interpretations of expressions. *ACM Transactions on Programming Languages and Systems*, 13(2):269–290, 1991. Preliminary version appeared in the Proceedings of the 1988 ACM Symposium on Principles of Programming Languages.
- [20] P. Hudak. Conception, evolution, and application of functional programming languages. *ACM Computing Surveys*, 21(3):359–411, 1989.
- [21] K. Li and P. Hudak. Memory coherence in shared virtual memory systems. *ACM Transactions on Computer Systems*, 7(4):321–359, November 1989.
- [22] P. Hudak. Exploring para-functional programming: Separating the *what* from the *how*. *IEEE Software*, 5(1):54–61, January 1988.
- [23] Z.G. Mou and P. Hudak. An algebraic model for divide-and-conquer and its parallelism. *Journal of Supercomputing*, 2(3), 1988.
- [24] A. Bloss, P. Hudak, and J. Young. An optimising compiler for a modern functional language. *The Computer Journal*, 31(6):152–161, 1988.
- [25] A. Bloss, P. Hudak, and J. Young. Code optimizations for lazy evaluation. *Lisp and Symbolic Computation: An International Journal*, 1(2):147–164, September 1988.
- [26] P. Hudak. Denotational semantics of a para-functional programming language. *International Journal of Parallel Programming*, 15(2):103–125, April 1986.
- [27] P. Hudak. Para-functional programming. *IEEE Computer*, 19(8):60–71, August 1986.
- [28] P. Hudak. A semantic model of reference counting and its abstraction. In *Abstract Interpretation of Declarative Languages*, pages 45–62. Ellis Horwood, 1987. (Preliminary version appeared in Proceedings 1986 ACM Conference on LISP and Functional Programming, August 1986, pp. 351-363).
- [29] K. Li and P. Hudak. A new list compaction method. *Software – Practice and Experience*, 16(2):145–163, February 1986.
- [30] P. Hudak and B. Goldberg. Distributed execution of functional programs using serial combinators. *IEEE Transactions on Computers*, C-34(10):881–891, October 1985. Also appeared in Proceedings of 1985 Int’l Conference on Parallel Processing, Aug. 1985, pp. 831-839.
- [31] P. Hudak. *Object and Task Reclamation in Distributed Applicative Processing Systems*. PhD thesis, University of Utah, July 1982.

- [32] J.R. Bourne, P. Hudak, and J.L. Duke. Computer automated electro-oculography. *Computers and Biomedical Research*, 5:654–658, June 1972.

Publications in Refereed Conferences

- [1] Daniel Winograd-Cort, Hai Liu, and Paul Hudak. Virtualizing real-world objects in frp. In *Proceedings of PADL'12: 14th International Symposium on Practical Aspects of Declarative Languages*, page to appear, January 2012.
- [2] Zhijing G. Mou, Hai Liu, and Paul Hudak. Compress-and-conquer for optimal multicore computing. In *Proceedings of the 5th ACM SIGPLAN workshop on Declarative aspects of multicore programming*, DAMP '10, pages 35–44, New York, NY, USA, 2010. ACM.
- [3] Andreas Voellmy and Paul Hudak. Nettle: Taking the sting out of programming network routers. In Ricardo Rocha and John Launchbury, editors, *PADL*, volume 6539 of *Lecture Notes in Computer Science*, pages 235–249. Springer, 2011.
- [4] Paul Liu, Eric Cheng, and Paul Hudak. Causal commutative arrows and their optimization. In *Proc. International Conference on Functional Programming*. ACM Sigplan, 2009.
- [5] Andreas Voellmy and Paul Hudak. Nettle: A language for configuring routing networks. In *DSL '09: Proceedings of the IFIP TC 2 Working Conference on Domain-Specific Languages*, pages 211–235, Berlin, Heidelberg, 2009. Springer-Verlag.
- [6] Hai Liu and Paul Hudak. An ode to arrows. In *Proc. Practical Aspects of Declarative Languages*. Springer Verlag LNCS 5937, January 2010.
- [7] Henrik Nilsson, John Peterson, and Paul Hudak. Functional hybrid modeling from an object-oriented perspective. In Peter Fritzson, François Cellier, and Christoph Nytsch-Geusen, editors, *Proceedings of the 1st International Workshop on Equation-Based Object-Oriented Languages and Tools*, number 24 in Linköping Electronic Conference Proceedings, pages 71–87. Linköping University Electronic Press, 2007.
- [8] P. Hudak, J. Hughes, S. Peyton Jones, and P. Wadler. A history of Haskell: being lazy with class. In *Proc. ACM SIGPLAN History of Programming Languages Conference*, New York, June 2007. ACM Press.
- [9] L. Huang, P. Hudak, and J. Peterson. Hporter: Using arrows to compose parallel processes. In *Proc. Practical Aspects of Declarative Languages*, pages 275–289. Springer Verlag LNCS 4354, January 2007.
- [10] Paul Hudak. Polymorphic temporal media. In *Proceedings of PADL'04: 6th International Workshop on Practical Aspects of Declarative Languages*. Springer Verlag LNCS, June 2004.
- [11] David Kranz, Richard Kesley, Jonathan Rees, Paul Hudak, James Philbin, and Norman Adams. Retrospective on: Orbit: an optimizing compiler for Scheme. *ACM SIGPLAN Notices, 20 Years of the ACM SIGPLAN Conference on Programming Language Design and Implementation (1979–1999): A Selection*, 39(4), April 2004.

- [12] Henrik Nilsson, John Peterson, and Paul Hudak. Functional hybrid modeling. In *Proceedings of PADL'03: 5th International Workshop on Practical Aspects of Declarative Languages*, pages 376–390. Springer Verlag LNCS 2562, January 2003.
- [13] Paul Hudak, Antony Courtney, Henrik Nilsson, and John Peterson. Arrows, robots, and functional reactive programming. In *Summer School on Advanced Functional Programming, Oxford University*. Springer Verlag, LNCS 2638, 2003.
- [14] Zhanyong Wan, Walid Taha, and Paul Hudak. Event-driven FRP. In *Proceedings of Fourth International Symposium on Practical Aspects of Declarative Languages*. ACM, Jan 2002.
- [15] Walid Taha, Paul Hudak, and Zhanyong Wan. Directions in functional programming for real(-time) applications. In Thomas A. Henzinger and Christoph M. Kirsch, editors, *Proc. First International Workshop, EMSOFT, LNCS 2211*, pages 185–203, Tahoe City, CA, USA, October 2001. Springer-Verlag.
- [16] Zhanyong Wan, Walid Taha, and Paul Hudak. Real-time FRP. In *Proceedings of Sixth ACM SIGPLAN International Conference on Functional Programming*, Florence, Italy, September 2001. ACM.
- [17] Zhanyong Wan and Paul Hudak. Functional reactive programming from first principles. In *Proceedings of the ACM SIGPLAN '00 Conference on Programming Language Design and Implementation (PLDI)*, pages 242–252, Vancouver, BC, Canada, June 2000. ACM, ACM Press.
- [18] John Peterson, Paul Hudak, and Conal Elliott. Lambda in motion: Controlling robots with Haskell. In *First International Workshop on Practical Aspects of Declarative Languages*. SIGPLAN, Jan 1999.
- [19] John Peterson, Gregory Hager, and Paul Hudak. A language for declarative robotic programming. In *International Conference on Robotics and Automation*, 1999.
- [20] Alastair Reid, John Peterson, Greg Hager, and Paul Hudak. Prototyping real-time vision systems: An experiment in DSL design. In *Proc. Int'l Conference on Software Engineering*, May 1999.
- [21] Mark Tullsen and Paul Hudak. Shifting expression procedures into reverse. In Olivier Danvy, editor, *Proceedings of the ACM SIGPLAN Workshop on Partial Evaluation and Semantics-Based Program Manipulation*, Technical report BRICS-NS-99-1, University of Aarhus, pages 95–104, San Antonio, Texas, January 1999.
- [22] Paul Hudak. Modular domain specific languages and tools. In *Proceedings of Fifth International Conference on Software Reuse*, pages 134–142. IEEE Computer Society, June 1998.
- [23] Sebastian Shaumyan and Paul Hudak. Linguistic, philosophical, and pragmatic aspects of type-directed natural language parsing. In *Proceedings of Logical Aspects of Computational Linguistics*. Springer-Verlag, September 1997.

- [24] Conal Elliott and Paul Hudak. Functional reactive animation. In *International Conference on Functional Programming*, pages 263–273, June 1997.
- [25] Chih-Ping Chen and Paul Hudak. Rolling your own mutable adt—a connection between linear types and monads. In *Proceedings of 24th ACM Symposium on Principles of Programming Languages*, pages 54–66, New York, January 1997. ACM Press.
- [26] Paul Hudak. Haskore music tutorial. In *Second International School on Advanced Functional Programming*, pages 38–68. Springer Verlag, LNCS 1129, August 1996.
- [27] Sheng Liang and Paul Hudak. Modular denotational semantics for compiler construction. In *Proc. European Symposium on Programming*, pages 219–234. Springer-Verlag, April 1996.
- [28] Paul Hudak and Jonathan Berger. A model of performance, interaction, and improvisation. In *Proceedings of International Computer Music Conference*. Int’l Computer Music Association, 1995.
- [29] M.P. Jones, P. Hudak, and S. Shaumyan. Using types to parse natural language. In *Proceedings of Glasgow Functional Programming Workshop*. IFIP, Springer Verlag, 1995.
- [30] Rajiv Mirani and Paul Hudak. First class schedules and virtual maps. In *Proceedings of Conference on Functional Programming Languages and Computer Architecture*, pages 78–85. ACM/IFIP, June 1995.
- [31] Sheng Liang, Paul Hudak, and Mark Jones. Monad transformers and modular interpreters. In *Proceedings of 22nd ACM Symposium on Principles of Programming Languages*, pages 333–343, New York, January 1995. ACM Press.
- [32] Martin Odersky, Dan Rabin, and Paul Hudak. Call-by-name, assignment, and the lambda calculus. In *Proceedings of the Twentieth Annual ACM Symposium on Principles of Programming Languages, Charleston, South Carolina*, pages 43–57, January 1993.
- [33] K. Chen, P. Hudak, and M. Odersky. Parametric type classes. In *Proceedings of ACM Conference on Lisp and Functional Programming*, pages 170–181. ACM, June 1992.
- [34] A. Kishon, C. Consel, and P. Hudak. Monitoring semantics: a formal framework for specifying, implementing and reasoning about execution monitors. In *ACM SIGPLAN Conference on Programming Language Design and Implementation*, pages 338–352, June 1991.
- [35] R.S. Sundaresh and P. Hudak. Incremental computation via partial evaluation. In *Proceedings 18th Symposium on Principles of Programming Languages*, pages 1–13. ACM, January 1991.
- [36] S. Anderson and P. Hudak. Compilation of Haskell array comprehensions for scientific computing. In *ACM SIGPLAN Conference on Programming Language Design and Implementation*, pages 137–149, June 1990.
- [37] J. Guzmán and P. Hudak. Single-threaded polymorphic lambda calculus. In *Proceedings of Symposium on Logic in Computer Science*, pages 333–343. IEEE, June 1990.

- [38] R. Kelsey and P. Hudak. Realistic compilation by program transformation. In *ACM Symposium on Principles of Programming Languages*, pages 181–192, January 1989.
- [39] P. Hudak and E. Mohr. Graphinators and the duality of SIMD and MIMD. In *Proceedings 1988 ACM Conference on Lisp and Functional Programming*, pages 224–234, Salt Lake City, Utah, August 1988. ACM SIGPLAN/SIGACT/SIGART.
- [40] B. Goldberg and P. Hudak. Implementing functional programs on a hypercube multiprocessor. In *Proceedings of Third Conference on Hypercube Concurrent Computers and Applications*. ACM, January 1988.
- [41] P. Hudak and J. Young. A collecting interpretation of expressions (without powerdomains). In *Proceedings of ACM Symposium on Principles of Programming Languages*, pages 107–118, January 1988.
- [42] P. Hudak and S. Anderson. Pomset interpretations of parallel functional programs. In *Proceedings of 1987 Functional Programming Languages and Computer Architecture Conference*, pages 234–256. Springer Verlag LNCS 274, September 1987.
- [43] P. Hudak. Para-functional languages for parallel and distributed computing. In *Proceedings of Spring COMPCON '87*, pages 334–337. IEEE, February 1987.
- [44] A. Bloss and P. Hudak. Path semantics. In *Proceedings of Third Workshop on the Mathematical Foundations of Programming Language Semantics*, pages 476–489. Tulane University, Springer-Verlag LNCS Volume 298, 1987.
- [45] A. Bloss and P. Hudak. Variations on strictness analysis. In *Proceedings of the 1986 ACM Conference on LISP and Functional Programming*, pages 132–142. ACM, August 1986.
- [46] P. Hudak. Arrays, non-determinism, side-effects, and parallelism: A functional perspective. In *Proceedings of the Santa Fe Graph Reduction Workshop*, pages 312–327. Los Alamos/MCC, Springer-Verlag LNCS 279, October 1986.
- [47] K. Li and P. Hudak. Memory coherence in a shared virtual memory system. In *5th ACM Symposium on Principles of Distributed Computing*, pages 229–239. ACM SIGACT-SIGOPS, August 1986.
- [48] P. Hudak. A semantic model of reference counting and its abstraction (detailed summary). In *Proceedings 1986 ACM Conference on LISP and Functional Programming*, pages 351–363. ACM, August 1986.
- [49] D. Kranz, R. Kelsey, J. Rees, P. Hudak, J. Philbin, and N. Adams. Orbit: an optimizing compiler for Scheme. In *SIGPLAN '86 Symposium on Compiler Construction*, pages 219–233. ACM, June 1986. Published as SIGPLAN Notices Vol. 21, No. 7, July 1986.
- [50] P. Hudak and L. Smith. Para-functional programming: a paradigm for programming multiprocessor systems. In *12th ACM Symposium on Principles of Programming Languages*, pages 243–254, January 1986.

- [51] P. Hudak and J. Young. Higher-order strictness analysis for untyped lambda calculus. In *12th ACM Symposium on Principles of Programming Languages*, pages 97–109, January 1986.
- [52] J.H. Fasel, P. Hudak, R.J. Douglass, and R. Michelson. A distributed implementation of functional program evaluation. In *Proceedings of AI-85*, May 1985.
- [53] P. Hudak and B. Goldberg. Serial combinators: “optimal” grains of parallelism. In *Functional Programming Languages and Computer Architecture*, pages 382–388. Springer-Verlag LNCS 201, September 1985.
- [54] P. Hudak and A. Bloss. The aggregate update problem in functional programming systems. In *12th ACM Symposium on Principles of Programming Languages*, pages 300–314. ACM, January 1985.
- [55] P. Hudak and B. Goldberg. Experiments in diffused combinator reduction. In *Proceedings 1984 ACM Conference on LISP and Functional Programming*, pages 167–176. ACM, August 1984.
- [56] P. Hudak and D. Kranz. A combinator-based compiler for a functional language. In *11th ACM Symposium on Principles of Programming Languages*, pages 121–132. ACM, January 1984.
- [57] P. Hudak. Distributed task and memory management. In *Proceedings of Symposium on Principles of Distributed Computing*, pages 277–289. ACM, August 1983.
- [58] P. Hudak and R.M. Keller. Garbage collection and task deletion in distributed applicative processing systems. In *Proceedings 1982 ACM Conference on LISP and Functional Programming*, pages 168–178. ACM, August 1982.

Research Reports and Other Papers

- [1] Andreas Voellmy, Ashish Agarwal, and Paul Hudak. Nettle: Functional reactive programming for openflow networks. Technical Report YALEU/DCS/RR-1431, Yale University, July 2010.
- [2] Andreas Voellmy, Ashish Agarwal, Paul Hudak, Nick Feamster, Sam Burnett, and John Launchbury. Don’t configure the network, program it! domain-specific programming languages for network systems. Technical Report YALEU/DCS/RR-1432, Yale University, July 2010.
- [3] Eric Cheng and Paul Hudak. Audio processing and sound synthesis in Haskell. Technical Report YALEU/DCS/RR-1405, Yale University, January 2009.
- [4] Paul Hudak, Paul Liu, Michael Stern, and Ashish Agarwal. Yampa meets the worm. Technical Report YALEU/DCS/RR-1408, Yale University, July 2008.
- [5] Paul Hudak. Polymorphic temporal media. Technical Report YALEU/DCS/RR-1259, Yale University, Department of Computer Science, August 2003.

- [6] Liwen Huang and Paul Hudak. Dance: A declarative language for the control of humanoid robots. Technical Report YALEU/DCS/RR-1253, Yale University, Department of Computer Science, July 2003.
- [7] John Peterson, Zhanyong Wan, Paul Hudak, and Henrik Nilsson. *Yale FRP User's Manual*. Department of Computer Science, Yale University, January 2001. Available at <http://www.haskell.org/frp/manual.html>.
- [8] John Peterson, Paul Hudak, and Gary Shu Ling. Principled dynamic code improvement. Research Report YALEU/DCS/RR-1135, Yale University, Department of Computer Science, July 1997.
- [9] Mark Tullsen and Paul Hudak. An intermediate meta-language for program transformation. Technical Report YALEU/DCS/RR-1154, Yale University, June 1998.
- [10] P. Hudak. Mutable abstract datatypes – or – how to have your state and munge it too. Research Report YALEU/DCS/RR-914, Yale University, Department of Computer Science, December 1992.
- [11] W.E. Carlson, P. Hudak, and M.P. Jones. An experiment using Haskell to prototype "geometric region servers" for navy command and control. Research Report 1031, Department of Computer Science, Yale University, November 1993.
- [12] M.P. Jones and P. Hudak. Implicit and explicit parallel programming in haskell. Research Report YALEU/DCS/RR-982, Department of Computer Science, Yale University, New Haven, Connecticut, Nov 1993.
- [13] P. Hudak and M.P. Jones. Haskell vs. ada vs. c++ vs. awk vs. ... an experiment in software prototyping productivity. Research Report YALEU/DCS/RR-1049, Department of Computer Science, Yale University, New Haven, CT, Oct 1994.
- [14] R.S. Sundaresh and P. Hudak. A theory of incremental computation and its application. Research Report YALEU/DCS/RR770, Yale University, Department of Computer Science, March 1990.
- [15] P. Varma and P. Hudak. Memo-functions in ALFL. Technical Report YALEU/DCS/RR-759, Yale University Department of Computer Science, December 1989.
- [16] S. Anderson and P. Hudak. Efficient compilation of Haskell array comprehensions. Research Report YALEU/DCS/RR693, Yale University, Department of Computer Science, March 1989.
- [17] Z.G. Mou, S. Anderson, and P. Hudak. Parallelism in sequential divide-and conquer (extended abstract). Research Report YALEU/DCS/RR-683, Yale University, Department of Computer Science, February 1989.
- [18] J. Young and P. Hudak. Finding fixpoints on function spaces. Research Report YALEU/DCS/RR-505, Yale University, Department of Computer Science, November 1986.
- [19] J. Guzmán and P. Hudak. Provably complete operational semantics for first-order lazy narrowing. Research Report YALEU/DCS/RR-525, Yale University, Department of Computer Science, March 1987.

- [20] P. Hudak and S. Anderson. Haskell solutions to the language session problems at the 1988 Salishan High-Speed Computing Conference. Technical Report YALEU/DCS/RR-627, Yale University, Department of Computer Science, January 1988.
- [21] P. Hudak, J-M. Delosme, and I. Ipsen. Parlance: A para-functional programming environment for parallel and distributed computing. Research Report YALEU/DCS/RR-524, Yale University, Department of Computer Science, April 1987.
- [22] B. Goldberg and P. Hudak. Detecting sharing of partial applications in functional languages. Research Report YALEU/DCS/RR-526, Yale University, Department of Computer Science, March 1987.
- [23] P. Hudak. Collecting interpretations of expressions. Research Report YALEU/DCS/RR-497, Yale University, Department of Computer Science, 1986.
- [24] P. Hudak. Functional programming on multiprocessor architectures. Research Report YALEU/DCS/RR-447, Yale University, Department of Computer Science, December 1985.
- [25] P. Hudak and J. Guzmán. A proof-stream semantics for lazy narrowing. Research Report YALEU/DCS/RR-446, Yale University, Department of Computer Science, December 1985.
- [26] K. Li and P. Hudak. A new list compaction method. Research Report YALEU/DCS/RR-362, Yale University, February 1985.
- [27] P. Hudak and J. Young. A set-theoretic characterization of function strictness in the lambda calculus. Research Report YALEU/DCS/RR-391, Yale University, June 1985.
- [28] P. Hudak and L. Smith. Para-functional programming: a paradigm for programming multiprocessor systems. Research Report YALEU/DCS/RR-390, Yale University, June 1985.
- [29] P. Hudak. Distributed applicative processing systems: Project goals, motivation and status report. Research Report YALEU/DCS/RR-317, Yale University, May 1984.
- [30] P. Hudak. ALFL Reference Manual and Programmer's Guide. Research Report YALEU/DCS/RR-322, Second Edition, Yale University, October 1984.
- [31] P. Hudak. Distributed graph marking. Research Report YALEU/DCS/RR-268, Yale University, January 1983.
- [32] P. Hudak. Call-graph reclamation: an alternative storage reclamation scheme. AMPS Technical Memorandum 4, Department of Computer Science, University of Utah, August 1981.

Courses Taught at Yale

- Advanced Compiler Design
- Advanced Programming Languages Seminar (with Zhong Shao)
- Autonomous Systems
- Category Theory for Computer Scientists
- Compilers and Interpreters
- Computer Music Seminar
- Data Structures
- Formal Semantics of Programming Languages
- Fundamentals of Computer Music I: Algorithmic and Heuristic Composition
- Fundamentals of Computer Music II: Sound Representation and Synthesis
- Introduction to Computer Music
- Introduction to Computer Science
- Introduction to Programming (in both Java and C#)
- Parallel Programming Languages
- Software Engineering
- Dataflow Seminar (with Josh Fisher)
- Partial Evaluation Seminar (with Charles Consel)
- Programming Language Seminar (with Alan Perlis, 4 semesters)
- Type Theory Seminar (with Martin Odersky)

Graduated PhD Students Supervised

1. Adrienne Bloss, thesis title: *Path Analysis: Using Order-of-Evaluation Information to Optimize Lazy Functional Languages*, 1989. Currently Professor, Department of Computer Science, and Assistant Dean for Curricular and Faculty Development, Roanoke College.
2. Kung Chen, thesis title: *A Parametric Extension of Haskell's Type Classes*, 1994. Currently Associate Professor, Department of Computer Science, National Chengchi University, Taiwan.
3. Chih-Ping Chen, thesis title: *Mutable Abstract Datatypes—A Connection Between Linear Types and State Monads*, 1999. Currently Software Engineer, Software Solution Group, Intel Corporation, Nashua, NH.
4. Antony Courtney, thesis title: *Modeling User Interfaces in a Functional Language*, 2004. Currently Vice President at Goldman Sachs, New York, NY.

5. Benjamin Goldberg, thesis title: *Multiprocessor Execution of Functional Programs*, 1988. Currently Associate Professor, Department of Computer Science, New York University.
6. Juan Guzman, thesis title: *On Expressing the Mutation of State in a Functional Programming Language*, 1992. Currently Associate Professor, Department of Business Administration, Southern Polytechnic State University, Marietta, GA.
7. Liwen Huang, thesis title: *Functional Reactive Control of Humanoid Robots*, 2007. Currently Member of Technical Staff, Oracle, Redwood Shores, CA.
8. Richard Kelsey, thesis title: *Realistic Compilation by Program Transformation*, 1989. Currently Software Architect, Ember Corporation, Boston, MA.
9. Siau-Cheng Khoo, thesis title: *Parameterized Partial Evaluation*, 1992. Currently Associate Professor, Department of Computer Science, National University of Singapore.
10. Amir Kishon, thesis title: *Theory and Art of Semantics-Directed Program Execution Monitoring*, 1992. Currently Chairman and CEO, Wellness Layers, Inc., East Rockaway, NY.
11. David Kranz, thesis title: *Orbit: A Compiler for T/Scheme*, 1988. Currently Vice President of Engineering, and Chief Technology Officer, Curl, Inc., Cambridge, MA.
12. Kai Li, thesis title: *Shared Virtual Memory on Loosely-Coupled Multiprocessors*, 1986. Currently Professor, Department of Computer Science, Princeton University.
13. Hai (Paul) Liu, thesis title: *The Theory and Practice of Causal Commutative Arrows*, 2010. Currently researcher at Intel, Inc.
14. Kevin Lynch, thesis title: *Programming in Distributed Systems Logic*, 1997 (co-advised with Alan Perlis). Currently working in industry.
15. Sheng Liang, thesis title: *Modular Monadic Semantics and Compilation*, 1997. Currently President and CEO, VMOps, Inc, Cupertino, CA.
16. Rajiv Mirani, thesis title: *High Level Abstractions for Parallel Functional Programming*, 1996. Currently Senior Director of Engineering, Citrix Systems, San Francisco, CA.
17. Eric Mohr, thesis title: *Dynamic Partitioning of Parallel Lisp Programs*, 1991, currently self-employed.
18. Z. George Mou, thesis title: *A Formal Model for Divide and Conquer and its Parallel Realization*, 1990. Currently President, Sinovate, LLC, Beijing, China.
19. James Philbin, thesis title: *The Design of an Operating System for Modern Programming Languages*, 1992. Currently Director, Center for Biomedical Imaging and Informatics, Johns Hopkins University, Baltimore, MD.
20. Daniel Rabin, thesis title: *Calculi for Functional Programming Languages with Assignment*, 1996. Currently Software Engineer, Google, Inc., Mountain View, CA.
21. Martin Sulzmann, thesis title: *A General Framework for Hindley-Milner Type Systems with Constraints*, 2000 (co-advised with Martin Odersky). Currently Systems Engineer, Informatik Consulting Systems AG, Stuttgart, Germany.

22. Raman Sundaresh, thesis title: *Incremental Computation via Partial Evaluation*, 1991. Currently Managing Director, Hutchin Hill Capital, New York, NY.
23. Mark Tullsen, thesis title: *PATH, A Program Transformation System for Haskell*, 2002. Currently Member of Senior Technical Staff, Galois Inc., Portland, OR.
24. Jonathan Young, thesis title: *The Semantic Analysis of Functional Programs: Theory and Practice*, 1989. Currently Research Scientist, MIT Laboratory for Computer Science.
25. Zhanyong Wan, thesis title: *Functional Reactive Programming for Real-Time Embedded Systems*, 2002 (co-advised with Walid Taha). Currently Staff Software Engineer, Google, Inc., Seattle, WA.

Postdoctoral Students Supervised

1. Ashish Agarwal, currently Research Scientist in the Genomics Core Computing Lab at NYU.
2. Mark Jones, currently Professor, Department of Computer Science, Portland State University.
3. Martin Odersky, currently Professor, School of Computer and Communications Sciences, Swiss Federal Institute of Technology, Lausanne, France.
4. Henrik Nilson, currently Lecturer, School of Computer Science, University of Nottingham, England.
5. Valery Trifonov, currently doing software development at Goldman Sachs, New York, NY.
6. Charles Consel, currently Professor, Enseirb (Ecole Nationale Suprieure dElectronique et de Radiolctricit de Bordeaux), and Inria / LaBRI (Laboratoire Bordelais de Recherche en Informatique), Bordeaux, France.
7. Alastair Reid, currently software engineer, ARM Limited, Cambridge, UK.
8. Sandra Loosemore, currently Software Engineer, CodeSourcery, Granite Bay, CA.
9. John Peterson, currently Associate Professor, Math and Computer Information Science Department, Western State College, Gunnison, CO.
10. Walid Taha, currently Assistant Professor, Department of Computer Science , Rice University, Houston, TX.

A professional appointment is a scheduled visit with some type of professional who renders services to clients or medical patients. To schedule an appointment with: * a dentist (or...
What is a professional appointment? Ad by JetBrains. The issue tracker for every team in your company. PROFESSIONAL APPOINTMENTS. Open All. Get Help. Introduction (320.1 to 320.2). 320.1. Professional accountants are required to comply with the fundamental principles and apply the conceptual framework set out in Section 120 to identify, evaluate and address threats. 320.2. Acceptance of a new client relationship or changes in an existing engagement might create a threat to compliance with one or more of the fundamental principles. Visiting Academic Appointments Professor, Computer Science and Engineering, IIT Bombay, 2009 – 2013. Visiting Professor, Kanwal Rekhi School of Information Technology, IIT Bombay, Fall 2006. III. Honors and Awards. Professional Society Honors. Leadership Roles in Professional Organizations. 1. Chair, ACM Special Interest Group on Bioinformatics, Computational Biology, and Biomedical Informatics, 2015 – present. 2. IEEE Computer Society Fellows Committee, 2010 – 2015, 2018. Need to translate "PROFESSIONAL APPOINTMENTS" from english and use correctly in a sentence? Here are many translated example sentences containing "PROFESSIONAL APPOINTMENTS" - english-spanish translations and search engine for english translations. Translation of Professional appointments in Spanish. Results: 205876, Time: 0.0363. Professional. Professional Appointments, Pretoria, South Africa. 195 likes. Professional Appointments. See more of Professional Appointments on Facebook. Log In. or. Create New Account. See more of Professional Appointments on Facebook. Log In. Forgotten account?