Philip Wadler

School of Informatics, University of Edinburgh
10 Crichton Street, Edinburgh EH8 9AB, SCOTLAND
wadler@inf.ed.ac.uk
http://homepages.inf.ed.ac.uk/wadler/
+44 131 650 5174 (W), +44 7976 507 543 (M)

Citizen of United States and United Kingdom. Born: 1956
Father of Adam and Leora Wadler. Separated from Catherine Lyons.

1 Education

1984. Ph.D., Computer Science, Carnegie-Mellon University. Dissertation title: *Listlessness is Better than Laziness.* Supervisor: Nico Habermann. Committee: James Morris, Guy Steele, Bill Scherlis.

1979. M.S., Computer Science, Carnegie-Mellon University.

1977. B.S., Mathematics, with honors, Phi Beta Kappa, Stanford University.

Awards

National Science Foundation Fellow, three year graduate fellowship.

1975 ACM Forsythe Student Paper Competition, first place.

2 Employment

2003–present. Edinburgh University. Professor of Theoretical Computer Science.

2017–present. IOHK. Senior Research Fellow, Area Leader Programming Languages.

1999–2003. Avaya Labs. Member of Technical Staff.

1996–1999. Bell Labs, Lucent Technologies. Member of Technical Staff.

1987–1996. University of Glasgow. Lecturer, 1987–90; Reader, 1990–93; Professor, 1993–96.

1983–1987. Programming Research Group and St. Cross College, Oxford. Visiting Research Fellow and ICL Research Fellow.

Awards

Most Influential POPL Paper Award 2003 (for 1993)
for Imperative Functional Programming by Simon Peyton Jones and Philip Wadler.

Royal Society Wolfson Research Merit Award, 2004–2009.

Fellow, Royal Society of Edinburgh, 2005–.

Fellow, Association for Computing Machinery, 2007–.

EUSA Teaching Awards, Overall High Performer, runner up, 2009.

3 Research overview

Hybrid trees yield the most robust fruit. My research in programming languages spans theory and practice, using each to fertilize the other. I have an h-index of 60 on Google Scholar, and appear at position 6 in a list of the most-acknowledged computer science authors.

Haskell Haskell is the most widely used language for lazy functional programming. I served as first editor of the Haskell report (the three other editors are Hudak, Hughes, and Peyton Jones), and contributed Haskell's two main language innovations: type classes to support overloading (joint with Blott); and monads to support interaction and concurrency (joint with Peyton Jones). The latter won the Most Influential POPL Paper Award 2003 (for 1993).

Java Java 5.0 supports generic types. With Bracha, Odersky, and Stoutamire, I devised GJ (Generic Java), the basis for Sun's design. GJ is one of dozens of designs based on the same theory; it succeeded in impacting industrial practice because we stressed simplicity and backward compatibility. With Igarashi and Pierce, I devised FJ (Featherweight Java), a formal model of Java that fits on one page; the conference and journal versions jointly have over 1000 citations on Google Scholar. GJ pushes theory into practice; FJ pulls practice into theory.

XML XQuery is the W3C standard for querying XML data ('SQL for XML'). With Fernández and Siméon, I put forward proposals for the core algebra, formal semantics, and static type system for XQuery, one of the first uses of formal semantics in an industrial standard.

Links Links generates three tiers of a web application from a single source: it runs interpreted on the server for program logic; it compiles to Javascript on the front end for user interface; and it compiles to SQL on the back end for database access. Our research on Links has led to improvements in Microsoft's LINQ, Language Integrated Query. The *formlets* we devised for Links have been reimplemented for commercial use by Tupil and Intellifactory.

From Data Types to Session Types Session Types, invented by Honda, describe communication protocols in a typed process calculus. I was one of the first to apply linear logic to functional programming, and recent work by Caires and Pfenning, further extended by myself, demonstrates a close tie between session types and linear logic. The tie to logic suggests that session types are far from arbitrary, and may provide a basis for concurrency and distribution. I lead a £3.9M EPSRC programme grant, including Gay (Glasgow) and Yoshida (Imperial), to push session types from research towards a practical tool.

Service I am area editor, programming languages, for *Journal of the ACM* (2013–), and I served as as Chair and Past Chair of SIGPLAN (2009–2015); founding editor and editor-in-chief of the *Journal of Functional Programming* (1991–2003); as program chair for ICFP (2000), Plan-X (2002), PADL (2003), Erlang (2004), FOOL (2005), FLOPS (2006) and POPL (2008); and on NSF and EPSRC panels. Invited speaking engagements include POPL (Albuquerque, 1992), ILPS (Portland, 1995), Software Reuse (Victoria, 1998), DSL (Austin, 1999), TCS (New Delhi, 2000), ICDT (London, 2001), VLDB (Rome, 2001), FLOPS (Aizu, 2002), RTA (Nara, 2005), OOPSLA (Portland, 2006), FMCO (Amsterdam, 2006), AOSD (Brussels, 2008), and ETAPS (Cyprus, 2010).

4 Teaching overview

I enjoy teaching. I started early: when I was a Stanford undergraduate, Bob Floyd gave me the unusual opportunity to teach introductory Pascal programming to my fellow undergraduates, with graduate students TAing the course.

At Glasgow, I initiated 1'st year and MSc courses on social implications of computing; a 2'nd year course on functional programming; 3'rd year, 4'th year, and MSc courses on communications skills; and a 4'th year course on type theory. This selection of courses reflects my belief that computing must draw on the arts as well as the sciences. I served for three years as class head for 1'st year students, a major administrative responsibility. I secured a grant to improve teaching of communication skill in 1'st year.

At Edinburgh, I initiated teaching of functional programming in the first course seen by incoming Informatics students. This course was awarded runner-up for Overall High Performer in the first EUSA Teaching Awards in 2009. I teach Types and Sematics for Programming Languages to 4'th year and MSc students, based on Pierce's *Software Foundations*; and Professional Issues to 3'rd year students, with an emphasis on communication skills. For five years I taught the mandatory System Design Project to 3rd year students, in which teams collaborated on building robots to play football.

I liaison with secondary schools, running special programs to attract students to apply to Informatics. I ran an after school club on programming Mindstorms robots for First Lego League at my children's school, James Gillespie's Primary.

I taught at several summer schools, including the Marktoberdorf International Summer School (1992) and Advanced Functional Programming (Båstad 1995 and Oxford 2002). I organized and secured funding for a Summer School on Functional Programming and Constructive Logic (Glasgow 1989). I supervised numerous MSc and 4'th year projects, and graduated nine PhD students.

The textbook written by Richard Bird and myself, *Introduction to Functional Programming*, is used in 1'st and 2'nd year courses in the UK, US, and Australia, and has been translated into German, Dutch, and Japanese.

4.1 Summary of teaching experience

Introductory Pascal programming (Stanford and Carnegie-Mellon).

Functional programming (Oxford, Glasgow, Edinburgh), 1'st year, 2'nd year, 4'th year, and MSc. I initiated the Glasgow 2'nd year course and Edinburgh 1'st year course.

Denotational semantics (Göteborg), 4'th year.

Type theory (Glasgow, Edinburgh), 4'th year. I initiated the Glasgow course.

Mathematical foundations (Glasgow), 1'st year.

Compilers (Glasgow), 2'nd year.

Computing and Society (Glasgow), 1'st year and MSc. I initiated these courses.

Communication skills (Glasgow, Edinburgh), 3'rd year, 4'th year, and MSc. I initiated the Glasgow and Edinburgh courses.

Systems Design Project (Edinburgh), 3'rd year.

Professional Issues (Edinburgh), 3'rd year.

Types and Semantics for Programming Languages (Edinburgh), 4'th year and MSc.

Administration. Class head, 1'st year (Glasgow) three years; assistant class head, 1'st year (Glasgow) one year; assistant class head, 2'nd year (Glasgow) two years.

Project supervision. Supervised numerous MSc dissertations (Oxford, Edinburgh), and 4'th year projects (Glasgow, Edinburgh).

4.2 Schools

Betty Summer School on Behavioral Types, 30 June–4 July 2014, Lovan, Croatia and 27 June–1 July 2016, Limassol, Cyprus.

International Summer School on Trends in Computing, Tarragona, 22–26 July 2013 and 7–11 July 2014.

Midlands Graduate School in the Foundations of Computing Science, Leicester, 8–12 April 2013, invited lecturer.

Integrating Static and Dynamic Types Systems, PhD Open, 28–29 October and 2–3 December 2011, University of Warsaw.

Advanced Functional Programming, Båstad 1995 and Oxford 2002.

Marktoberdorf International Summer School, 1992.

Summer School on Functional Programming and Constructive Logic, Glasgow, 1989.

4.3 Doctoral supervision

Oxford.
David Lester, 1988.
Phil Trinder, 1989.

Glasgow.
Stephen Blott, 1991.
Kei Davis, 1994.
Alex Ferguson, 1995.
Simon Marlowe, 1995.

Edinburgh.
DeLesley Hutchins, 2008.
Ezra Cooper, 2009.
Jeremy Yallop, 2009.
Shayan Javadipour (current, to finish 2017).
Simon Fowler (current, to finish 2018).
Jack Williams (current, to finish 2018).
Jakub Zalewski (current, to finish 2019).
Pepijn Kokke (current, to finish 2020).

5 Other employment

5.1 Visiting positions

2003 (May). Visitor, École-Normale Supérieure, Paris, France.

1994 (December). Visitor, Technical University of Sydney, Australia.

1991 (January-February). Guest professor, University of Sydney, Australia.

1989 (August). Guest professor, DIKU, University of Copenhagen.

1986–7 (winter term). Visiting Research Fellow. Programming Methodology Group, Chalmers University of Technology, Göteborg, Sweden.

5.2 Internships

1980 (summer). Bell Labs, Murray Hill. Brian Kernighan, supervisor. Research Intern.

1979 (summer). Xerox Palo Alto Research Center. James Morris, supervisor. Research Intern.

1977–78 (summers). Xerox Palo Alto Research Center. Clarence Ellis, supervisor. Research Intern.

1976 (summer). Stanford Artificial Intelligence Laboratory. John McCarthy, supervisor. Technical Assistant.

1974–76. Stanford University. Vinton Cerf, supervisor. Technical Assistant.

1977 (one quarter). Stanford University. Bob Floyd, supervisor. Lecturer on introductory programming course.

1973 (summer). NSF Summer Program in Mathematics. Lecturer.

5.3 Consulting

2010–. Hypernumbers.

1994–1997. Ericsson Telecommunications.

1987. Masthaven Development Limited.

1986–7. Royal Signal and Radar Establishment, Aldermaston.

1976. R.G.B. Laboratories.

1975–6. Intersil, Inc.

6 Professional activities

6.1 Editorial

Journal of the ACM, area editor, programming languages, 2013–present.

Journal of Object Technology, editorial board, 2010–2014.

Logical Methods in Computer Science, editorial board, 2004–2013.

Journal of Functional Programming, Cambridge University Press. Editor-in-Chief and editor, 1990–2004, editorial board, 2004–2008.

Types Forum, Moderator, 1992–1996. The Types Forum is an electronic mailing list initiated by Albert Meyer at MIT.

Functional Programming Column, ACM SIGPLAN Notices, Editor, 1997–2001.

Functional Programming Column, BCS FACS/FME Newsletter, Editor, 1994–2003.

Computer Journal, Guest editor, special issue on lazy functional programming, April 1989.

6.2 Professional groups

ACM SIGPLAN, Executive Board, Member-at-Large, 2003–2007; Chair, 2009–2012; Past Chair, 2012–2015.

Charter member of IFIP WG2.8, Functional Programming, 1988–present.

ACM Lectureship Program, 1989–1993.

6.3 Steering committees

International Conference on Functional Programming (ICFP), 2000–2003 & 2009–2012,

Principles of Programming Languages (POPL), 2008–15.

Programming Langauages Design and Implementation (PLDI), 2009–2018.

Object-Oriented Programming Systems, Languages, and Applications (OOPSLA/SPLASH), 2009–2015.

European Symposium on Programming (ESOP), 2013–2019.

Formal Structures for Computation and Deduction (FSCD), 2014–2016.

6.4 Program chair

Principles of Programming Languages (POPL), San Francisco, January 2008.

Functional and Logic Programming (FLOPS), Fuji, April 2006.

Foundations of Object-Oriented Languages (FOOL), Long Beach, January 2005.

Erlang Workshop, Snowbird, September 2004.

Practical Aspects of Declarative Languages (PADL), New Orleans, January 2003.

Programming Language Technologies for XML (PLAN-X), Pittsburgh, October 2002.

International Conference on Functional Programming (ICFP), Vancouver, September 2000.

6.5 Local arrangements

International Conference on Functional Programming (ICFP), Edinburgh, 31 August–2 September 2009.

IFIP Working Group on Functional Programming (WG2.8), Edinburgh, 11–16 June 2017.

6.6 Organizing committee

Relations and Data Integrity Constraints and Languages (RADICAL), Cambridge, 10–11 May 2010.

6.7 Program committee

Practical Aspects of Declarative Languages (PADL), Paris, 16–17 January 2017.

Evaluation and Usability of Programming Languages and Tools (PLATEAU), Amsterdam, 1 November 2016.

European Symposium on Programming (ESOP), Grenoble, April 2014.

Evaluation and Usability of Programming Languages and Tools (PLATEAU), Tucson, Arizona, 21 October 2012.

Principles and Practice of Declarative Programming (PPDP), Hagenberg, Austria, 26–28 July 2010.

Script to Program evolution (STOP), Genova, 6 July 2009.

European Symposium on Programming (ESOP), Budapest, 29 March–6 April 2008.

OOPSLA Essays, Portland, 22-6 October 2006

Classical Logic and Computation (CL&C), Venice, July 2006.

XQuery Implementation, Experience and Perspectives (XIME-P), Chicago, 30 June 2006.

Principle of Programming Languages (POPL), Charleston, 11–13 January 2006.

Trends in Functional Programming (TFP), Munich, 25-26 November 2004.

European Conference on Object-Oriented Programming (ECOOP), Oslo, 14–18 June 2004

Workshop on the Web and Databases (WebDB), Paris, 17–18 June 2004.

Symposium on Applied Computing (SAC), Object-Oriented Programming Languages and Systems (OOPS) track, Nicosia, 14–17 March 2004.

Symposium on Functional and Logic Programming (FLOPS), Nara, 7–9 April 2004.

Foundations of Object-Oriented Languages (FOOL), Venice, 17 January 2004.

Types in Language Design and Implementation (TLDI), New Orleans, 18 January 2003.

International Conference on Distributed Computing Systems (ICDCS), Vienna, 2–5 July 2002.

World Wide Web Conference (Posters), Honolulu, 7-11 May 2002.

Database Programming Languages (DBPL), Marino, Rome, September 2001.

European Association for Computer Science Logic (CSL), Paris, September 2001.

International Workshop on the Web and Databases (WebDB), in conjunction with SIGMOD, Santa Barbera, May 2001.

International Conference on Distributed Computing Systems (ICDCS), Vienna, 2-5 July 2002.

Object-Oriented Programming Systems, Languages, and Applications (OOPSLA), Minneapolis, October 2000.

Mathematics of Program Construction (MPC), Pont de Lima, July 2000.

ECOOP Workshop on XML and Object Technology (XOT), Sophia Antipolis, June 2000.

Haskell Workshop, Paris, October 1999.

Symposium on Generative and Component-based Software Engineering (GCSE), Erfurt, September 1999.

Latin American Conference on Functional Programming (CLAPF), Recife, March 1999.

Practical Aspects of Declarative Languages (PADL), Austin, January 1999.

International Conference on Functional Programming (ICFP), Amsterdam, June 1997.

Programming Language Design and Implementation (PLDI), Las Vegas, June 1997.

Computing: The Australia Theory Seminar (CATS), 1997.

Asian Computing Science Conference, 1996.

Principles of Programming Languages (POPL), 1996.

Static Analysis Symposium (SAS), 1995.

European Symposium on Programming (ESOP), 1990 and 1992.

6.8 Judge

Twenty years of PLDI, ACM. (Best papers from the first twenty years of *Programming Language Design and Implementation.*)

Software Carpentry tools competition, July 2000.
`http://www.software-carpentry.com/`

6.9 NSF Panel

NSF review panel, 30 January 2003.

NSF review panel, 28 January 1999.

NSF review panel, 5 February 1998.

6.10 Organizer

The Essence of Programming: In Honor of John Reynolds, Special Session of MFPS, Birmingham, 18 May 2005 [with Olivier Danvy and Peter O'Hearn].

Component-based programming in multiple paradigms, Dagstuhl, 21–26 February 1999 [with Karsten Weihe].

Workshop on High-Level Concurrent Languages, Dagstuhl, 20–22 January 1997 [with Kohei Honda, Martin Odersky, Benjamin Pierce, Gert Smolka].

Workshop on Syntactic Control of Interference and Linear Logic (SCILL), Glasgow, 31 July–3 August 1995 [with John Reynolds].

Summer School on Functional Programming and Constructive Logic, SERC Logic for IT Initiative, Glasgow, September 1989.

6.11 External examiner

EPFL, Lausanne (PhD examiner)
Chalmers University of Technology (PhD examiner)
Oxford University (Master's degree programme)
Birmingham University (PhD examiner)
Imperial College London (PhD examiner, three times)
Karlsruhe University (Habilitation examiner)
Manchester University (PhD examiner)
Oxford University (PhD examiner)
Stirling University (PhD examiner)
Tel Aviv University (PhD examiner)
University VII Paris (Habilitation examiner)

Member, Board of Advisors, Standing panel of experts in computer science, University of London, 1993–1996.

7 Grants

EPSRC (EP/K034413/1), "From data types to session types: A Basis for Concurrency and Distribution", programme grant, £3,900,000, 60 months, June 2013–May 2018, extended to May 2019. With Simon Gay and Nobuko Yoshida.

EPSRC (EP/D046769/1), "Links: Linking Theory to Practice for the Web", £292,000, 36 months, January 2006–December 2008.

Ericsson Telecommunications, "A type system for Erlang (renewal)", £65,000, 9 months, April 1996–December 1996.

EPSRC (GR/L 01053) "Visiting fellowship for A. Appel", £47,488, 12 months, July 1996–June 1997. With Simon Peyton Jones and Malcolm Atkinson.

Ericsson Telecommunications, "A type system for Erlang", £70,000, 12 months, April 1995–March 1996.

ESPRIT (ECUS025) "Atlantique: EC-US collaborative activity, semantics based program manipulation", Working Group, 125,000 ECU, 36 months, July 1993–June 1996. The partner universities are Copenhagen, Ecole Normale Supérieure, Imperial College, Glasgow, Göteborg, Århus in Europe, and Carnegie Mellon, City University of New York, Kansas State, Northeastern, New York Courant Institute, Oregon Graduate Institute, Stanford, and Yale in the US.

EPSRC (GR/K 26608) "Visiting fellowship for S. Sokolowski", £8,094, 2 months, April–May 1995.

EPSRC (GR/J 52709) "Save space with linear types", £131,923, 36 months, April 1994–March 1997. With Simon Peyton Jones.

ESPRIT (6809) "Semantique: Semantics-based program manipulation techniques", Working group, 180,000 ECU, 36 months, July 1992–July 1995. With Flemming and Hanne Nielson (Aarhus), John Launchbury (Glasgow), Chris Hankin, Geoffrey Burn, and Samson Abramsky (Imperial), Patrick Cousot and Radhia Cousot (Ecole Normale Superiure), Neil Jones and Mads Tofte (Copenhagen).

SERC (GR/H 78801) "Declarative systems architecture: A quantitative approach", £252,000, 3 years, Jan 1993–December 1996. With Simon Peyton Jones and John Launchbury.

SERC (GR/J 12819) "Visiting fellowship for R. J. Popplestone", £8834, 6 months, July–December 1993.

SERC/IED (GR/F 34671), "GRASP Infrastructure Project", £233,817, 36 Months, July 1989–June 1992. With Simon Peyton Jones (UCL/Glasgow) and Jeff Reynolds (Essex).

ESPRIT (3124), "Semantique: Semantics-based program manipulation techniques", Basic Research Action, 1,100,000 ECU, 30 Months, May 1989–October 1991 (extended until May 1992). With John Hughes (Glasgow), Samson Abramsky and Chris Hankin (Imperial), Patrick Cousot and Radhia Cousot (Ecole Polytechnic), Neil Jones (Copenhagen).

SERC (GR/G 54566) "Visiting fellowship for Dale Miller", £8560, 4 months, July–October 1991.

SERC (GR/F 28892), "Visiting fellowship for Thomas Johnsson", £23,356, 12 Months, September 1989–August 1990. With John Hughes (Glasgow).

SERC (SO/128/89), "Vacation school for Logic for IT initiative", £16,200, Summer 1989.

SERC (GR/F 28946), "Compile-time Analysis and Type Systems", £3600, 24 Months, January 1990–December 1991. With John Hughes (Glasgow).

SERC (GR/E 85799), “Committee establishing a standard lazy functional language”, £2500, 18 Months, February 1988–August 1989. With John Hughes (Glasgow) and Simon Peyton Jones (UCL).

Enterprise in Higher Education Unit, £2000, grant to improve teaching of communication skills in first-year computing science, October 1992–September 1993.

8 Publications

8.1 Books

Java Generics and Collections. Maurice Naftalin and Philip Wadler. O'Reilly, 2006.

> "A brilliant exposition of generics. By far the best book on the topic, it provides a crystal clear tutorial that starts with the basics and ends leaving the reader with a deep understanding of both the use and design of generics." — Gilad Bracha, Java Generics Lead, Sun Microsystems

Introduction to Functional Programming. Richard Bird and Philip Wadler. Prentice-Hall International Series in Computer Science, C.A.R. Hoare, series editor, 1988. [Japanese translation, 1991. Dutch translation, 1991. German translation, 1992.]

> "The significance of *Introduction to Functional Programming* is that with its publication the necessary revolution in Computer Science/Software Engineering education, long overdue, at last becomes feasible." — Paul Bailes, *Science of Computer Programming*, July 1989.
>
> "Every student and teacher of computer science should buy – beg, borrow, or steal – Richard Bird and Philip Wadler's *Introduction to Functional Programming.* Not only is it the best introduction to functional programming published, but it is also an excellent introduction to programming in general." — Richard Jones, *Times Higher Education Supplement*, 17 February 1989.

8.2 Chapters in books

Types in XQuery. Introduction to the XQuery Formal Semantics. Mary Fernández, Jérôme Siméon, and Philip Wadler. Two chapters in Howard Katz, editor, *XQuery for Experts*, Addison-Wesley, 2003.

A prettier printer. Philip Wadler. Chapter in Jeremy Gibbons and Oege de Moor, editors, *The Fun of Programming*, Essays in Honor of Richard Bird, Palgrave, 2003.

XQuery, a typed functional language for querying XML Philip Wadler. Chapter in Johan Juring, editor, *Advanced Functional Programming*, Springer-Verlag, 2003.

How enterprises use functional languages, and why they don't. Philip Wadler. Chapter in *The Logic Programming Paradigm: A 25 Year Perspective*, Shakertown, April 1998. Springer-Verlag. (This paper combines the February and August 1998 columns for Sigplan Notices.)

Monads for functional programming. Chapter in *Advanced Functional Programming*, proceedings of the International Spring School at Båstad, directed by J. Jeuring and E. Meijer. Springer Verlag, LNCS 925, 1995. (This is a slightly revised version of the following paper.)

Monads for functional programming. Chapter in *Program Design Calculi*, proceedings of the International Summer School at Marktoberdorf directed by F. L. Bauer, M. Broy, E. W. Dijkstra, D. Gries, and C. A. R. Hoare. M. Broy, editor. Springer Verlag, NATO ASI series, Series F: Computer and System Sciences, Volume 118, 1994.

Strictness analysis on non-flat domains (by abstract interpretation over finite domains). Chapter in *Abstract Interpretation*, Samson Abramsky and Chris Hankin, editors. Ellis Horwood, 1987.

Structured types and the semantics of pattern-matching (with S.L. Peyton Jones); Efficient compilation of pattern-matching; List comprehensions. Three chapters in *The Implementation of Functional Programming Languages*, S.L. Peyton Jones. Prentice-Hall International Series in Computer Science, C.A.R. Hoare, series editor, 1987.

8.3 Book editor

P. Wadler. *Principles of Programming Languages*, ACM, 2008.

P. Wadler and V. Dahl, editors. *Practical Appications of Declarative Languages*, LNCS 2562, Springer Verlag, 2003.

P. Wadler, *International Conference on Functional Programming*, ACM, 2000.

R. Heldal, C. Kehler Holst, and P. Wadler, editors. *Functional Programming*, Glasgow 1991. Workshops in Computing Series, Springer Verlag, 1992.

8.4 Invited papers

Links: Web programming without tiers. Ezra Cooper, Sam Lindley, Philip Wadler, and Jeremy Yallop. *Formal Methods for Components and Objects (FMCO)*, Amsterdam, November 2006. Springer Verlag LNCS 4709.

The Girard-Reynolds Isomorphism (second edition). Philip Wadler. *Theoretical Computer Science*, Special issue in honor of John Reynolds, 375(1–3):201–226, May 2007.

Call-by-name is dual to call-by-value, reloaded. Philip Wadler *Rewriting Techniques and Applications (RTA)*, Nara, April 2005. (Invited paper.)

The Essence of XML (preliminary version). Jérôme Siméon and Philip Wadler. *Symposium on Functional and Logic Programming (FLOPS)*, Aizu, September 2002. LNCS 2441, Springer Verlag. (Invited paper.)

A semistructured monad for semistructured data. Mary Fernández, Jérôme Siméon, Philip Wadler. *International Conference on Database Theory*, London, January 2001. LNCS 1973, Springer Verlag. (Invited paper.)

An Algebra for XML Query. Mary Fernández, Jérôme Siméon, Philip Wadler. *Foundations of Software Technology and Theoretical Computer Science*, New Delhi, December 2000. LNCS 1974, Springer Verlag. (Invited paper.)

Leftover Curry and reheated Pizza: How functional programming nourishes software reuse. Martin Odersky and Philip Wadler. *IEEE Fifth International Conference on Software Reuse*, Vancouver, BC, June 1998. (Keynote address.)

How to declare an imperative. *International Logic Programming Symposium*, Portland, Oregon, December 1995. MIT Press. (Invited paper.)

A taste of linear logic. *Mathematical Foundations of Computer Science*, Gdansk, Poland, August 1993. Springer Verlag, LNCS 711. (Invited paper.)

The essence of functional programming. *19'th ACM Symposium on Principles of Programming Languages*, Albuquerque, New Mexico, January 1992. (Invited paper.)

8.5 Papers in refereed journals

Technical Perspective: Why Didn't I Think of That? *Communications of the ACM*, 59(8), page 92, August 2016.

Propositions as types. *Communications of the ACM*, 58(12), pp 75–84, December 2015.

Propositions as sessions. *Journal of Functional Programming*, special issue of best papers from ICFP 2012, 24(2–3), pp 384–418, March–May 2014.

The arrow calculus. Sam Lindley, Philip Wadler, and Jeremy Yallop. *Journal of Functional Programming*, 20(1): 5169, 2010.

The Girard-Reynolds isomorphism. Philip Wadler. *Information and Computation* 186(2):260–284, 1 November 2003. (Invited submission, best papers from *Theoretical Aspects of Computer Software*, Sendai, October 2001.)

The marriage of effects and monads. Philip Wadler and Peter Thiemann. *ACM Transactions on Computational Logic* 4(1): 1–32, 2003. (Invited submission.)

MSL: A model for W3C XML Schema. Allen Brown, Matthew Fuchs, Jonathan Robie, and Philip Wadler. *Computer Networks* 39(5): 507–521, 2002. (Invited submission, best papers from *World-Wide Web 10*, Hong Kong, May 2001.)

Featherweight Java: A minimal core calculus for Java and GJ. Atsushi Igarashi, Benjamin Pierce, and Philip Wadler. *Transactions on Programming Languages and Systems (TOPLAS)*, 23(3):396-450, May 2001.

A formal semantics of patterns in XSLT. Philip Wadler. *Markup Languages*, 2(2):183–202, Spring 2000.

Operational interpretations of linear logic. David N. Turner and Philip Wadler. *Theoretical Computer Science*, 227(1999), 231–248. (Invited submission to special issue on linear logic.)

A reflection on call-by-value. Amr Sabry and Philip Wadler. *ACM Transactions on Programming Languages and Systems*, 19(6):916–941, November 1997.

How to declare an imperative. *ACM Computing Surveys*, 29(3):240–263, September 1997.

Lazy vs Strict. *ACM Computing Surveys*, 28(2):318–320, June 1996.

Type classes in Haskell. C. V. Hall, K. Hammond, S. L. Peyton Jones, and P. L. Wadler *ACM Transactions on Programming Languages and Systems*, 18(2):109–138, March 1996.

Monads and composable continuations. *Lisp and Symbolic Computation*, 7(1):39–56, January 1994.

Comprehending monads. *Mathematical Structures in Computer Science*, 2:461–493, 1992. (Special issue of selected papers from 6'th Conference on Lisp and Functional Programming.)

Deforestation: transforming programs to eliminate trees. *Theoretical Computer Science*, 73: 231–248, 1990. (Special issue of selected papers from 2'nd ESOP.)

Fixing a space leak with a garbage collector. *Software Practice and Experience*, 17(9):595–608, September 1987.

Analysis of an algorithm for real-time garbage collection. *Communications of the ACM*, 19(9):491–500, September 1976. (Forsythe Award Paper.)

On pairs of non-intersecting faces of cell-complexes. *Proceedings of the American Mathematical Society*, 51(2):438–440, September 1975.

8.6 Papers in refereed conferences

It may help to know the acceptance rate for various conferences. The following table lists conferences for which these data are available.

Conference	Count	Rate
POPL Principles of Programming Languages	10	30/180
FPCA Functional Programming Languages and Computer Architecture	7	30/130
ICFP International Conference on Functional Programming	5	25/85
Lisp and Functional Programming	2	30/160
OOPSLA Object Oriented Programming Systems, Languages, Applications	2	30/150
ESOP European Symposium on Programming	2	25/75
LICS Logic in Computer Science	1	50/150
PEPM Partial Evaluation and Program Manipulation	1	30/70

The second column counts the number of times I have had papers in the listed conference. The third column shows typical figures for number of papers in the conference over number of papers submitted.

Coherence Generalises Duality: a logical explanation of multiparty session types. Marco Carbone, Sam Lindley, Fabrizio Montesi, Carsten Schürmann, Philip Wadler. *International Conference on Concurrency Theory (CONCUR)*, Toronto, August 2016.

Blame and Coercion:Together Again for the First Time. Jeremy Siek, Peter Thiemann, Philip Wadler. *Programming Language Design and Implementation (PLDI)*, Portland, June 2015.

A complement to blame. Philip Wadler. *1st Summit on Advances in Programming Languages (SNAPL)*, Asilomar, May 2015.

Query shredding: efficient relational evaluation of queries over nested multisets James Cheney, Sam Lindley, Philip Wadler. *SIGMOD*, Snowbird, June 2014.

A practical theory of language-integrated query. James Cheney, Sam Lindley, Philip Wadler. *International Conference on Functional Programming (ICFP)*, Boston, September 2013.

Propositions as Sessions. Philip Wadler. *International Conference on Functional Programming (ICFP)*, Copenhagen, September 2012. (Invited for submission to a special issue of *Journal of Functional Programming.*)

Blame for All. Amal Ahmed, Robert Bruce Findler, Jeremy G. Siek, and Philip Wadler. *Principles of Programming Languages (POPL)*, Austin, January 2011.

Threesomes, with and without blame. Jeremy G. Siek and Philip Wadler. *Principles of Programming Languages (POPL)*, Madrid, January 2010.

The RPC Calculus. Ezra Cooper and Philip Wadler. *Principles and Practice of Declarative Programming (PPDP)*, Coimbra, 2009.

Well-typed programs can't be blamed. Philip Wadler and Robert Bruce Findler. *European Symposium on Programming (ESOP)*, York, 2009.

The Essence of form abstraction. Ezra Cooper, Sam Lindley, Philip Wadler, Jeremy Yallop. *Asian Symposium on Programming Languages and Systems*, Bangalore, December 2008.

Comprehensive comprehensions. Simon Peyton Jones and Philip Wadler. *Haskell Symposium*, Berlin, September 2007.

A history of Haskell: being lazy with class Paul Hudak, John Hughes, Simon Peyton Jones, Philip Wadler *History of Programming Languages III (HOPL)*, San Diego, June 2007.

Call-by-value is dual to call-by-name. Philip Wadler. *International Conference on Functional Programming (ICFP)*, Uppsala, September 2003.

The Essence of XML. Jérôme Siméon and Philip Wadler. *ACM Symposium on Principles of Programming Languages*, New Orleans, January 2003.

The Girard-Reynolds isomorphism. Philip Wadler. *Theoretical Aspects of Computer Software*, Sendai, October 2001. LNCS 2215, Springer Verlag.

MSL: A model for W3C XML Schema. Allen Brown, Matthew Fuchs, Jonathan Robie, and Philip Wadler. *World Wide Web 10*, Hong Kong, May 2001.

A formal semantics of patterns in XSLT. Philip Wadler. *Markup Technologies*, Philadelphia, December 1999.

Featherweight Java: A minimal core calculus for Java and GJ. Atsushi Igarishi, Benjamin Pierce, and Philip Wadler. *OOPSLA*, Denver, November 1999.

Making the future safe for the past: Adding Genericity to the Java Programming Language. Gilad Bracha, Martin Odersky, David Stoutamire, and Philip Wadler. *OOPSLA*, Vancouver, October 1998. [This is the GJ overview paper. There is also a GJ tutorial and a GJ specification, available at the web site.]

The marriage of effects and monads. Philip Wadler. *International Conference on Functional Programming (ICFP)*, Baltimore, September 1998.

A practical subtyping system for Erlang. Simon Marlow and Philip Wadler. *International Conference on Functional Programming (ICFP)*, Amsterdam, June 1997.

Pizza into Java: Translating theory into practice. Martin Odersky and Philip Wadler. *Principles of Programming Languages (POPL)*, Paris, January 1997.

Linear logic, monads, and the lambda calculus. Nick Benton and Philip Wadler. *Logic in Computer Science (LICS)*, Trenton, New Jersey, July 1996.

A reflection on call-by-value. Amr Sabry and Philip Wadler. *International Conference on Functional Programming (ICFP)*, Philadelphia, Pennsylvania, May 1996.

Once upon a type. David N. Turner, Philip Wadler, and Christian Mossin. *International Conference on Functional Programming and Computer Architecture (FPCA)*, San Diego, California, June 1995.

A second look at overloading. Martin Odersky, Philip Wadler, and Martin Wehr. *International*

Conference on Functional Programming and Computer Architecture (FPCA), San Diego, California, June 1995.

Call-by-name, call-by-value, call-by-need, and the linear lambda calculus. John Maraist, Martin Odersky, David N. Turner, and Philip Wadler. *11'th International Conference on the Mathematical Foundations of Programming Semantics*, New Orleans, Lousiana, March–April 1995.

A call-by-need lambda calculus. Zena Ariola, Matthias Felleisen, John Maraist, Martin Odersky, and Philip Wadler. *Principles of Programming Languages (POPL)*, San Francisco, California, January 1995.

A syntax for linear logic. *Mathematical Foundations of Programming Semantics (MFPS)*, New Orleans, Lousiana, April 1993.

Imperative functional programming. Simon Peyton Jones and Philip Wadler. *Principles of Programming Languages (POPL)*, Charlotte, North Carolina, January 1993.

Is there a use for linear logic? *Symposium on Partial Evaluation and Semantics-Based Program Manipulation (PEPM)*, New Haven, Connecticut, June 1991.

Comprehending monads. *Lisp and Functional Programming*, Nice, France, pp. 61–78, June 1990.

Linear types can change the world! *IFIP TC 2 Working Conference on Programming Concepts and Methods*, Sea of Galilee, Israel, April 1990. Published as M. Broy and C. Jones, editors, *Programming Concepts and Methods*, North Holland, 1990.

Improving list comprehension database queries. Phil Trinder and Philip Wadler. *TENCON '89* (IEEE Region 10 Conference), Bombay, India, November 1989.

Theorems for free! *Functional Programming Languages and Computer Architecture (FPCA)*, London, September 1989.

How to make *ad-hoc* polymorphism less *ad hoc*. Philip Wadler and Stephen Blott. *Principles of Programming Languages (POPL)*, Austin, Texas, January 1989.

Deforestation: transforming programs to eliminate trees. *European Symposium on Programming (ESOP)*, Nancy, France, March 1988.

Strictness analysis aids time analysis. *Principles of Programming Languages (POPL)*, San Diego, California, January 1988.

Projections for strictness analysis. Philip Wadler and R.J.M. Hughes. *Conference on Functional Programming Languages and Computer Architecture (FPCA)*, Portland, Oregon, September 1987.

Views: a way for pattern matching to cohabit with data abstraction. *Principles of Programming Languages (POPL)*, Munich, FRG, January 1987.

How to replace failure by a list of successes. *Functional Programming Languages and Computer Architecture (FPCA)*, Nancy, France, September 1985. Lecture Notes in Computer Science 201, Springer-Verlag, 1985.

Listlessness is better than laziness: lazy evaluation and garbage collection at compile-time. *Lisp and Functional Programming*, Austin, Texas, 1984.

Applicative style programming, program transformation, and list operators. *Functional Programming Languages and Computer Architecture (FPCA)*, Wentworth-by-the-Sea, New Hampshire, October 1981.

Experience with an applicative string processing language. James Morris, Eric Schmidt, and Philip Wadler. *Principles of Programming Languages (POPL)*, Las Vegas, Nevada, January 1980.

Transforming offices to be highly parallel. J. Barth, C. Ellis, and P. Wadler. *1'st European Conference on Parallel and Distributed Computing*, Tolouse, France, 1979.

8.7 Papers in workshops

Special Interest Group: Usability of Programming Languages (extended abstract). B. A. Myers, A. Stefik, S. Hanenberg, A-J. Kaijanaho, M. Burnett, F. Turbak, P. Wadler. *Computer Human Interaction (CHI)*, San Jose, May 2016.

Everything old is new again: Quoted Domain Specific Languages. Shayan Najd, Sam Lindley, Josef Svenningsson, Philip Wadler. *Partial Evaluaration and Program Manipulation (PEPM)*, St. Petersberg, Florida, January 2016.

Typecasting Actors: From Akka to TAkka. Jiansen He, Philip Wadler, Phil Trinder. *Scala Workshop*, Uppsala, June 2014.

Blame for all. Amal Ahmed, Robert Bruce Findler, Jacob Matthews, and Philip Wadler. Workshop on Script to Program Evolution (STOP), Genova, 2009.

Threesomes, With and Without Blame. Jeremy G. Siek and Philip Wadler. Workshop on Script to Program Evolution (STOP), Genova, 2009.

Idioms are oblivious, arrows are meticulous, monads are promiscuous. Sam Lindley, Philip Wadler, Jeremy Yallop. *Mathematically Structured Functional Programming*, Reyjavik, July 2008.

Well-typed programs can't be blamed. Philip Wadler and Robert Bruce Findler. *Scheme Workshop*, Berlin, September 2007.

A recipe for raw types. Atsushi Igarashi, Benjamin Pierce, and Philip Wadler. *7'th Workshop on Foundations of Object-Oriented Languages (FOOL)*, London, January 2001.

How to add laziness to a strict language, without even being odd. Philip Wadler, Walid Taha, and David MacQueen. *Workshop on Standard ML*, Baltimore, September 1998.

A statically safe alternative to virtual types. Kim B. Bruce, Martin Odersky, and Philip Wadler. *European Conference on Object-Oriented Programming*, July 1998. (An earlier version was presented at *5'th Workshop on Foundations of Object-Oriented Languages*, San Diego, January 1998.)

Combining monads. David King and Philip Wadler. Proceedings of the 1992 Glasgow Workshop on Functional Programming, Ayr, Scotland (July 1992), 134–143. Springer Verlag Workshops in Computing Series.

The Glasgow Haskell compiler: a retrospective. C. Hall, K. Hammond, W. Partain, S. L. Peyton Jones, and P. Wadler. Proceedings of the 1992 Glasgow Workshop on Functional Programming, Ayr, Scotland (July 1992), 134–143. Springer Verlag Workshops in Computing Series.

Avoiding unnecessary updates. J. Launchbury, A. Gill, J. Hughes, S. Marlow, S. L. Peyton Jones, and P. Wadler. Proceedings of the 1992 Glasgow Workshop on Functional Programming, Ayr, Scotland (July 1992), 134–143. Springer Verlag Workshops in Computing Series.

There's no substitute for linear logic. Workshop on Mathematical Foundations of Programming Semantics, Oxford, April 1992 (no proceedings).

Strictness analysis in 4D. Kei Davis and Philip Wadler. Proceedings of the 1990 Glasgow Workshop on Functional Programming, Ullapool, Scotland (August 1990), 23–43. Springer Verlag Workshops in Computing Series.

Backwards strictness analysis: proved and improved. Kei Davis and Philip Wadler. Proceedings of the 1989 Glasgow Workshop on Functional Programming, Fraserburgh, Scotland (August 1989), 12–30. Springer Verlag Workshops in Computing Series.

List comprehensions and the relational calculus. Phil Trinder and Philip Wadler. Proceedings of the 1988 Glasgow Workshop on Functional Programming, Rothesay, Scotland (August 1988), 115–123.

A new array operation. *Workshop on Graph Reduction*, Santa Fe, New Mexico, October 1986. LNCS 279, Springer-Verlag 1987.

Listlessness is better than laziness II: composing listless functions. *Workshop on Programs as Data Objects*, Copenhagen, Denmark, October 1985. LNCS 217, Springer-Verlag, 1985.

Views: a way for elegant definitions and efficient implementations to co-exist. *Workshop on Implementation of Functional Languages*, Aspenäs, Sweden, February 1985. Report 17, Programming Methodology Group, Chalmers University, Göteborg, Sweden.

8.8 Patents

Joanne J. Ordille, Thomas A. Petsche, and Philip L. Wadler. Method and Apparatus for Automatic Notification and Response Based on Communication Flow Expressions. US Patent no.: 7,436,947 B2. Date of Patent: October 14, 2008.

8.9 XML standards

XQuery 1.0: An XML Query Language, W3C Working Draft, 15 November 2002.
`http://www.w3.org/TR/xquery/`

XQuery 1.0 and XPath 2.0 Formal Semantics, W3C Working Draft, 15 November 2002 [editor].
`http://www.w3.org/TR/query-semantics/`

XML Path Language (XPath) 2.0, W3C Working Draft, 15 November 2002.
`http://www.w3.org/TR/xpath20/`

XQuery 1.0 and XPath 2.0 Functions and Operators, W3C Working Draft, 15 November 2002.
`http://www.w3.org/TR/xquery-operators/`

XML Schema: Formal Description, W3C Working Draft, 25 September 2001 [editor].
`http://www.w3.org/TR/xmlschema-formal/`

XSL Transformations (XSLT), Version 1.0, W3C Recommendation 16 November 1999.
`http://www.w3.org/TR/xslt`

XML Path Language (XPath), Version 1.0, W3C Recommendation 16 November 1999.
`http://www.w3.org/TR/xpath`

8.10 Haskell reports

[I am a co-author of all reports, and co-editor of the earliest three.]

Haskell 98 Languages and Libraries, The Revised Report. Simon Peyton Jones, editor. Cambridge University Press, 2003.

Haskell 98: A Non-strict, Purely Functional Language. Simon Peyton Jones and John Hughes, editors. www.haskell.org, February 1999.

Report on the programming language Haskell, a non-strict purely-functional programming language, Version 1.3. John Peterson and Kevin Hammond, editors. Technical report, Yale University, May 1996.

Report on the programming language Haskell, a non-strict purely-functional programming language, Version 1.2. Paul Hudak, S. L. Peyton Jones, and P. L. Wadler, editors. *Sigplan Notices* 27(5), May 1992.

Report on the programming language Haskell, a non-strict purely-functional programming language, Version 1.1. Paul Hudak, S. L. Peyton Jones, and P. L. Wadler, editors. Technical Report, University of Glasgow and Yale University, August 1991.

Report on the programming language Haskell, a non-strict purely-functional programming language. Paul Hudak and Philip Wadler, editors. Technical Report, University of Glasgow and Yale University, January 1989.

8.11 Articles

GJ: A generic Java. *Dr. Dobb's Journal* 25(2):23–28, February 2000.

New Languages, Old Logic. *Dr. Dobb's Journal* 25(13):37–41, December 2000 (special supplement).

8.12 Columns

Functional Programming: Why no one uses functional languages. Philip Wadler. *Sigplan Notices* 33(8):23–27, August 1998.

Functional Programming: An angry half dozen. Philip Wadler. *Sigplan Notices* 33(2):25–30, February 1998.

8.13 Book reviews

'Superpowerful computing', *New Scientist*, 14 June 1997. Reviews *The Supermen* by Charles Murray, Wiley.

'In loving memory of humankind', *New Scientist*, 5 October 1996. Reviews *Moths to the Flame* by Gregory Rawlins, MIT Press.

'Banishing errors', *Times Higher Education Supplement*, 21 February 1992. Reviews *ML for the Working Programmer* by Paulson, *The Definition of Standard ML* by Milner, Tofte, and Harper, *Commentary on Standard ML* by Milner and Tofte, and *Applicative High Order Programming: The Standard ML Perspective* by Sokolowski.

8.14 Other publications

A critique of Ablelson and Sussmann, or, Why calculating is better than scheming. *Sigplan Notices* 22(3), March 1987.

An Introduction to Orwell. Programming Research Group, Oxford University, April 1985.

A simple language is also a functional language (short communication). *Software Practice and Experience*, 15(2), February 1985.

Listlessness is Better than Laziness. Ph.D. dissertation, Carnegie-Mellon University. Technical report CMU-CS-85-171. August 1984.

8.15 Videos

Propositions as Types Lambda Days, Krakow, 2016. [2K views.]
`https://www.youtube.com/watch?v=aeRVdYN6fE8`

Propositions as Types. Strange Loop, St Louis, 2015. [28K views.]
`https://www.youtube.com/watch?v=IOiZatlZtGU`

Quoted Domain Specific Languages. Strange Loop, St Louis, 2015. [7K views.]
`https://www.youtube.com/watch?v=DlBwJ4rvz5c`

Quoted Domain Specific Languages. Curry On, Prague, 2015. [6K views.]
`https://www.youtube.com/watch?v=FiflFiZ6pPI`

Computability (stand up, 7 minutes). Bright Club, The Stand, Edinburgh, 28 April 2015. [7K views.]
`https://www.youtube.com/watch?v=GnpcMCWORUA`

Church's Coincidences. Turing Centennial, Princeton, 2012. [6K views.]
`https://www.youtube.com/watch?v=2PJ_DbKGFUA`

Faith, Evolution, and Programming Languages: from Haskell to Java. Tech Mesh, London, 2012. [2K views.]
`https://www.youtube.com/watch?v=NZeDRs6snm0`

9 Presentations

9.1 Invited talks

The Third International Conference on Parallel, Distributed and Grid Computing (PDGC), Himachal Pradesh, India, 11-13 December 2014 (keynote).

Functional Programming eXchange, London, 14 March 2014 (invited speaker).

YOW! Australia Developers Conference, Melbourne 5-6, Brisbane 9-10, Sydney 12–13 December 2013, (keynote).

FP Days, Cambridge, 24 October 2013 (keynote).

Colin Runciman Celebration, York, 23 October 2013 (keynote).

Functional Programming eXchange, London, 14 March 2014, invited speaker. ESOP 2014, 711 April 2014, program committee.

International Summer School on Trends in Computing (SSTC 2013), Tarragona, 22-26 July 2013 (invited lecturer).

Scala Workshop, Montpellier, 2 July 2013 (keynote).

Midlands Graduate School in the Foundations of Computing Science, 8-12 April 2013 (invited lecturer).

Data-Driven Functional Programming (DDFP), workshop affiliated with POPL, Rome, 22 January 2013 (keynote).

Tech Mesh, London, 4-6 December 2012 (keynote).

SICSA, Glasgow, 10–22 June 2012 (keynote).

Turing Centennial Celebration, Princeton, 10–12 May 2012 (invited speaker).

QCON, London, 7–9 March 2012 (invited talk).

DBPL, Seattle, 29 August 2011 (invited speaker).

ETAPS, Paphos, Cyprus, 20–28 March 2010 (invited speaker).

AOSD, Brussels, 2–4 April 2008 (keynote).

FMCO, Amsterdam, 7–10 November 2006 (invited talk).

OOPSLA, Portland, 22–6 October 2006 (invited talk).

Intel, Berkeley, 30 April 2007 (distinguished lecture series).

NICTA, Melbourne, 2 February 2006 (distinguished lecture series).

PADL, Charleston, January 2006 (invited speaker).

MFPS, Birmingham, 18 May 2005 (Reynolds special session, invited talk).

University of York, 11 May 2005 (distinguished lecture series).

University of Oxford, 26 April 2005 (Strachey lecture).

RTA, Nara, 19–21 April 2005 (invited talk).

St Andrews, 21 March 2005 (distinguished lecture series).

FGUC, London (colocated with CONCUR), 3-4 September 2004 (invited speaker).

APPSEM, Talinn, 14-16 April 2004 (invited speaker).

ETH Zurich, 17 November 2003 (distinguished lecture series).

Bard College, 10 April 2003 (distinguished lecture series).

University of Utah, 20 November 2002 (distinguished lecture series).

Erlang Workshop (attached to ICFP), Pittsburgh, 7 October 2002 (invited talk).

FLOPS, Aizu, 15-17 September 2002 (invited talk).

Summer School on Advanced Functional Programming, Oxford, 19-24 August 2002 (invited lecturer).

XML 2001, XQuery Tutorial, Walt Disney World, Orlando, 10 December 2001 (tutorial).

University of Pennsylvania, 20 November 2001 (distinguished lecture series).

Columbia University, 12 November 2001 (distinguished lecture series).

Very Large Databases (VLDB), Rome, 11–14 September 2001 (keynote).

International Conference on Database Theory (ICDT), London, 3-5 January 2001 (invited talk).

Foundations of Software Technology and Theoretical Computer Science (FST TCS), New Delhi, 13-15 December 2000 (invited talk).

Alan J. Perlis Symposium, "Programming Languages: Theory Meets the Real World", Yale University, 27 April 2000 (invited talk).

"Data Processing on the Web: a Look into the Future", DIMACS workshop, New Brunswick, 6–7 March 2000 (invited talk).

LXNY, New York Free Software Organization, 4 January 2000 (invited talk).

2'nd USENIX Conference on Domain Specific Languages, Austin, Texas, 3–5 October 1999 (invited talk).

8'th International Conference on Compiler Construction, (part of ETAPS), Amsterdam, 22–26 March 1999 (invited talk). [Talk delivered by Gilad Bracha, while I awaited arrival of my twins.]

New York PC club, Java and C++ special interest group, 3 December 1998, New York.

Programming Language Design and Implementation, 16 June 1998, Montreal (invited tutorial).

5'th International Conference on Software Reuse, 2–5 June 1998, Victoria (keynote).

Foundational Methods in Computer Science, Portland, May 1998 (invited talk).

The Logic Programming Paradigm: Current trends and future directions, Shakertown, April 1998 (invited talk).

Williams College, November 1997 (distinguished lecture series).

Languages and Models with Objects, 22–24 October 1997, Brest (invited talk).

First Workshop on Functional Programming, 11 September 1996, Buenos Aires, Argentina (invited talk).

Spring School on Functional Programming, Laboratoria de Investigacion e Formacion en Informatica Avanada, La Plata, Argentina, 9–12 September 1996 (invited lecturer).

First Brazilian Symposium On Programming Languages, 4–6 September 1996, Belo Horizonte, Brazil (invited talk).

International Logic Programming Symposium, Portland, Oregon, 4–7 December 1995 (invited talk).

Spring School on Advanced Functional Programming, Båstad, Sweden, 24–31 May 1995 (invited lecturer).

Computing, the Australian Theory Seminar (CATS), Sydney, Australia, December 1994 (invited talk).

Special seminar series on programming languages, CWI Amsterdam, 5 March 1994 (invited talk).

International Symposium on Mathematical Foundations of Computer Science, Gdańsk, Poland, 30 August–3 September 1993 (invited talk).

Marktoberdorf Summer School on Program Design Calculi, Broy, Dijkstra, Gries, and Hoare, directors, 28 July–9 August 1992 (invited lecturer).

19'th Symposium on Principles of Programming Languages, Albuquerque, New Mexico, 20–22 January 1992 (invited talk).

5'th International Conference on Functional Programming Languages and Computer Architecture, Boston, 27 August 1991 (invited tutorial).

Workshop on Functional Programming Languages, Edinburgh University, 12 March 1990 (invited tutorial).

4'th International Conference on Functional Programming Languages and Computer Architecture, London, 11–13 September 1989 (invited tutorial).

Weekend School on Lambda Calculus, SERC Logic for IT Initiative, University of Leeds, December 1988 and January 1989 (invited lecturer).

9.2 Workshops

Foundations of Scripting Languages, Dagstuhl, 2–6 January 2012.

STOP, Austin, 29 January 2011 (steering committee).

STOP, Genova, 6 July 2009 (program committee).

Programming Paradigms for the Web, Dagstuhl, 28 January–2 February 2007 (co-organizer).

Links Workshop, Edinburgh, 6 April 2005 (co-organizer).

Avaya Software Symposium, Basking Ridge, NJ, 22–25 October 2002.

Avaya Software Symposium, Westminster, CO, 27 November 2001.

Lucent Technologies Software Symposium, Holmdel, NJ, 29–30 October 1997

Workshop on High-Level Concurrent Languages, Dagstuhl, Germany, 20–22 January 1997 (co-organizer).

2'nd Erlang Users Group Meeting, 11–12 October 1995.

Workshop on Functional Programming in the Real World, Dagstuhl, Germany, 16–20 May 1994.

Categorical Logic in Computer Science Workshop, Aarhus University, Aarhus, Denmark, 23–27 March 1992.

Workshop on Implementation of Lazy Functional Languages, Aspenäs, Sweden, 5–8 September 1988.

Workshop on Graph Reduction, Santa Fe, New Mexico, 29 September–1 October 1986.

Workshop on Functional and Logic Programming, University of Stirling, 25 April 1986.

Workshop on Accidental Nuclear War, Manchester, 9 November 1985.

Workshop on Programs as Data Objects, Copenhagen, Denmark, 17–19 October 1985.

Workshop on Abstract Interpretation, University of Kent, 20–23 August 1985.

Workshop on Software Tools, Hull, 11–12 July 1985.

Workshop on Implementing Functional Languages, Aspenäs, Sweden, 4–6 February 1985.

9.3 Working groups

IFIP 2.8 Working group on functional programming
Boston, 17–21 July 2006.
Las Vegas, 3–8 March 2002.
Warm Springs, Oregon, 30 March–3 April 1998.
Mohonk, New York, 22–27 September 1996.
Roligned, Denmark, 6–8 June 1993.
Oxford, 3–4 September 1992.
Santa Fe, 14–17 January 1992.
Paris, 16–19 April 1991.
Rome, 26–30 March 1990.
Mystic, Connecticut, 1–5 May 1989.
Glasgow, 11–15 July 1988.

W3C XQuery, XML Schema Formalism, and XSLT Working Groups
XQuery, San Jose, CA, 17–18 December 2002.
XQuery, Chapel Hill, NC, 17–18 October 2002.
XQuery, Redwood Shores, CA, 16–17 May 2002.
XQuery, Oakland, CA, 6–7 December 2001.
XQuery, Reston, VA, 2–3 October 2001.
XQuery, Fairfax, VA, 17–18 May 2001.
XQuery, San Jose, CA, 22–23 March 2001.
XQuery, Boston, MA, 24–26 February 2001.
XQuery, Menlo Park, CA, 15–16 November 2000.
Schema, Vancouver, Canada, 28 September 2000.
XQuery, Yorktown Heights, NY, 21–22 September 2000.
Schema, Cupertino, CA, 29 August 2000.
XQuery, Redmond, WA, 25–26 July 2000.
XQuery, Murray Hill, NJ, 5–7 June 2000.

XQuery, Berkeley, CA, 7–8 February 2000.
XQuery, Florham Park, NJ, 2–3 December 1999.
XQuery, San Jose, CA, 15–16 September 1999.
XSLT, New York, NY, 19–21 July 1999.

Haskell design committee meetings
Glasgow, 6–9 July 1988.
Glasgow, 6–9 April 1988.
New Haven, Connecticut, 9–11 January 1988.

10 Other Interests

I have a keen interest in educating the public on issues relating to the use and misuse of computers. I have given public talks on the role of computers in the control of nuclear weapons in Glasgow, Oxford, Manchester, Göteborg, Gdansk, and Warsaw. Former students have twice invited me to talk on computing and society (once as an after-dinner speaker at a Christmas meal).

In my early career, I regularly provided interviews for the media on computer-related stories.

> News on the Internet, BBC Radio Scotland, 'Eye-to-eye', October 1995.
> Pornography on the Internet, BBC Radio Scotland, morning news, July 1995.
> The Internet, Channel 4 Television, 'Science and Society', May 1993.
> Computer viruses, BBC Radio Scotland, morning news, August 1993.
> Computer break-ins, BBC Radio Scotland, morning news, November 1992.

I maintain an interest in public communication of science, and continue to participate regularly in related events.

> Sock the Scientist, Edinburgh Science Festival, 11 April 2010.
> I'm a Scientist Get me out of here, July 2010.

I am an amateur cartoonist. My published artwork includes the emblem for the International Conference on Functional Programming Languages and Computer Architecture, illustrations for articles on computing in the Co-Evolution Quarterly, publications at Glasgow and Carnegie-Mellon Universities, the cover of the Scribe manual (published by Unilogic), and three designs for T-shirts.

My other interests include squash, hill walking, collecting comic books, and calligraphy.

Contents