

John Hughes' Publication List

This is a list of my publications as of March 2009, with citation counts fetched from Google Scholar at that time. My most highly cited articles are [57, 15, 23, 64, 27, 38, 75, 71, 70, 72, 40]; all of these have over 100 citations according to Google Scholar.

References

- [1] John Hughes, *Experiences from teaching functional programming at Chalmers*, ACM SIGPLAN Notices , Volume 43 Issue 11, November 2008. **(0 citations)**.
- [2] Thomas Arts, Laura M. Castro, John Hughes. *Testing Erlang data types with quviq quickcheck*. ERLANG '08: Proceedings of the 7th ACM SIGPLAN workshop on ERLANG, Victoria, BC, Canada. September 2008. **(0 citations)**.
- [3] Alejandro Russo, Koen Claessen, and John Hughes. *A Library for Light-Weight Information-Flow Security in Haskell*. Haskell '08: Proceedings of the first ACM SIGPLAN symposium on Haskell, Victoria, BC, Canada, September 2008. **(2 citations)**.
- [4] Paul Hudak, John Hughes, Simon Peyton Jones, and Philip Wadler. A History of Haskell: being lazy with class. In *The Third ACM SIGPLAN History of Programming Languages Conference (HOPL-III)*, San Diego, California, June 2007. **(30 citations)**.
- [5] Ta-Chung Tsai, Alejandro Russo, and John Hughes. A Library for Secure Multi-threaded Information Flow in Haskell. In *Proceedings of the 20th IEEE Computer Security Foundations Symposium*, Venice, Italy, July 6-8, 2007. IEEE Computer Society Press. **(5 citations)**.
- [6] John Hughes. Functional Programming: a Secret Weapon for Software Testing (invited talk). *12th ACM SIGPLAN International Conference on Functional Programming*, ed. Norman Ramsey. Freiburg, Germany, October 2007. Available on Google Video. **Citation count missing!**
- [7] John Hughes. QuickCheck Testing for Fun and Profit (invited paper). In *9th International Symposium on Practical Aspects of Declarative Languages*, ed. Michael Hanus. Nice, France. January 2007. Lecture Notes in Computer Science vol. 4354, Springer. **(4 citations)**.
- [8] Alejandro Russo, Andrei Sabelfeld, John Hughes, and David Naumann. Closing Internal Timing Channels by Transformation. In *Proceedings of the 11th Annual Asian Computing Science Conference*, Tokyo, Japan, December 6-8, 2006. **(10 citations)**.

- [9] Thomas Arts, John Hughes, Joakim Johansson, and Ulf Wiger. Testing Telecoms Software with Quviq QuickCheck. In *Fifth ACM SIGPLAN Erlang Workshop*, Portland, Oregon, September 2006. **(12 citations)**.
- [10] Nils Anders Danielsson, Jeremy Gibbons, John Hughes, and Patrik Jansson. Fast and loose reasoning is morally correct. In *Proceedings of the 33rd ACM SIGPLAN-SIGACT symposium on Principles of programming languages*, 2006. **(22 citations)**.
- [11] Thomas Arts, Koen Claessen, John Hughes, and Hans Svensson. Testing Implementations of Formally Verified Algorithms. In SERPS, Västerås, October 2005. **(4 citations)**.
- [12] Andreas Abel, Marcin Benke, Ana Bove, John Hughes, and Ulf Norell. Verifying Haskell programs using constructive type theory. In *ACM SIGPLAN Workshop Haskell'05*, Tallinn, Estonia, 30 September, 2005. ACM Press, 2005. **(11 citations)**.
- [13] J. Hughes, *Global Variables in Haskell*, Journal of Functional Programming, Volume 14, Issue 05, September 2004, pp 489-502 **(9 citations)**.
- [14] John Hughes, *Programming with Arrows*, Fifth International Summer School on Advanced Functional Programming, Tartu, Estonia, August 2004. Editors: Varmo Vene and Tarmo Uustalu. Springer LNCS Vol. 3622. **(6 citations)**.
- [15] Simon Peyton-Jones et al. Haskell 98 Language and Libraries. The Revised Report. Cambridge University Press, 2003. **(610 citations)**.
- [16] John Hughes, Per Larsson Edefors, Mary Sheeran, Per Stenström and Lars Svensson, *FlexSoC: Combining Flexibility and Efficiency in SoC Designs*, Norchip, Riga, November 2003. **(4 citations)**.
- [17] John Hughes and Doaitse Swierstra, *Polish parsers, step by step (functional pearl)*, ACM International Conference on Functional Programming, Uppsala, Sverige, August 2003. **(8 citations)**.
- [18] K. Claessen and J. Hughes, *Specification based testing with QuickCheck*, chapter in *The Fun of Programming* (festschrift for Richard Bird), eds. Jeremy Gibbons and Oege de Moor, Palgrave “Cornerstones of Computing” series, 2003. **(9 citations)**.
- [19] Pablo E. Martinez Lpez and John Hughes, *Principal Type Specialisation*, ACM SIGPLAN Asian Symposium on Partial Evaluation and Semantics-based Program Manipulation, Aizu, Japan, September 2002. **(2 citations)**.
- [20] Koen Claessen, Colin Runciman, Olaf Chitil, John Hughes and Malcolm Wallace, *Testing and Tracing Lazy Functional Programs Using QuickCheck and Hat*, in *Advanced Functional Programming: 4th International School*, eds. Johan Jeuring and Simon Peyton Jones, Oxford, Springer LNCS vol. 2638, 2002. **(19 citations)**.

- [21] K. Claessen and J. Hughes, *Testing Monadic Code with QuickCheck*, ACM SIGPLAN Haskell Workshop (part of ACM *Principles, Logics, and Implementations of High-Level Programming Languages*), ed. M. Chakravarty, Pittsburgh, 2002. **(25 citations)**.
- [22] R. Heldal, J. Hughes. *Binding-time Analysis for Polymorphic Types*. In Dines Bjorner, Manfred Broy, and Alexandre Zamulin (editors), Andrei Ershov Fourth International Conference on Perspectives of System Informatics, Novosibirsk, 2001. Springer-Verlag LNCS vol. 2224. **(4 citations)**.
- [23] Koen Claessen and John Hughes. *QuickCheck, A Lightweight Tool for Random Testing of Haskell Programs*. In Phil Wadler (editor), International Conference on Functional Programming, Montreal, Canada; 18–20 September 2000. **(252 citations)**.
- [24] Walid Taha, Henning Makholm, and John Hughes. *Tag Elimination and Jones-Optimality*. In Olivier Danvy and Andrzej Filinski (editors), Second Symposium on Programs as Data Objects PADO II, Aarhus, Denmark. Springer Verlag, LNCS, vol. 2053, 2001. **(37 citations)**.
- [25] Nick Benton, John Hughes, and Eugenio Moggi. *Monads and Effects*. APPSEM Summer School on Applied Semantics, Caminha, Portugal. September 2000. Springer-Verlag LNCS Volume 2395, pp42-122. **(64 citations)**.
- [26] J. Hughes. *The Correctness of Type Specialisation*. In Gert Smolka, editor, *European Symposium on Programming*, Lecture Notes in Computer Science. Springer-Verlag, 2000. **(9 citations)**.
- [27] J. Hughes, *Generalising Monads to Arrows*, *Science of Computer Programming*, 37 (1-3) (2000) pp. 67-111. **(183 citations)**.
- [28] R. Heldal, J. Hughes *Extending a Partial Evaluator which Supports Separate Compilation*, *Theoretical Computer Science*, 248 (1-2) (2000) pp. 99-145. **(4 citations)**.
- [29] J. Hughes, *Restricted Data Types in Haskell*, in Proceedings of the 3rd Haskell Workshop, available as a Utrecht University technical report, October, 1999. **(29 citations)**.
- [30] J. Hughes and L. Pareto, *Recursion and Dynamic Data-Structures in bounded space; Towards Embedded ML Programming*, ACM SIGPLAN International Conference on Functional Programming, September, 1999. **(92 citations)**.
- [31] J. Hughes, *Type Specialisation*, in *1998 Symposium on Partial Evaluation*, editors Olivier Danvy, Robert Glück, and Peter Thiemann, vol. 30 of *Computing Surveys*, September, 1998. **(12 citations)**.

- [32] J. Hughes, *A Type Specialisation Tutorial*, in DIKU Summer School on Partial Evaluation, 1998. **(8 citations)**.
- [33] D. Dussart, R. Heldal, J. Hughes, Module-Sensitive Program Specialisation, ACM SIGPLAN Conference on Programming Language Design and Implementation, Las Vegas, June, 1997. **(14 citations)**.
- [34] R. Heldal, J. Hughes, *Partial Evaluation and Separate Compilation*, ACM SIGPLAN Symposium on Partial Evaluation and Semantics-Based Program Manipulation, Amsterdam, June, 1997. **(7 citations)**.
- [35] D. Dussart, J. Hughes, P. Thiemann, *Type Specialisation for Imperative Languages*, ACM International Conference on Functional Programming, Amsterdam, June, 1997. **(25 citations)**.
- [36] J. Hughes, *An Introduction to Program Specialisation by Type Inference*, Glasgow Workshop on Functional Programming, July, 1996. Published electronically as <http://www.dcs.gla.ac.uk/fp/workshops/fpw96/Proceedings96.html>. **(14 citations)**.
- [37] J. Hughes, *Type Specialisation for the Lambda-Calculus*, Dagstuhl Workshop on Partial Evaluation, February, 1996. Springer LNCS Vol. 1110. **(64 citations)**.
- [38] J. Hughes, L. Pareto and A. Sabry, *Proving the Correctness of Reactive Systems using Sized Types*, ACM POPL, 1996. **(178 citations)**.
- [39] J. Hughes and J. Sparud, *Haskell++: an Object-Oriented Extension of Haskell*, Proc. the Haskell Workshop, Yale University Technical Report, 1995. **(23 citations)**.
- [40] J. Hughes, *The Design of a Pretty-printing Library*, Proc. Spring School on Functional Programming, Springer-Verlag LNCS Vol. 925, May 1995. **(111 citations)**.
- [41] J. Hughes and A. Moran, *Making Choices Lazily*, Proc. ACM Conference on Functional Programming and Computer Architecture, June 1995. **(22 citations)**.
- [42] John Hughes and John Launchbury, *Reversing Abstract Interpretations*, Science of Computer Programming, Vol. 22, No. 3, June 1994. **(29 citations)**.
- [43] A. Ferguson and J. Hughes, *Fast Abstract Interpretation Using Sequential Algorithms*, Proc. Workshop on Static Analysis, Padova, Springer LNCS Vol. 724, 1993. **(15 citations)**.

- [44] J. Launchbury, A. Gill, J. Hughes, S. Marlow, S.L.Peyton-Jones, and P. Wadler, *Avoiding Unnecessary Updates*, Proc. Glasgow 1992 Workshop on Functional Programming, Springer-Verlag Workshops in Computing. **(27 citations)**.
- [45] A. Ferguson and J. Hughes, *Abstract Interpretation of Higher-Order Functions using Concrete Data Structures*, Proc. Glasgow 1992 Workshop on Functional Programming, Springer-Verlag Workshops in Computing. **(3 citations)**.
- [46] J. Hughes and A. Moran, *A Semantics for Locally Bottom-Avoiding Choice*, Proc. Glasgow 1992 Workshop on Functional Programming, Springer-Verlag Workshops in Computing. **(5 citations)**.
- [47] J. Hughes and A. Ferguson, *A Loop-detecting Interpreter for Lazy, Higher-order Programs*, Proc. Glasgow 1992 Workshop on Functional Programming, Springer-Verlag Workshops in Computing. **(7 citations)**.
- [48] John Hughes and John Launchbury, *Relational Reversal of Abstract Interpretation*, Journal of Logic and Computation, special issue on Abstract Interpretation, Vol.2 No. 4, pp465-482, 1992. **(1 citations)**.
- [49] John Hughes and John Launchbury, *Reversing Abstract Interpretations*, Proc. European Symposium on Programming, Rennes, 1992. **(0 citations)**.
- [50] John Hughes and John Launchbury, *Projections for Polymorphic First-Order Strictness Analysis*, *Mathematical Structures in Computer Science*, Vol. 2, No. 3, pp301-326, 1992. **(5 citations)**.
- [51] Ryszard Kubiak, John Hughes and John Launchbury, *Implementing Projection-based Strictness Analysis*, Proc. Glasgow 1991 Workshop on Functional Programming, Springer-Verlag Workshops in Computing, 1992. **(8 citations)**.
- [52] Carsten Kehler Holst and John Hughes, *A Loop Detecting Interpreter for Lazy Programs*, Proc. Glasgow 1991 Workshop on Functional Programming, Springer-Verlag Workshops in Computing, 1992. **(4 citations)**.
- [53] John Hughes (ed.), *Proc. ACM Conference on Functional Programming Languages and Computer Architecture*, Springer LNCS Vol. 523, 1991. **Citation count missing!**
- [54] Carsten Kehler Holst and John Hughes, *Towards Binding-time Improvement for Free!* Proc. Glasgow 1990 Workshop on Functional Programming, Springer-Verlag Workshops in Computing, 1991. **(19 citations)**.
- [55] John Hughes and John Launchbury, *Towards Relating Forwards and Backwards Analyses*, Proc. 1990 Glasgow Workshop on Functional Programming, Springer-Verlag Workshops in Computing, 1991. **(13 citations)**.

- [56] John Hughes and John O'Donnell, *Nondeterministic functional programming with sets*, Proceedings of the 4th Higher-Order Workshop, ed. Graham Birtwhistle, Springer-Verlag Workshops in Computing, Banff, 1990. **(10 citations)**.
- [57] John Hughes, *Why Functional Programming Matters*, Computer Journal, Vol. 32, No. 2, 1989, and in Research Topics in Functional Programming, ed. David Turner, Addison Wesley, 1990. **(614 citations)**.
- [58] John Hughes and John O'Donnell, *Expressing and Reasoning about Nondeterministic Functional Programs*, Proc. Glasgow 1989 Workshop on Functional Programming, Springer-Verlag Workshops in Computing, 1990. **(49 citations)**.
- [59] Alex Ferguson and John Hughes, *An Iterative Powerdomain Construction*, Proc. Glasgow 1989 Workshop on Functional Programming, Springer Verlag Workshops in Computing, 1990. **(6 citations)**.
- [60] John Hughes, *Compile-time Analysis of Functional Programs*, in Research Topics in Functional Programming, ed. David Turner, Addison Wesley, 1990. **(44 citations)**.
- [61] John Hughes, *Projections for Polymorphic Strictness Analysis*, Proc. Conference on Category Theory and Computer Science, Manchester, 1989. **(17 citations)**.
- [62] John Hughes, *Abstract Interpretation of First-Order Polymorphic Functions*, Proc. Workshop on Implementation of Lazy Functional Languages, eds. Johnsson, Peyton-Jones and Karlsson, Chalmers University, 1988. **(19 citations)**.
- [63] John Hughes, *Backwards Analysis of Functional Programs*, IFIP Workshop on Partial Evaluation and Mixed Computation, eds. Bjoerner and Ershov, 1987. **(99 citations)**.
- [64] Phil Wadler and John Hughes, *Projections for Strictness Analysis*, IFIP Conference on Functional Languages and Computer Architecture, 1987. **(203 citations)**.
- [65] John Hughes, *Analyzing Strictness by Abstract Interpretation of Continuations*, in Abstract Interpretation of Declarative Languages, eds. Abramsky and Hankin, Ellis-Horwood, 1987. **(25 citations)**.
- [66] Guy Argo, Jon Fairbairn, John Hughes, John Launchbury, and Phil Trinder, *Implementing Functional Databases*, in Proc. International Workshop on Database Programming Languages, Roscoff, France, 1987. **(19 citations)**.

- [67] Richard Bird and John Hughes, *The Alpha-Beta Algorithm: An Exercise in Program Transformation*, Information Processing Letters Vol 24 No 1, 1986. **(14 citations)**.
- [68] John Hughes, *A Novel Representation of Lists and its Application to the Function “reverse”*, Information Processing Letters Vol 22 No 3, 1986. **(59 citations)**.
- [69] John Hughes, *Strictness Detection in Non-flat Domains*, Workshop on Programs as Data Objects, Springer LNCS Vol. 217, 1985. **(42 citations)**.
- [70] John Hughes, *Lazy Memo-functions*, IFIP Symposium on Functional Languages and Computer Architecture, 1985. **(137 citations)**.
- [71] John Hughes, *A Distributed Garbage Collection Algorithm*, IFIP Symposium on Functional Languages and Computer Architecture, 1985. **(144 citations)**.
- [72] John Hughes, *The Design and Implementation of Programming Languages*, Oxford University D.Phil. thesis, 1983. **(125 citations)**.
- [73] John Hughes, *A Semi-incremental Garbage Collection Algorithm*, Software Practice and Experience, 1982. **(13 citations)**.
- [74] John Hughes, *Graph reduction with super-combinators*. Oxford University Computing Laboratory tech report, 1982. **(17 citations)**.
- [75] John Hughes, *Super-combinators: A New Implementation Method for Applicative Languages*, Proc. ACM Symposium on Lisp and Functional Programming, 1982. **(164 citations)**.