Publications by Cordelia Hall

January 2004

Books

Cordelia Hall and John O'Donnell, *Discrete Mathematics Using a Computer*, Springer Verlag, January 2000, ISBN 1-85233-089-9. 339 pages.

Presents discrete mathematics (logic, sets, relations, etc.) using Haskell, a functional programming language. The student learns new concepts, then writes functions and experiments by applying them. This process refines and develops first impressions of the concepts required. A second edition is scheduled for autumn of 2004.

Papers

Publications marked 'J' are either in journals, or in 'journal equivalents': international conferences with rigorous peer review and acceptance rates comparable to or more exclusive than those of many prominent journals. Papers marked 'I' were invited, papers marked 'W' are refereed but appeared at workshops where the acceptance rate was quite high, and papers marked 'T' are technical reports, possibly of published papers. Papers marked 'S' are submitted for publication.

1. **C**: Cordelia Hall and Eileen Maitland, "Cryogenics and Creativity: The Frankenstein Factor in Cultural Preservation", *Digital Resources in the Humanities 2003*, University of Gloucestershire, Cheltenham.

2. **C**: C. Boehm, D. MacLellan, C. Hall, "MuTaTeD II: A system for music information retrieval of encoded music", *Proceedings of the International Computer Music Conference*, 4 pages, August-September 2000.

3. **J**: S. Brewster, A. Capriotti, C. Hall, "Using compound earcons to represent hierarchies", HCI Letters 1, 1, pp 6-8, 1998.

4. **J**:N. Hutchison, U. Neuhaus, M. Schmidt-Schauss, C. Hall, "NATURAL EXPERT: A commercial functional programming environment", *Journal of Functional Programming*, 7(2), pp 163-182, March 1997.

5. **C**: B. Findlay, C. Hall, "Simulation of bowing decisions on a string instrument", Quatriemes Journees d'Informatique Musicale, pp 89—97, 1997.

6. **C**: S. Brewster, A. Capriotti, C. Hall, "Using compound earcons to represent hierarchies", Proceedings of BCS HCI'97 (Bristol, UK), Springer Verlag.

7. **J**: C. Hall, K. Hammond, S. Peyton Jones P. Wadler, "Type Classes With a Static Semantics For Haskell", ACM Trans. on Programming Languages and Systems (TOPLAS), Vol 18, No 2, pp 109-138, 1996.

8. **J**: C. Hall,"Using Hindley-Milner type inference to optimise list representation", *1994 ACM SIGPLAN Symposium on LISP and Functional Programming*, pp 162-172, June 1994.

9. **J**: C. Hall, K. Hammond, S. Peyton Jones and P. Wadler, "Type classes in Haskell", in *5th European Symposium on Programming*, Springer-Verlag LNCS 788, pp 241-256, April 1994.

10. **T**: C. Hall, K. Hammond, S. L. Peyton Jones and P. Wadler, “Type classes in Haskell”, Report FP-94-04, Department of Computing Science, University of Glasgow, January 1994 .

11. **J**: C. Hall, “Using overloading to express distinctions between evaluators”, *Information Processing Letters* 48, pp 1—8, 1993.

12. **T**: C. Hall, “Using overloading to express distinctions between evaluators”, Report FP-1993-6, Department of Computing Science, University of Glasgow, July 1993.

13. **I**: S. L. Peyton Jones, C. Hall, K. Hammond, W. D. Partain, P. Wadler, “ The Glasgow Haskell compiler: a technical overview”, in *Proceedings of the Joint Framework for Information Technology (JFIT) Conference*, 10 pages, March 1993.

14. **W**: C. Hall, “A framework for optimising abstract data types”, in eds K. Hammond, J. T. O’Donnell, *Proceedings of the Glasgow Workshop on Functional Programming*, Springer-Verlag Workshops in Computing, ed. C.J. van Rijsbergen, 1993, about 10 pages.

15. **W**: C. Hall, K. Hammond, W. Partain, S. L. Peyton Jones and P. Wadler, “The Glasgow Haskell Compiler: A retrospective”, in eds J. Launchbury, P. Sansom, *Proceedings of the Glasgow Workshop on Functional Programming*, Springer-Verlag Workshops in Computing, ed. C.J. van Rijsbergen, pp 62—71, 1992.

16. **W**: C. Hall, “Strictness analysis using Hindley-Milner type inference”, in eds R. Heldal, C. Kehler Holst, P. Wadler, *Proceedings of the Glasgow Workshop on Functional Programming*, Springer-Verlag Workshops in Computing, ed. C.J. van Rijsbergen, pp 128—133, 1991.

17. **I**: C. Hall, “Using lazy evaluation to find fixpoints in infinite domains”, in, eds. G. David, R.T.Boute and B.D.Shriver, *Declarative Systems*, Elsevier (North-Holland), pp 85—98, 1990.

18. **W**: C. Hall, K. Hammond, J. O’Donnell, “An algorithmic and semantic approach to debugging”, eds. S. Peyton Jones, G. Hutton and C. Kehler Holst, *Proceedings of the Glasgow Workshop on Functional Programming*, Springer-Verlag Workshops in Computing, ed. C. van Rijsbergen, pp 44—53, 1990.

19. **J**: C. Hall, D. Wise, “Generating function versions with rational strictness patterns”, *Science of Computer Programming* 12, pp 39—74, 1989 .

20. **T**: J. O’Donnell, C. Hall, “Hydra: A digital circuit design system”, Report SA/TR-4/89, Sabbagh Associates Inc., Bloomington (March 1989).

21. **J**: J. O’Donnell, C. Hall, “Debugging in applicative languages”, *Lisp and Symbolic Computation*, Vol. 1, No. 2, pp 113—145, 1988.

22. **W**: C. Hall, “Finding fixpoints in infinite domains”, eds. K. Davis, R. J. M. Hughes, *Proceedings of the Glasgow Workshop on Functional Programming*, Springer-Verlag Workshops in Computing, ed. C. van Rijsbergen, 1988, about 10 pages.

23. **J**: C. Hall, D. Wise, “Compiling strictness into streams”, *Fourteenth Annual ACM SIGACT-SIGPLAN Symposium on Principles of Programming Languages*, pp 132—143, January 1987.

24. **T**: C. Hall, “Strictness analysis applied to programs with lazy list constructors”, PhD Thesis, Indiana University, 1987.

25. **J**: C. Hall, J. O'Donnell, "Debugging in a side effect free programming environment", *1985 ACM SIGPLAN Symposium on Programming Languages and Programming Environments*, SIGPLAN Notices Vol. 20, No. 7, pp 60—68, June 1985.

26. **T**: K. Hammond, C. Hall, "A dynamic semantics for Haskell", 10 pages.