JFP **12** (4 & 5): 293–294, July & September 2002. © 2002 Cambridge University Press
DOI: 10.1017/S095679680200922X Printed in the United Kingdom

Special Double Issue on Haskell

GRAHAM HUTTON

School of Computer Science and IT, University of Nottingham, Nottingham, UK

Since its inception in 1987, Haskell has provided a focal point for research in lazy functional programming. During this time the language has continually evolved, as a result of both theoretical advances and practical experience. Haskell has proved to be a powerful tool for many kinds of programming tasks, and an excellent vehicle for many aspects of computing pedagogy and research. The recent definition of Haskell 98 provides a long-awaited stable version of the language, but there are many exciting possibilities for future versions of Haskell.

This special issue of the Journal of Functional Programming is devoted to Haskell, and follows on from a series of four workshops on the language that were held during the period 1995–2000, and have continued to be held every year since. Contributors to any of the four workshops were invited to submit full articles to the special issue, but submission was open to everyone. The original call solicited articles on any aspect of Haskell, including critiques of Haskell 98, new proposals for Haskell, applications or case studies, programming techniques, reasoning about programs, semantic issues, pedagogical issues, and implementation. Six articles were selected for publication, each of which is briefly summarised below:

- Faxén gives a more-or-less complete static semantics for Haskell 98 that formally specifies the details of the type system;
- Marlow shows how to develop a web server using Haskell that is less than 1500 lines long but provides surprising performance;
- McBride shows how recent extensions to the Haskell type system can be used to simulate certain aspects of dependent types;
- Peyton Jones and Marlow reveal the techniques and tricks used to build an effective inliner for the Glasgow Haskell Compiler;
- Thiemann presents combinator libraries for generating HTML and XML documents that guarantee certain validity properties;
- Trinder, Loidl and Pointon provide a comprehensive survey of current extensions of Haskell for parallel and distributed programming.

These articles cover a diverse range of topics, but there are of course many other areas of Haskell research that are not touched upon here. It is encouraging to note that a special *double* issue was necessary to accommodate the accepted articles, and that four of the six articles are already looking beyond Haskell 98 to the future, by making essential use of additional language features such as multi-parameter classes, functional dependencies, and concurrency.

I would like to thank the authors and the referees for their efforts in producing and reviewing the articles, and Phil Wadler for the opportunity to publish the articles as a special issue of the Journal of Functional Programming.

Graham Hutton
School of Computer Science and IT
University of Nottingham

https://doi.org/10.1017/S095679680200922X Published online by Cambridge University Press