



Contents lists available at ScienceDirect

Computer Languages, Systems & Structures

journal homepage: www.elsevier.com/locate/cl

Editorial

Special issue on Trends in Functional Programming 2013/14



This special issue represents a selection of the best papers of TFP'13 and TFP'14, in substantially revised versions. The authors of the three best papers in each instance were invited to submit revised versions of their research, reflecting new developments since the TFP version of the paper. In each case, at least 30% of the material in the journal paper is new.

The Symposium on Trends in Functional Programming (TFP) is dedicated to promoting new research directions related to the field of functional programming and to investigate the relationships of functional programming with other branches of computer science. It is designed to be a platform for novel and upcoming research, offering a lively environment for presenting the latest research results. A formal post-symposium refereeing process then selects a high-quality set of articles from those presented at the symposium and submitted for formal publication. The journal versions of the best papers from these proceedings then offer an opportunity for the authors to take a longer term view of their research efforts and to present new results since the original presentation at TFP.

This special issue covers two instances of the TFP symposium. TFP 2013 was held in Provo, Utah in the United States of America in May 2013, hosted by Brigham Young University. For TFP 2013, 24 papers were submitted to the formal refereeing process and 10 of these were accepted by the programme committee. TFP 2014 was the main event of a series of events around functional programming, was held in Soesterberg, the Netherlands in May 2014. In total, 22 papers were submitted to the formal refereeing process and eight papers were accepted.

TFP traditionally pays special attention to research students, acknowledging that students are almost by definition part of new subject trends. These papers also receive an extra round of feedback by the programme committee before they are submitted to the standard review process for formal publication. In this way, students can improve their papers before they compete within a full formal refereeing process. This is often a useful stepping stone in the process of academic publications, although in this specific special issue no student papers are presented. Details on TFP can be found on this webpage: <http://www.tifp.org/>.

The paper by Joachim Breitner, Call Arity, discusses an analysis to improve run-time performance of functional programs, by eta-expanding functions based on how they are used. Introducing the notion of co-call graphs, it improves on existing analyses, enabling fusion for foldl-like functions. It has been implemented in the de facto standard Haskell compiler GHC, and part of the default setting since version 7.10.1. The paper shows by means of a benchmark that the gains are large in some cases, and good overall.

Debugging has always been a challenge for (lazy) functional languages. In the paper by Maarten Faddegon and Olaf Chitil, the authors apply generic programming techniques to reduce to burden on the programmer for defining how values observed, and use metaprogramming techniques to specify which parts of those values can be observed.

We would like to thank the sponsors for these two TFP instances: Erlang Solutions Ltd, Well-Typed, the Dutch Association for Science (NWO), the Department of Information and Computing Sciences of Utrecht University in the Netherlands, and

Brigham Young University from Provo, Utah in the United States of America. Finally, we would like to thank our editor Marjan Mernik and the people at Elsevier for their efforts in making this special issue happen.

Jurriaan Hage*

Department of Information and Computing Sciences, Utrecht University, Utrecht, The Netherlands

Jay McCarthy

Department of Computer Science, University of Massachusetts Lowell, Lowell, MA, USA

*Corresponding author.

E-mail address: J.Hage@uu.nl (J. Hage)