Math. Struct. in Comp. Science (2016), *vol.* 26, *pp.* 1–2.
doi:10.1017/S0960129514000097 First published online 27 November 2014

Selected papers from Dependently Typed Programming 2010 – Overview

THORSTEN ALTENKIRCH[†] and CONOR MCBRIDE[‡]

[†]*University of Nottingham, School for Computer Science, Nottingham, NG9 2BB, United Kingdom*
Email: txa@cs.nott.ac.uk
[‡]*University of Strathclyde, United Kingdom*

This special issue comprises selected papers which were presented at the workshop on dependently typed programming (DTP 10) in Edinburgh in July 2010 – affiliated with Federated Logic conferences (FLOC 10). Earlier workshops on dependently typed programming took place in Nottingham in 2008 (DTP 2008) and there also has been a Dagstuhl seminar (04381) on this subject in 2004. After DTP 2010, in 2011 a workshop on dependently typed programming (DTP 11) took place in Nijmegen affiliated with Interactive Theorem Proving 2011 (ITP 11). In September 2011 there also was a DTP workshop in Shonan, Japan.

DTP seeks to incorporate ideas from Martin-Löf type theory into functional programming languages. This enables programmers to capture relationships between data, internalizing invariants necessary for appropriate computation. When data describe types, we can express patterns of programming in code. We are beginning to see how to take advantage of the power and precision which dependent types afford, but there are still plenty of problems to address and issues to resolve. The design space is huge: there is much to do in designing programming languages like Agda, Coq, Idris or Epigram put also in applying DTP in discovering new programming patterns.

The present special issue is a sample of the work going on in this community covering many diverse aspects. Ahrens in his paper *Modules over relative monads for syntax and semantics* describes a very high level programming pattern for implementing high level programming languages inspired by category theory and implements this in a DTP language: Agda. The overview paper by Bove, Krauss and Sozeau on *Partiality and Recursion in Interactive Theorem Provers* provides a summary of the work on a particularly thorny subject: the question of totality which is essential for logical soundness but often hard to achieve for the programmer. Hancock and Ghani in *Containers, Monads and Induction Recursion* use category theory to investigate the concept of induction–recursion, which is frequently used in DTP but so far lacks a sound theoretical underpinning. Kanso and Setzer tackle an important practical issue – the integration of theorem provers in DTP – in their paper *A Light-Weight Integration of Automated and Interactive Theorem Proving* and describe interesting applications of their technology to safety sensitive areas like railways.

We would like to thank all the contributors to this special issues and the anonymous referees and apologize to everybody for the delay in publishing these proceedings. The passing of time shows that DTP is becoming increasingly relevant reaping the rewards of the work presented here and elsewhere.