Contents

Preface vii
Acknowledgements ix

Generic Programming within Dependently Typed Programming 1
Thorsten Altenkirch, Conor McBride

Generic Haskell, Specifically 21
Dave Clarke, Andres Löh

Generic Accumulations 49
Alberto Pardo

A Generic Algorithm for Minimum Chain Partitioning 79
Sharon A. Curtis

Concrete Generic Functionals 89
Raymond T. Boute

Discussion: The Rôle of Types in Generic Programming 121

Making the Usage of STL Safe 127
Douglas Gregor, Sibylle Schupp

Static Data Structures 141
Michael Burton, William Griswold, Andrew McCulloch, Gary Huber

Adaptive Extensions of Object-Oriented Systems 171
Olaf Seng, Thomas Genussler, Benedikt Schulz

Complete Traversals as General Iteration Patterns 187
William Klostermeyer, David Musser, Arturo Sánchez-Ruiz

Efficient Implementation of Run-time Generic Types for Java 207
Eric Allen, Robert Cartwright, Brian Stoler

Discussion: Views of Genericity 237
Preface

Generic programming is a matter of making programs more adaptable by making them more general. Generic programs often embody non-traditional kinds of polymorphism; ordinary programs are obtained from them by suitably instantiating their parameters. In contrast with normal programs, the parameters of a generic programs are often quite rich in structure. For example they may be other programs, types or type constructors, class hierarchies, or even programming paradigms.

Generic programming techniques have always been of interest, both to practitioners and to theoreticians, but only recently have generic programming techniques become a specific focus of research in the functional and object-oriented programming language communities. The IFIP TC2 Working Conference on Generic Programming, held at Schloß Dagstuhl, Germany, on 11th and 12th July 2002, brought together leading researchers in generic programming from around the world, and featured papers capturing the state of the art in this important emerging area.

The conference was sponsored by IFIP Technical Committee 2, and organized in cooperation with Working Group 2.1 on Algorithmic Languages and Calculi.

This book contains revised versions of the papers that were presented at the conference. The papers are recorded in order of presentation. The editors are grateful to the members of the Programme Committee and the reviewers for their careful work, and particularly to Bernhard Möller for also making the local arrangements for the conference. They would also like to thank Lambert Meertens, chairman of WG2.1, for proposing the conference and providing useful guidance and inspiration, Tim Sheard for his elegant definition of generic programming (quoted above), and Yana Lambert of Kluwer Academic Publishers for general support in the preparation of these proceedings. Financial support for the conference from the Deutsche Forschungsgemeinschaft is gratefully acknowledged.

Jeremy Gibbons
Johan Jeuring
Acknowledgements

Programme Committee

Matt Austern  
Eerke Boiten  
Ulrich W. Eisenecker  
Jeremy Gibbons (co-chair)  
Ralf Hinze  
Johan Jeuring (co-chair)  
Gary T. Leavens  
Karl Lieberherr  
Lambert Meertens  
Eugenio Moggi  
Bernhard Möller  
Oege de Moor  
David Musser  
Martin Odersky  
Ross Paterson  
Simon Peyton Jones  
Colin Runciman  
Doaitse Swierstra  
Stephanie Weirich

Reviewers

Matt Austern  
Andreas Behrend  
Eerke Boiten  
Dave Clarke  
Ulrich W. Eisenecker  
Jeremy Gibbons  
Ralf Hinze  
Johan Jeuring  
Gary T. Leavens  
Karl Lieberherr  
Lambert Meertens  
Andres Löh  
Eugenio Moggi  
Bernhard Möller  
Oege de Moor  
David Musser  
Martin Odersky  
Ross Paterson  
Dusko Pavlovic  
Simon Peyton Jones  
Christine Röckl  
Colin Runciman  
Doaitse Swierstra  
Stephanie Weirich  
Matthias Zenger