

Commenced Publication in 1973

Founding and Former Series Editors:

Gerhard Goos, Juris Hartmanis, and Jan van Leeuwen

Editorial Board

David Hutchison

Lancaster University, UK

Takeo Kanade

Carnegie Mellon University, Pittsburgh, PA, USA

Josef Kittler

University of Surrey, Guildford, UK

Jon M. Kleinberg

Cornell University, Ithaca, NY, USA

Alfred Kobsa

University of California, Irvine, CA, USA

Friedemann Mattern

ETH Zurich, Switzerland

John C. Mitchell

Stanford University, CA, USA

Moni Naor

Weizmann Institute of Science, Rehovot, Israel

Oscar Nierstrasz

University of Bern, Switzerland

C. Pandu Rangan

Indian Institute of Technology, Madras, India

Bernhard Steffen

TU Dortmund University, Germany

Madhu Sudan

Microsoft Research, Cambridge, MA, USA

Demetri Terzopoulos

University of California, Los Angeles, CA, USA

Doug Tygar

University of California, Berkeley, CA, USA

Gerhard Weikum

Max Planck Institute for Informatics, Saarbruecken, Germany

Jeremy Gibbons (Ed.)

Generic and Indexed Programming

International Spring School, SSGIP 2010
Oxford, UK, March 22-26, 2010
Revised Lectures

Volume Editor

Jeremy Gibbons
University of Oxford
Department of Computer Science
Wolfson Building, Parks Road, Oxford OX1 3QD, UK
E-mail: jeremy.gibbons@cs.ox.ac.uk

ISSN 0302-9743 e-ISSN 1611-3349
ISBN 978-3-642-32201-3 e-ISBN 978-3-642-32202-0
DOI 10.1007/978-3-642-32202-0
Springer Heidelberg Dordrecht London New York

Library of Congress Control Number: 2012943139

CR Subject Classification (1998): D.3, F.3, D.2, D.1, F.4.1, E.1

LNCS Sublibrary: SL 1 – Theoretical Computer Science and General Issues

© Springer-Verlag Berlin Heidelberg 2012

This work is subject to copyright. All rights are reserved, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, re-use of illustrations, recitation, broadcasting, reproduction on microfilms or in any other way, and storage in data banks. Duplication of this publication or parts thereof is permitted only under the provisions of the German Copyright Law of September 9, 1965, in its current version, and permission for use must always be obtained from Springer. Violations are liable to prosecution under the German Copyright Law.

The use of general descriptive names, registered names, trademarks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

Preface

Generic programming is about making programs more widely applicable via exotic kinds of parametrization—not just along the dimensions of values or of types, but also of things such as the shape of data, algebraic structures, strategies, computational paradigms, and so on. *Indexed programming* is a lightweight form of dependently typed programming, constraining flexibility by allowing one to state and check relationships between parameters: that the shapes of two arguments agree, that an encoded value matches some type, that values transmitted along a channel conform to the stated protocol, and so on.

The two forces of genericity and indexing balance each other nicely, simultaneously promoting and controlling generality. The *Generic and Indexed Programming* project at Oxford was funded by the UK Engineering and Physical Sciences Research Council over the period 2006–2010 to explore the interaction between these two forces. The closing activity of the project took the form of a *Spring School on Generic and Indexed Programming*, held at Wadham College, Oxford, during March 22–24, 2010; this volume collects the revised lecture notes from the school.

The school was—and these lecture notes are—aimed at doctoral students, researchers, and practitioners in programming languages and related areas. A good grounding is assumed in typed functional programming, as in Haskell or OCaml. Six lecturers from the programming languages community, each an acknowledged expert in their specialism, covered various aspects of generic and indexed programming; each gave about four hours’ lectures, distributed throughout the week of the school. Lecture notes from five of those six sets of lectures are included here:

- Nate Foster on three approaches to *bidirectional programming*, for specifying consistent mappings to and from a data representation
- Ralf Hinze on using *adjunctions* to unify and generalize a number of familiar generic recursion schemes
- Oleg Kiselyov on the *typed tagless interpreter* approach for encoding a typed object language in a typed meta-language
- Jeremy Siek on the debates that took place over the attempt to incorporate *concepts in the revised C++ standard*
- Stephanie Weirich on *datatype- and arity-generic programming within a dependently typed language*

The sixth lecturer, Simon Peyton Jones, spoke on *type functions* in Haskell, and their use in generic programming; his lecture notes are represented by his chapter “Fun with Type Functions” with Oleg Kiselyov and Chung-chieh Shan in the book *Reflections on the Work of C. A. R. Hoare*, edited by Cliff Jones, Bill Roscoe, and Ken Wood (Springer, 2010, ISBN 978-1-84882-911-4) in honour of

Sir Tony Hoare's 75th birthday. Slides for all six sets of lectures are available on the school's website <http://www.cs.ox.ac.uk/projects/gip/school.html>.

I would like to express my sincere thanks to the six lecturers at the school, for the considerable effort they devoted to making the event a success; to their co-authors, for helping to write up the lecture notes; to the staff of Wadham College, especially Jan Trinder, for their hospitality in the college's 400th year; to EPSRC, for their financial support; and last but not least, to the 41 participants, for making it all worthwhile.

May 2012

Jeremy Gibbons

Table of Contents

Three Complementary Approaches to Bidirectional Programming	1
<i>Nate Foster, Kazutaka Matsuda, and Janis Voigtländer</i>	
Generic Programming with Adjunctions	47
<i>Ralf Hinze</i>	
Typed Tagless Final Interpreters	130
<i>Oleg Kiselyov</i>	
The C++0x “Concepts” Effort	175
<i>Jeremy G. Siek</i>	
Generic Programming with Dependent Types	217
<i>Stephanie Weirich and Chris Casinghino</i>	
Author Index	259