Lecture Notes in Computer Science

Edited by G. Goos and J. Hartmanis

389

D.H. Pitt D.E. Rydeheard
P. Dybjer A.M. Pitts A. Poigné (Eds.)

Category Theory and Computer Science

Manchester, UK, September 5–8, 1989
Proceedings

Springer-Verlag
Berlin Heidelberg New York London Paris Tokyo Hong Kong

Editors

David H. Pitt
Department of Mathematics, University of Surrey
Guildford, Surrey, GU2 5XH, UK

David E. Rydeheard
Department of Computer Science, The University
Manchester, M13 9PL, UK

Peter Dybjer
Chalmers University of Technology, Department of Computer Sciences
S-41296 Göteborg, Sweden

Andrew M. Pitts
University of Cambridge Computer Laboratory, New Museum Site
Pembroke Street, Cambridge, CB2 3QG, UK

Axel Poigné
GMD, F2 G2, Schloss Birlinghoven, Postfach 1240
D-5205 Sankt Augustin 1, FRG

CR Subject Classification (1987): D.2.1, D.3.1, D.3.3, F.3, F.4.1

ISBN 3-540-51662-X Springer-Verlag Berlin Heidelberg New York
ISBN 0-387-51662-X Springer-Verlag New York Berlin Heidelberg

Printed in Germany

Printing and binding: Druckhaus Beltz, Hemsbach/Bergstr.
2145/3140-543210 – Printed on acid-free paper

Preface

This collection of papers records some applications of category theory to computer science. The papers were presented at the third in a series of biennial conferences on Category Theory and Computer Science. The proceedings of the previous two conferences (Guildford 1985 and Edinburgh 1987) also appear in the Springer Lecture Notes in Computer Science, Numbers 240 and 283.

Traditionally, category theory has been used in mathematics as an abstract language for the formulation of definitions and the organisation of concepts in areas such as algebra and topology. Applications in computer science draw upon this and other aspects of the theory. It may help the reader if we briefly outline several of the more important links that have been established between category theory and computing.

One of the key ideas is the representation of programming languages as categories. This is particularly appropriate for languages based upon typed lambda calculi where the types become objects in a category and lambda terms (programs) become arrows. Conversions between programs are treated as equality, or alternatively, making the conversions explicit, as 2-cells. Composition is substitution of programs for free variables. Multiple variables are handled by admitting categories with finite products. This treatment enforces a stratification based upon the types of variables and expressions. For example, languages with type variables lead to indexed (or fibred) categories. Constructs in programming languages correspond to structure within categories, and categories with sufficient structure delimit the semantics of a language.

Correctness issues in programming require the introduction of logics for reasoning about program behaviour. Logics arising in category theory, for example internal logics in toposes, are, in general, intuitionistic, and in this sense appropriate for reasoning about constructions and programs. Propositional logics can be modelled directly as categories whose objects are propositional formulae and whose arrows are formal proofs. Predicate logics then become indexed categories of propositional logics. Logical constants are introduced as adjoints. Here we see a general algebraic framework for the analysis of programs, programming languages and their logics.

Other applications of category theory may be found in the development of universal algebra (theories for both program specification and computation) and of domain theory (for application in denotational semantics). Altogether, category theory provides a rich setting for the investigation of some foundational issues in computer science. The papers collected here illustrate recent progress in this area.

The organisers would like to thank the Science and Engineering Research Council for their support under the Logic for IT Initiative and the London Mathematical Society for their support. The Department of Computer Science at the University of Manchester kindly hosted the meeting. Local arrangements were ably handled by Jane Gray.

D.H.Pitt
D.E.Rydeheard
P.Dybjer
A.Pitts
A.Poigné

Organising and Program Committee

S. Abramsky, P. Dybjer, P.L. Curien, D.H. Pitt, A.M. Pitts, A. Poigné, D.E. Rydeheard, D.T. Sannella, E. Wagner.

Referees

S. Abramsky, S. Ambler, A. Asperti, I. Bethke, J. Bradfield, D.P. Carlisle, T. Coquand, P.L. Curien, M. Dam, P. Dybjer, H. Ehrig, F. Fages, M. Fourman, D. Gurr, T. Hagino, W. Harwood, M. Hedberg, B. Hilken, H. Huwig, J.M.E. Hyland, P.T. Johnstone, K.Karlsson, Y. Lafont, F. Lamarche, P. Mathieu, M. Mauny, J.Mitchell, W.P.R.Mitchell, E. Moggi, K. Moody, L.Moss, F. Nielson, D.H. Pitt, A.M.Pitts, A.Poigné, J.C.Raoult, G. Rosolini, D.E. Rydeheard, H.Sander, D.T.Sannella, B.Steffen, T.Streicher, P.Taylor, E. Wagner, G. Winskel.

Contents