ABRAMSKY, GABBAY, and MAIBAUM

Handbook of Logic in Computer Science

VOLUME 3

Handbook of Logic in Computer Science

0 19 853762 X

OXFORD

OXFORD SCIENCE PUBLICATIONS

HANDBOOKS OF LOGIC IN COMPUTER SCIENCE
and
ARTIFICIAL INTELLIGENCE AND LOGIC PROGRAMMING

Executive Editor
Dov M. Gabbay

Administrator
Jane Spurr

Handbook of Logic in Computer Science

Volume 1 Background: Mathematical structures
Volume 2 Background: Computational structures
Volume 3 Semantic structures
Volume 4 Semantic modelling
Volume 5 Theoretical methods in specification and verification
Volume 6 Logical methods in computer science

Handbook of Logic in Artificial Intelligence and Logic Programming

Volume 1 Logical foundations
Volume 2 Deduction methodologies
Volume 3 Nonmonotonic reasoning and uncertain reasoning
Volume 4 Epistemic and temporal reasoning
Volume 5 Logic programming

Handbook of Logic in Computer Science

Volume 3

Semantic Structures

Edited by

S. ABRAMSKY
Professor of Computing Science
DOV M. GABBAY
Professor of Computing Science
and
T. S. E. MAIBAUM
Professor of Foundations of Software Engineering

Imperial College of Science, Technology and Medicine London

Volume Co-ordinator

S. ABRAMSKY

CLARENDON PRESS · OXFORD
1994

Oxford University Press, Walton Street, Oxford OX2 6DP
Oxford New York
Athens Auckland Bangkok Bombay
Calcutta Cape Town Dar es Salaam Delhi
Florence Hong Kong Istanbul Karachi
Kuala Lumpur Madras Madrid Melbourne
Mexico City Nairobi Paris Singapore
Taipei Tokyo Toronto
and associated companies in
Berlin Ibadan

Oxford is a trade mark of Oxford University Press

Published in the United States by
Oxford University Press Inc., New York

A catalogue record for this book is available from the British Library

Library of Congress Cataloging in Publication Data
(Data available)
ISBN 0 19 853762 X

Typeset using LaTeX by Jane Spurr

Printed in Great Britain by
Biddles Ltd
Guildford and King's Lynn

Preface

We are happy to present Volume 3 of our Handbook project, on *Semantic Structures*. The previous two volumes presented the background on fundamental mathematical structures—consequence relations, model theory, recursion theory, category theory, universal algebra, topology, and on computational structures—term-rewriting systems, λ-calculi, modal and temporal logics and algorithmic proof systems. The computational structures considered thus far have been predominantly syntactic in character; while the discussion of mathematical structures has been quite general and free-standing.

In the present volume these threads are drawn together. We look at how mathematical structures are used to model computational processes. At first sight, this enterprise looks quite similar to the standard notions of model theory in logic. However, programming language semantics has significant special features and problems of its own, which have required an extensive development both of novel semantic techniques, anf of the underlying mathematical foundations.

The first chapter of this volume concerns Domain Theory, a mathematical theory originated by Dana Scott to provide a mathematical foundation for the semantics of programming languages. In particular, Domain theory allows meaning to be given to *recursive definitions*, both of programs and of data types. In the 25 years since its inception, a very rich mathematical theory has been developed, and the chapter gives a systematic presentation of this theory.

The second Chapter concerns denotational semantics. The idea here is to assign meaning to programs as elements of, or functions between, suitable mathematical structures. Crucially, this assignment of meanings should be *compositional*; the meaning $[\![C(P_1, \ldots, P_n]\!]$ of a complex program formed by applying some constructor C to the sub-programs $P_1, \ldots, P_n$ should be a function of the meaning of its parts:

$$[\![C(P_1, \ldots, P_n]\!] = \mathcal{C}([\![P_1]\!], \ldots, [\![P_n]\!]),$$

where $\mathcal{C}$ is a suitable mathematical operation on meanings. In practice, the "suitable mathematical structures" will very often be domains; note also that the above equation can be read as saying that semantics is a *homomorphism*, thus invoking concepts of universal algebra. Chapter 2 on

denotational semantics sets out the general principles and techniques of compositional semantics, and shows how they can be applied to a range of programming language features.

Chapter 3 takes up the algebraic theme, focussing on how syntax can be viewed as a free algebra, and a semantics as another algebra, with the semantic algebra as a homomorphism between them (uniquely given because of the freeness of the syntactic algebra). For this programme to work properly it is necessary to combine ideas from Universal algebra (Volume 1) and Domain theory; ideas from category theory are also used extensively.

Finally, Chapter 4 concerns the semantics of types. Types are of increasing importance in modern programming languages. They provide a very effective conceptual discipline in program design, supported by efficient type-checking algorithms which allow many program errors to be detected by the compiler. A wide variety of type systems for programming languages have been considered, embracing such notions as higher-order and recursive types, polymorphism and subtyping. Chapter 4 provides an overview of such type systems and their semantics, which plays an esential role e.g. in establishing the soundness of type inference systems. For the very rich type systems currently being studied, many foundational problems are raised by the work on semantics, so this area shows an important interplay between theory and practice.

The Handbooks

The *Handbook of Logic in Theoretical Computer Science* and its companion, the *Handbook of Logic in Artificial Intelligence and Logic Programming*, have been created in response to a growing need for an in-depth survey of the application of logic in computer science and AI.

We see the creation of the Handbook as a combination of authoritative exposition, comprehensive survey, and fundamental reasearch exploring the underlying unifying themes in the various areas. The intended audience is graduate students and researchers in the areas of computing and logic, as well as other people interested in the subject. We assume as background some mathematical sophistication. Much of the material will also be of interest to logicians and mathematicians.

The tables of contents of the volumes were finalized after extensive discussions between Handbook authors and second readers. The first two volumes present the Background—Mathematical Structures and Computational Structures.

The chapters, which in many cases are of monographic length and scope, are written with emphasis on possible unifying themes. The chapters have an overview, introduction, and main body. A final part is dedicated to more specialized topics.

Chapters are written by internationally renowned researchers in their

respective areas. The chapters are co-ordinated and their contents were discussed in joint meetings. Each chapter has been written using the following procedures:

1. A very detailed table of contents was discussed and co-ordinated at several meetings between authors and editors of related chapters. The discussion was in the form of a series of lectures by the authors. Once an agreement was reached on the detailed table of contents, the authors wrote a draft and sent it to the editors and to other related authors. For each chapter there is a second reader (the first reader is the author) whose job it has been to scrutinize the chapter together with the editors. The second reader's role is very important and has required effort and serious involvement with the authors.

   Second readers for this volume are:
   Chapter 1: Domain Theory—R Heckman
   Chapter 2: Denotational Semantics—C Wadsworth
   Chapter 3: Algebraic Semantics—I Guessarian
   Chapter 4: Semantics of Types—R Crole

2. Once this process was completed (i.e. drafts seen and read by a large enough group of authors), there were other meetings on several chapters in which authors lectured on their chapters and faced the criticism of the editors and audience. The final drafts were prepared after these meetings.
3. We attached great importance to group effort and co-ordination in the writing of chapters. The first two parts of each chapter, namely the introduction-overview and main body, are not completely under the discretion of the author, as he/she had to face the general criticism of all the other authors. Only the third part of the chapter is entirely for the authors' own personal contribution.

The Handbook meetings were generously financed by OUP, by SERC contract SO/809/86, by the Department of Computing at Imperial College, and by several anonymous private donations.

We would like to thank our colleagues, authors, second readers, and students for their effort and professionalism in producing the manuscripts for the Handbook. We would particularly like to thank the staff of OUP for their continued and enthusiastic support, and Mrs Jane Spurr, our OUP Administrator, for her dedication and efficiency.

London
April 1994

S. Abramsky and D. M. Gabbay

Contents