CAMBRIDGE

Cambridge University Press
978-0-521-03251-3 - Deduction and Declarative Programming
Peter Padawitz
Frontmatter
More information

DEDUCTION AND DECLARATIVE PROGRAMMING

© Cambridge University Press www.cambridge.org

CAMBRIDGE

Cambridge University Press
978-0-521-03251-3 - Deduction and Declarative Programming
Peter Padawitz
Frontmatter
More information

Cambridge Tracts in Theoretical Computer Science

Managing Editor Professor C.J. van Rijsbergen,
Department of Computing Science, University of Glasgow

Editorial Board

S. Abramsky, Department of Computing Science, Imperial College of Science and Technology
P.H. Aczel, Department of Computer Science, University of Manchester
J.W. de Bakker, Centrum voor Wiskunde en Informatica, Amsterdam
J.A. Goguen, Programming Research Group, University of Oxford
J.V. Tucker, Department of Mathematics and Computer Science, University College of Swansea

Titles in the series

1. G. Chaitin *Algorithmic Information Theory*
2. L.C. Paulson *Logic and Computation*
3. M. Spivey *Understanding Z*
4. G. Revesz *Lambda Calculus, Combinators and Functional Programming*
5. A. Ramsay *Formal Methods in Artificial Intelligence*
6. S. Vickers *Topology via Logic*
7. J-Y. Girard, Y. Lafont & P. Taylor *Proofs and Types*
8. J. Clifford *Formal Semantics & Pragmatics for Natural Language Processing*
9. M. Winslett *Updating Logical Databases*
10. K. McEvoy & J.V. Tucker (eds) *Theoretical Foundations of VLSI Design*
11. T.H. Tse *A Unifying Framework for Stuctured Analysis and Design Models*
12. G. Brewka *Nonmonotonic Reasoning*
14. S.G. Hoggar *Mathematics for Computer Graphics*
15. S. Dasgupta *Design Theory and Computer Science*
17. J.C.M. Baeten (ed) *Applications of Process Algebra*
18. J.C.M. Baeten & W. P. Weijland *Process Algebra*
23. E.-R. Olderog *Nets, Terms and Formulas*
26. P.D. Mosses *Action Semantics*
27. W.H. Hesselink *Programs, Recursion and Unbounded Choice*
28. P. Padawitz *Deduction and Declarative Programming*
29. P. Gärdenfors (ed) *Belief Revision*
30. M. Anthony & N. Biggs *Computational Learning Theory*
33. E.G. Manes *Predicate Transformer Semantics*
34. F. Nielson & H.R. Nielson *Two Level Functional Languages*
35. L. Feijs & H. Jonkers *Formal Specification and Design*

© Cambridge University Press www.cambridge.org

CAMBRIDGE

Cambridge University Press
978-0-521-03251-3 - Deduction and Declarative Programming
Peter Padawitz
Frontmatter
More information

DEDUCTION AND DECLARATIVE PROGRAMMING

PETER PADAWITZ
Department of Computer Science
University of Dortmund

CAMBRIDGE
UNIVERSITY PRESS

© Cambridge University Press www.cambridge.org

CAMBRIDGE

Cambridge University Press
978-0-521-03251-3 - Deduction and Declarative Programming
Peter Padawitz
Frontmatter
More information

CAMBRIDGE UNIVERSITY PRESS
Cambridge, New York, Melbourne, Madrid, Cape Town, Singapore, São Paulo

Cambridge University Press
The Edinburgh Building, Cambridge CB2 2RU, UK

Published in the United States of America by Cambridge University Press, New York

www.cambridge.org
Information on this title: www.cambridge.org/9780521417235

© Cambridge University Press 1992

This publication is in copyright. Subject to statutory exception
and to the provisions of relevant collective licensing agreements,
no reproduction of any part may take place without
the written permission of Cambridge University Press.

First published 1992
This digitally printed first paperback version 2006

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

ISBN-13 978-0-521-41723-5 hardback
ISBN-10 0-521-41723-6 hardback

ISBN-13 978-0-521-03251-3 paperback
ISBN-10 0-521-03251-2 paperback

© Cambridge University Press www.cambridge.org

CAMBRIDGE

Cambridge University Press
978-0-521-03251-3 - Deduction and Declarative Programming
Peter Padawitz
Frontmatter
More information

Contents

0	**Introduction**	**1**
0.1	Design specifications	1
0.2	Functional versus logic programs	2
0.3	Goal-directed reasoning	3
0.4	Confluent specifications	5
0.5	Miscellaneous	6
1	**Preliminaries**	**8**
1.1	Terms and substitutions	9
1.2	Atoms, goals and Horn clauses	10
1.3	Models of a specification	13
1.4	Gentzen clauses and inductive theorems	15
1.5	Clause inversion	16
2	**Guards, Generators and Constructors**	**18**
2.1	From verification to synthesis	19
2.2	Generator matrices	20
2.3	Constructor-based clauses	22
2.4	Sample constructor-based theorems	26
2.5	Turning Gentzen into Horn clauses	29
2.6	The decomposition principle	30
3	**Models and Correctness**	**33**
3.1	Initial semantics	35
3.2	Initial correctness	36
3.3	Canonical term structures	38
3.4	Conservative extensions	42
3.5	Program equivalence	44
4	**Computing Goal Solutions**	**47**
5	**Inductive Expansion**	**53**
5.1	Case analyses	53
5.2	Induction steps	59
5.3	Inductive expansion of guarded clauses	61
5.4	The soundness proof	65
5.5	Sample expansions	68
6	**Directed Expansion and Reduction**	**88**
6.1	Ground confluence	90
6.2	Strong termination	95
6.3	Critical clauses	107
6.4	Proving reductive validity	115
6.5	Sample confluence proofs	123
6.6	On rewriting logics	128

© Cambridge University Press www.cambridge.org

CAMBRIDGE

Cambridge University Press
978-0-521-03251-3 - Deduction and Declarative Programming
Peter Padawitz
Frontmatter
More information

7 Implications of Ground Confluence 130
7.1 Strategic expansion 130
7.2 Narrowing 132
7.3 Constructor criteria 135
7.4 Proof by consistency 137
7.5 Reductive versus inductive expansion 146
7.6 Specification refinement 151

8 Examples 162
8.1 Attributed grammars 162
8.2 More arithmetic 164
8.3 Searching and sorting 166
8.3.1 Bubble-, merge- and quicksort 167
8.3.2 Heapsort 169
8.4 Balanced trees 176
8.5 Game trees 181
8.6 Binary graphs 188
8.6.1 Depth-first marking 190
8.6.2 Recursive Schorr-Waite marking 195
8.6.3 Iterative Schorr-Waite marking 204
8.7 Streams and networks 209

9 EXPANDER: Inductive Expansion in SML 219
9.1 The specifications 219
9.2 The commands 221
9.2.1 Starting an expansion 222
9.2.2 Resolving 223
9.2.3 Paramodulating 226
9.2.4 Composed rules 225
9.2.5 Stopping an expansion 226
9.2.6 Miscellaneous 227
9.2.7 Sample expansions 227
9.3 The implementation 241
9.3.1 Scanning 242
9.3.2 Abstract syntax 243
9.3.3 Parsing 245
9.3.4 Unparsing 249
9.3.5 Unifying 251
9.3.6 Renaming 254
9.3.7 Searching and recording 254
9.3.8 The commands 259
9.3.9 Starting an expansion 262
9.3.10 Resolving 263
9.3.11 Paramodulating 264
9.3.12 Composed rules 266
9.3.13 Stopping an expansion 268

References 269

Index 275

© Cambridge University Press www.cambridge.org