CAMBRIDGE

Cambridge University Press
978-0-521-87940-8 - Programming with Higher-Order Logic
Dale Miller and Gopalan Nadathur
Frontmatter
More information

Programming with Higher-Order Logic

Formal systems that describe computations over syntactic structures occur frequently in computer science. Logic programming provides a natural framework for encoding and animating such systems. However, these systems often embody variable binding, a notion that must be treated carefully at a computational level. This book aims to show that a programming language based on a simply typed version of higher-order logic provides an elegant and declarative means for realizing such a treatment. Three broad topics are covered in pursuit of this goal. First, a proof-theoretic framework that supports a general view of logic programming is identified. Second, an actual language called λProlog is developed by applying this view to a higher-order logic. Finally, a methodology for computing with specifications is exposed by showing how several computations over formal objects such as logical formulas, functional programs, λ-terms, and π-calculus expressions can be encoded in λProlog.

DALE MILLER is Director of Research at INRIA-Saclay and LIX, École Polytechnique where he is the Scientific Leader of the Parsifal team. He has been a professor at the University of Pennsylvania, the Pennsylvania State University, and the École Polytechnique, France. Miller is the Editor-in-Chief of the *ACM Transactions on Computational Logic* and has editorial duties on several other journals. He was awarded an ERC Advanced Grant in 2011 and is the recipient of the 2011 Test-of-Time award of the IEEE Symposium on Logic in Computer Science. He works on many topics in the general area of computational logic, including automated reasoning, logic programming, proof theory, unification theory, operational semantics, and, most recently, proof certificates.

GOPALAN NADATHUR is a Professor of Computer Science at the University of Minnesota. He has previously held faculty appointments at Duke University, University of Chicago, and Loyola University of Chicago. Nadathur's research interests span the areas of computational logic, programming languages, and logic programming. His work has been regularly funded by the National Science Foundation and has appeared in the *Journal of the Association of Computing Machinery*, *Information and Computation*, the *Journal of Automated Reasoning*, *Theoretical Computer Science*, and *Theory and Practice of Logic Programming* among other places.

© in this web service Cambridge University Press www.cambridge.org

CAMBRIDGE

Cambridge University Press
978-0-521-87940-8 - Programming with Higher-Order Logic
Dale Miller and Gopalan Nadathur
Frontmatter
More information

© in this web service Cambridge University Press www.cambridge.org

CAMBRIDGE

Cambridge University Press
978-0-521-87940-8 - Programming with Higher-Order Logic
Dale Miller and Gopalan Nadathur
Frontmatter
More information

PROGRAMMING WITH HIGHER-ORDER LOGIC

DALE MILLER
INRIA-Saclay, Île de France &
LIX, École Polytechnique

GOPALAN NADATHUR
University of Minnesota

CAMBRIDGE
UNIVERSITY PRESS

© in this web service Cambridge University Press www.cambridge.org

CAMBRIDGE

Cambridge University Press
978-0-521-87940-8 - Programming with Higher-Order Logic
Dale Miller and Gopalan Nadathur
Frontmatter
More information

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

Cambridge University Press
32 Avenue of the Americas, New York, NY 10013-2473, USA

www.cambridge.org
Information on this title: www.cambridge.org/9780521879408

© Dale Miller and Gopalan Nadathur 2012

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 2012

Printed in the United States of America

A catalog record for this publication is available from the British Library.

ISBN 978-0-521-87940-8 Hardback

Cambridge University Press has no responsibility for the persistence or accuracy of URLs
for external or third-party Internet Web sites referred to in this publication and does not
guarantee that any content on such Web sites is, or will remain, accurate or appropriate.

© in this web service Cambridge University Press www.cambridge.org

Cambridge University Press
978-0-521-87940-8 - Programming with Higher-Order Logic
Dale Miller and Gopalan Nadathur
Frontmatter
More information

To Catuscia, Nadia, and Alexis
— Dale

To the memory of my parents
— Gopalan

© in this web service Cambridge University Press www.cambridge.org

CAMBRIDGE

Cambridge University Press
978-0-521-87940-8 - Programming with Higher-Order Logic
Dale Miller and Gopalan Nadathur
Frontmatter
More information

© in this web service Cambridge University Press www.cambridge.org

CAMBRIDGE

Cambridge University Press
978-0-521-87940-8 - Programming with Higher-Order Logic
Dale Miller and Gopalan Nadathur
Frontmatter
More information

Contents

© in this web service Cambridge University Press www.cambridge.org

CAMBRIDGE

Cambridge University Press
978-0-521-87940-8 - Programming with Higher-Order Logic
Dale Miller and Gopalan Nadathur
Frontmatter
More information

© in this web service Cambridge University Press www.cambridge.org

CAMBRIDGE

Cambridge University Press
978-0-521-87940-8 - Programming with Higher-Order Logic
Dale Miller and Gopalan Nadathur
Frontmatter
More information

© in this web service Cambridge University Press www.cambridge.org

CAMBRIDGE

Cambridge University Press
978-0-521-87940-8 - Programming with Higher-Order Logic
Dale Miller and Gopalan Nadathur
Frontmatter
More information

© in this web service Cambridge University Press www.cambridge.org

CAMBRIDGE

Cambridge University Press
978-0-521-87940-8 - Programming with Higher-Order Logic
Dale Miller and Gopalan Nadathur
Frontmatter
More information

Preface

Formal systems in computer science frequently involve specifications of computations over syntactic structures such as λ-terms, π-calculus expressions, first-order formulas, types, and proofs. This book is concerned, in part, with using higher-order logic to express such specifications. Properties are often associated with expressions by formal systems via syntax-based inference rules. Examples of such descriptions include presentations of typing and operational semantics. Logic programming, with its orientation around rule-based specifications, provides a natural framework for encoding and animating these kinds of descriptions. Variable binding is integral to most syntactic expressions, and its presence typically translates into side conditions accompanying inference rules. While many of the concepts related to binding, such as variable renaming, substitution, and scoping, are logically well understood, their treatment at a programming level is surprisingly difficult. We show here that a programming language based on a simply typed version of higher-order logic provides an elegant approach to performing computations over structures embodying binding.

The agenda just described has a prerequisite: We must be able to make sense of a higher-order logic as a programming language. This is a nontrivial task that defines a second theme that permeates this book. Usual developments of logic programming are oriented around formulas in clausal form with resolution as the sole inference rule. Sometimes a semantics-based presentation is also used, expanding typically into the idea of minimal (Herbrand) models. Neither of these approaches is suitable in a higher-order setting: Model theory is not a well-developed tool here, and substitutions for predicate variables that can appear in a higher-order logic can take formulas in a restricted form, such as the conjunctive-normal clausal form, into new formulas that no longer adhere to this form. Faced with this situation, we have turned in our work to the sequent calculus of Gentzen. We have found this to be a versatile and flexible tool for understanding and analyzing the metatheory and computational properties of

© in this web service Cambridge University Press www.cambridge.org

CAMBRIDGE

Cambridge University Press
978-0-521-87940-8 - Programming with Higher-Order Logic
Dale Miller and Gopalan Nadathur
Frontmatter
More information

logics. Using it, we have been able to identify logic programming languages as ones that support a particular goal-directed approach to proof search. This viewpoint allows us to extend naturally the Horn clause logic that underlies languages such as Prolog to richer first-order logics that offer support at the programming level to scoping mechanisms. The same approach generalizes to higher-order logic and also to contexts that we do not explicitly treat here, such as linear logic and the dependently typed λ-calculus. Indeed, understanding proof search through the perspective of the sequent calculus seems to be an essential part of grasping the significance of logic as a tool for computing. We accordingly expose this line of thinking as we develop a higher-order logic for programming.

Gaining facility with new ideas in programming usually requires concrete experimentation with them. Many of the ideas that we expose here have an actual realization in the language λProlog. Programs written in λProlog often will be used to illuminate discussions of logic and theoretical principles. These programs can be run using the Teyjus implementation of λProlog that we provide an introduction to in the Appendix. The Teyjus system can be freely downloaded, and the distribution material accompanying it contains many programs, including the ones discussed in this book, that illustrate the special capabilities of λProlog. We anticipate that a reader of this book eventually will have enough expertise to develop his or her own programs in a number of application areas where binding is an important part of syntactic structure.

This book, then, covers three broad topics: a proof search–based view of computation, a higher-order logic–based approach to programming, and a particular language that realizes these ideas. We hope to leave the reader in the end with an appreciation of how higher-order logic may be used to specify computations and with the ability to use a logic programming language based on such a logic to build actual systems. We believe that this kind of background is becoming increasingly useful as demands of the programming process get more sophisticated. One pertinent application area is that where logic and deduction function as "gatekeepers" that ensure the security and integrity of lower-level processes; a specific example of this kind appears within the proof-carrying-code framework that has been proposed as a vehicle for ensuring, for instance, the safety of mobile code. Another application area is the mechanization of the metatheory of logics and languages that is the focus, for example, of the recently posed POPLmark challenge. This book develops a fruitful approach to specifying computations over logical expressions and program phrases, all of which are central to such metatheoretic manipulations. The λProlog language also provides a means for prototyping and implementing such specifications. While we do not discuss this issue significantly in this book, these λProlog specifications should further facilitate rich and interesting new approaches to

© in this web service Cambridge University Press www.cambridge.org

CAMBRIDGE

Cambridge University Press
978-0-521-87940-8 - Programming with Higher-Order Logic
Dale Miller and Gopalan Nadathur
Frontmatter
More information

reasoning about metatheoretic properties of the logics and languages encoded in them.

The material we present here owes a lot to collaborations with colleagues. The foundational ideas relating to logic programming were developed in the late 1980s in interactions with Frank Pfenning and Andre Scedrov. We also received valuable input from Natarajan Shankar during this phase. A Prolog-based implementation of λProlog followed soon after. The understanding we now have of the capabilities of this language derives significantly from the experiments conducted with this system by Amy Felty, Elsa Gunter, John Hannan, Fernando Pereira, Remo Pereschi, and Frank Pfenning, among many other researchers who we surely err in not mentioning explicitly. The language subsequently has received implementations in Common Lisp by Conal Elliott and Frank Pfenning; in C by Pascal Brisset and Olivier Ridoux; and in Standard ML by Conal Elliott, Amy Felty, Dale Miller, Frank Pfenning, and Philip Wickline. Gopalan Nadathur has led a long-term project focused on compiling this language in which Andrew Gacek, Steven Holte, Bharat Jayaraman, Keehang Kwon, Dustin Mitchell, Xiaochu Qi, and Zachary Snow have participated. This work has resulted in two different versions of the Teyjus system. We have received comments and helpful suggestions on this book from Jim Blandy, Iliano Cervesato, Giorgio Delzanno, Joern Dinkla, Zhiping Duan, Daniel Friedman, Andrew Gacek, Clément Houtmann, H. Krishnapriyan, Gary Leavens, Chuck Liang, Jim Lipton, Tong Mei, Catuscia Palamidessi, Olivier Ridoux, Jenny Simon, and Yuting Wang. INRIA has provided support during the writing of this book by facilitating a sabbatical visit by Gopalan Nadathur and through its "Equipes Associées" Slimmer. The National Science Foundation has funded the development of the ideas we present through grants at various points, most recently through the Grants CCF-0429572 and NSF/CCF-0917140. Any opinions, findings, and conclusions or recommendations expressed in this book are those of the authors and do not necessarily reflect the views of the National Science Foundation.

Palaiseau, France *Dale Miller*
Minneapolis, MN, USA *Gopalan Nadathur*
October 2011

© in this web service Cambridge University Press www.cambridge.org

CAMBRIDGE

Cambridge University Press
978-0-521-87940-8 - Programming with Higher-Order Logic
Dale Miller and Gopalan Nadathur
Frontmatter
More information

© in this web service Cambridge University Press www.cambridge.org