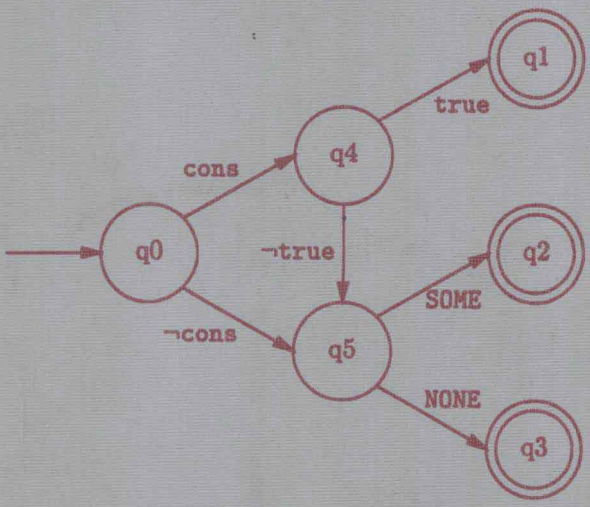
Lecture Notes in Computer Science 1549

Mikael Pettersson

Compiling Natural Semantics



Springer

Mikael Pettersson

Compiling Natural Semantics

Springer

Series Editors

Gerhard Goos, Karlsruhe University, Germany
Juris Hartmanis, Cornell University, NY, USA
Jan van Leeuwen, Utrecht University, The Netherlands

Author

Mikael Pettersson
Computing Science Department
Uppsala University
P.O. Box 311, S-751 05 Uppsala, Sweden
E-mail: Mikael.Pettersson@csd.uu.se

Cataloging-in-Publication data applied for

Die Deutsche Bibliothek - CIP-Einheitsaufnahme

Pettersson, Mikael:
Compiling natural semantics / Mikael Pettersson. - Berlin ;
Heidelberg ; New York ; Barcelona ; Hong Kong ; London ; Milan ;
Paris ; Singapore ; Tokyo : Springer, 1999
(Lecture notes in computer science ; 1549)
ISBN 3-540-65968-4

CR Subject Classification (1998): D.3, F.3

ISSN 0302-9743
ISBN 3-540-65968-4 Springer-Verlag Berlin Heidelberg New York

Typesetting: Camera-ready by author
SPIN 10693148 06/3142 – 5 4 3 2 1 0 Printed on acid-free paper

Lecture Notes in Computer Science 1549

Edited by G. Goos, J. Hartmanis and J. van Leeuwen

Springer
Berlin
Heidelberg
New York
Barcelona
Hong Kong
London
Milan
Paris
Singapore
Tokyo

Foreword

When Mikael asked me to write this foreword, I happily agreed. Since the time I wrote my own Ph.D. thesis 1984 on incremental programming environments, compilers, and generator tools for such systems, I have been interested in the problem of automatically generating practical tools from formal semantics language specifications. This problem was unsolved in my own work at that time, which focused on incremental environment architecture, debugging, and code generation aspects, and could generate parts of incremental programming environments from specifications. However, I always wished for a good opportunity to attack the semantics problem in a practical context.

This opportunity came in the fall of 1989, when Mikael Pettersson started as a graduate student in the PELAB research group (Programming Environments Laboratory) at Linköping University, of which I recently had become leader. Mikael shared my interest in efficient programming language implementations and generator tools, and was very research minded from the start. Already during his undergraduate studies he read many research papers on language implementation techniques and formal semantics, at that time primarily Denotational Semantics, and did several experimental designs. This also became the topic of his first 3 years of research, on the problem of generating fast compilers that emit efficient code from denotational specifications. Mikael developed the DML system (Denotational Meta-Language), which included a subset of Standard ML, concrete syntax notation for pattern matching, and implementation techniques that made generated compilers both execute quickly and emit efficient code.

Around 1990–92, we became increasingly aware of the Natural Semantics specification formalism, which was gaining in popularity and seemed to be rather easy to use, yet provided high abstraction power and good modularity properties. During 1992 my group had just started to cooperate in an Esprit project with Gilles Kahn's group at the Sophia-Antipolis branch of INRIA in France. Kahn proposed the Natural Semantics formalism 1985, and his group had since then developed the Centaur system which includes Typol as the meta-language and tool for Natural Semantics specifications. This system provides a nice interactive environment for prototyping language specifications; however, generated compilers and interpreters execute quite slowly.

Therefore, Mikael, with my support, decided to slightly change his Ph.D. thesis topic. The new goal was to develop techniques and tools for generating very efficient language implementations from Natural Semantics specifications. Generated implementations should be comparable with or better than handwritten ones in performance. This was an ambitious goal that nobody had realized before.

After studying the new area, analyzing possible approaches, and implementation techniques used by logic programming and functional programming languages, Mikael completed the first RML prototype in February 1994. At that time RML was both relational and nondeterministic, similar to most logic programming languages, so the name *Relational Meta-Language* was appropriate. Even the first prototype was rather efficient compared to Typol, but better was to come. Mikael observed that the great majority of language specifications in Natural Semantics are deterministic. Only seldom is nondeterminism really needed, and in those cases specifications can usually be reformulated in a deterministic way. Mikael decided to make RML deterministic to enable further improvements in efficiency. A year later, with the addition of more sophisticated optimizations in the RML compiler, the generated code had improved another factor of five in performance. You can read about all the details in this book.

I am very proud of Mikael's work. His RML system is the first generator tool for Natural Semantics that can produce really efficient implementations. The measured efficiency of generated example implementations seems to be roughly the same as (or sometimes better than) comparable hand implementations in Pascal or C. Another important property is compatibility and modularity. Generated modules are produced in C, and can be readily integrated with existing frontends and backends.

I feel quite enthusiastic about the future prospects of automatically generating practically useful implementations from formal specifications of programming languages, using tools such as RML. Perhaps we will soon reach the point where ease of use and efficiency of the generated result will make it as attractive and common to generate semantic processing parts of translators from Natural Semantics specifications, as is currently the case for generating scanners and parsers using tools such as Lex and Yacc. Only the future will tell.

Linköping, October 1998
Peter Fritzson

Preface

Abstract

Natural Semantics has become a popular tool among programming language researchers. It is used for specifying many aspects of programming languages, including type systems, dynamic semantics, translations between representations, and static analyses. The formalism has so far largely been limited to theoretical applications, due to the absence of practical tools for its implementation. Those who try to use it in applications have had to translate their specifications by hand into existing programming languages, which can be tedious and prone to error. Hence, Natural Semantics is rarely used in applications.

Compiling high-level languages to correct and efficient code is nontrivial, hence implementing compilers is difficult and time-consuming. It has become customary to specify *parts* of compilers using special-purpose specification languages, and to compile these specifications to executable code. While this has simplified the construction of compiler front-ends, and to some extent their back-ends, little is available to help construct those parts that deal with *semantics* and translations between higher-level and lower-level representations. This is especially true for the Natural Semantics formalism.

In this thesis, we introduce the Relational Meta-Language, RML, which is intended as a practical language for natural semantics specifications. Runtime efficiency is a prerequisite if natural semantics is to be generally accepted as a *practical* tool. Hence, the main parts of this thesis deal with the problem of compiling natural semantics, actually RML, to highly efficient code.

We have designed and implemented a compiler, `rml2c`, that translates RML to efficient low-level C code. The compilation phases are described in detail. High-level transformations are applied to reduce the usually enormous amount of nondeterminism present in specifications. The resulting forms are often completely deterministic. Pattern-matching constructs are expanded using a pattern-match compiler, and a translation is made into a continuation-passing style intermediate representation. Intermediate-level CPS optimizations are applied before low-level C code is emitted. A new and efficient technique for mapping *tailcalls* to C has been developed.

We have compared our code with other alternative implementations. Our

benchmarking results show that our code is much faster, sometimes by orders of magnitude. This supports our thesis that the given compilation strategy is suitable for a significant class of specifications.

A natural semantics specification for RML itself is given in the appendix.

Acknowledgements

I thank my thesis supervisor Peter Fritzson for giving me free rein to explore my interests in formal semantics and language implementation technology. I also thank the members of my thesis committee, Reinhard Wilhelm, Isabelle Attali, Tore Risch, and Björn Lisper, for their interest in my work, and my friends and colleagues at the Department of Computer Science at Linköping University. And finally, I thank my family for being there.

Addendum

This book is a revised version of the Ph.D. dissertation I defended in December 1995 at the University of Linköping. The RML system has evolved in several directions since then, and I summarize the main developments here.

In the RML type system, implicit logical variables have been replaced by a polymorphic type `'a lvar` with explicit binding and inspection operators, and the notion of *equality types* has been borrowed from Standard ML.

Top-level declarations are now subject to a dependency analysis and reordering phase before type checking, as in Haskell [129, Section 4.5.11].

More techniques for implementing tailcalls in C have been tested, including one used by two Scheme compilers [63]. However, no real performance improvements have been achieved to date.

The RML compiler has been made much more user-friendly. The type checker now gives accurate and relevant error messages, and a new compiler driver automates the many steps involved in compiling and linking code. Work is underway to support debugging and profiling [139].

Students at Linköping University have used the system to construct compilers for real-world languages, including Java and Modelica [95]. The experience has been positive, but a simplified foreign C code interface, a debugger, and support for more traditional programming are sometimes requested.

The Swedish National Board for Industrial and Technical Development (NUTEK) and the Center for Industrial Information Technology (CENIIT) supported my research at Linköping University. Recent developments were implemented during my postdoc at INRIA Sophia-Antipolis 1997–98, funded by the Swedish Research Council for Engineering Sciences (TFR).

Uppsala, April 1999
Mikael Pettersson

Lecture Notes in Computer Science

For information about Vols. 1–1513
please contact your bookseller or Springer-Verlag

Vol. 1514: K. Ohta, D. Pei (Eds.), Advances in Cryptology – ASIACRYPT'98. Proceedings, 1998. XII, 436 pages. 1998.

Vol. 1515: F. Moreira de Oliveira (Ed.), Advances in Artificial Intelligence. Proceedings, 1998. X, 259 pages. 1998. (Subseries LNAI).

Vol. 1516: W. Ehrenberger (Ed.), Computer Safety, Reliability and Security. Proceedings, 1998. XVI, 392 pages. 1998.

Vol. 1517: J. Hromkovič, O. Sýkora (Eds.), Graph-Theoretic Concepts in Computer Science. Proceedings, 1998. X, 385 pages. 1998.

Vol. 1518: M. Luby, J. Rolim, M. Serna (Eds.), Randomization and Approximation Techniques in Computer Science. Proceedings, 1998. IX, 385 pages. 1998.

1519: T. Ishida (Ed.), Community Computing and Support Systems. VIII, 393 pages. 1998.

Vol. 1520: M. Maher, J.-F. Puget (Eds.), Principles and Practice of Constraint Programming - CP98. Proceedings, 1998. XI, 482 pages. 1998.

Vol. 1521: B. Rovan (Ed.), SOFSEM'98: Theory and Practice of Informatics. Proceedings, 1998. XI, 453 pages. 1998.

Vol. 1522: G. Gopalakrishnan, P. Windley (Eds.), Formal Methods in Computer-Aided Design. Proceedings, 1998. IX, 529 pages. 1998.

Vol. 1524: G.B. Orr, K.-R. Müller (Eds.), Neural Networks: Tricks of the Trade. VI, 432 pages. 1998.

Vol. 1525: D. Aucsmith (Ed.), Information Hiding. Proceedings, 1998. IX, 369 pages. 1998.

Vol. 1526: M. Broy, B. Rumpe (Eds.), Requirements Targeting Software and Systems Engineering. Proceedings, 1997. VIII, 357 pages. 1998.

Vol. 1527: P. Baumgartner, Theory Reasoning in Connection Calculi. IX, 283. 1999. (Subseries LNAI).

Vol. 1528: B. Preneel, V. Rijmen (Eds.), State of the Art in Applied Cryptography. Revised Lectures, 1997. VIII, 395 pages. 1998.

Vol. 1529: D. Farwell, L. Gerber, E. Hovy (Eds.), Machine Translation and the Information Soup. Proceedings, 1998. XIX, 532 pages. 1998. (Subseries LNAI).

Vol. 1530: V. Arvind, R. Ramanujam (Eds.), Foundations of Software Technology and Theoretical Computer Science. XII, 369 pages. 1998.

Vol. 1531: H.-Y. Lee, H. Motoda (Eds.), PRICAI'98: Topics in Artificial Intelligence. XIX, 646 pages. 1998. (Subseries LNAI).

Vol. 1096: T. Schael, Workflow Management Systems for Process Organisations. Second Edition. XII, 229 pages. 1998.

Vol. 1532: S. Arikawa, H. Motoda (Eds.), Discovery Science. Proceedings, 1998. XI, 456 pages. 1998. (Subseries LNAI).

Vol. 1533: K.-Y. Chwa, O.H. Ibarra (Eds.), Algorithms and Computation. Proceedings, 1998. XIII, 478 pages. 1998.

Vol. 1534: J.S. Sichman, R. Conte, N. Gilbert (Eds.), Multi-Agent Systems and Agent-Based Simulation. Proceedings, 1998. VIII, 237 pages. 1998. (Subseries LNAI).

Vol. 1535: S. Ossowski, Co-ordination in Artificial Agent Societies. XV; 221 pages. 1999. (Subseries LNAI).

Vol. 1536: W.-P. de Roever, H. Langmaack, A. Pnueli (Eds.), Compositionality: The Significant Difference. Proceedings, 1997. VIII, 647 pages. 1998.

Vol. 1537: N. Magnenat-Thalmann, D. Thalmann (Eds.), Modelling and Motion Capture Techniques for Virtual Environments. Proceedings, 1998. IX, 273 pages. 1998. (Subseries LNAI).

Vol. 1538: J. Hsiang, A. Ohori (Eds.), Advances in Computing Science – ASIAN'98. Proceedings, 1998. X, 305 pages. 1998.

Vol. 1539: O. Rüthing, Interacting Code Motion Transformations: Their Impact and Their Complexity. XXI,225 pages. 1998.

Vol. 1540: C. Beeri, P. Buneman (Eds.), Database Theory – ICDT'99. Proceedings, 1999. XI, 489 pages. 1999.

Vol. 1541: B. Kågström, J. Dongarra, E. Elmroth, J. Waśniewski (Eds.), Applied Parallel Computing. Proceedings, 1998. XIV, 586 pages. 1998.

Vol. 1542: H.I. Christensen (Ed.), Computer Vision Systems. Proceedings, 1999. XI, 554 pages. 1999.

Vol. 1543: S. Demeyer, J. Bosch (Eds.), Object-Oriented Technology ECOOP'98 Workshop Reader. 1998. XXII, 573 pages. 1998.

Vol. 1544: C. Zhang, D. Lukose (Eds.), Multi-Agent Systems. Proceedings, 1998. VII, 195 pages. 1998. (Subseries LNAI).

Vol. 1545: A. Birk, J. Demiris (Eds.), Learning Robots. Proceedings, 1996. IX, 188 pages. 1998. (Subseries LNAI).

Vol. 1546: B. Möller, J.V. Tucker (Eds.), Prospects for Hardware Foundations. Survey Chapters, 1998. X, 468 pages. 1998.

Vol. 1547: S.H. Whitesides (Ed.), Graph Drawing. Proceedings 1998. XII, 468 pages. 1998.

Vol. 1548: A.M. Haeberer (Ed.), Algebraic Methodology and Software Technology. Proceedings, 1999. XI, 531 pages. 1999.

Vol. 1549: M. Pettersson, Compiling Natural Semantics. XVII, 240 pages. 1999.

Vol. 1550: B. Christianson, B. Crispo, W.S. Harbison, M. Roe (Eds.), Security Protocols. Proceedings, 1998. VIII, 241 pages. 1999.

Vol. 1551: G. Gupta (Ed.), Practical Aspects of Declarative Languages. Proceedings, 1999. VIII, 367 pgages. 1999.

Vol. 1552: Y. Kambayashi, D.L. Lee, E.-P. Lim, M.K. Mohania, Y. Masunaga (Eds.), Advances in Database Technologies. Proceedings, 1998. XIX, 592 pages. 1999.

Vol. 1553: S.F. Andler, J. Hansson (Eds.), Active, Real-Time, and Temporal Database Systems. Proceedings, 1997. VIII, 245 pages. 1998.

Vol. 1554: S. Nishio, F. Kishino (Eds.), Advanced Multimedia Content Processing. Proceedings, 1998. XIV, 454 pages. 1999.

Vol. 1555: J.P. Müller, M.P. Singh, A.S. Rao (Eds.), Intelligent Agents V. Proceedings, 1998. XXIV, 455 pages. 1999. (Subseries LNAI).

Vol. 1556: S. Tavares, H. Meijer (Eds.), Selected Areas in Cryptography. Proceedings, 1998. IX, 377 pages. 1999.

Vol. 1557: P. Zinterhof, M. Vajteršic, A. Uhl (Eds.), Parallel Computation. Proceedings, 1999. XV, 604 pages. 1999.

Vol. 1558: H. J.v.d. Herik, H. Iida (Eds.), Computers and Games. Proceedings, 1998. XVIII, 337 pages. 1999.

Vol. 1559: P. Flener (Ed.), Logic-Based Program Synthesis and Transformation. Proceedings, 1998. X, 331 pages. 1999.

Vol. 1560: K. Imai, Y. Zheng (Eds.), Public Key Cryptography. Proceedings, 1999. IX, 327 pages. 1999.

Vol. 1561: I. Damgård (Ed.), Lectures on Data Security.VII, 250 pages. 1999.

Vol. 1562: C.L. Nehaniv (Ed.), Computation for Metaphors, Analogy, and Agents. X, 389 pages. 1999. (Subseries LNAI).

Vol. 1563: Ch. Meinel, S. Tison (Eds.), STACS 99. Proceedings, 1999. XIV, 582 pages. 1999.

Vol. 1565: P. P. Chen, J. Akoka, H. Kangassalo, B. Thalheim (Eds.), Conceptual Modeling. XXIV, 303 pages. 1999.

Vol. 1567: P. Antsaklis, W. Kohn, M. Lemmon, A. Nerode, S. Sastry (Eds.), Hybrid Systems V. X, 445 pages. 1999.

Vol. 1568: G. Bertrand, M. Couprie, L. Perroton (Eds.), Discrete Geometry for Computer Imagery. Proceedings, 1999. XI, 459 pages. 1999.

Vol. 1569: F.W. Vaandrager, J.H. van Schuppen (Eds.), Hybrid Systems: Computation and Control. Proceedings, 1999. X, 271 pages. 1999.

Vol. 1570: F. Puppe (Ed.), XPS-99: Knowledge-Based Systems. VIII, 227 pages. 1999. (Subseries LNAI).

Vol. 1571: P. Noriega, C. Sierra (Eds.), Agent Mediated Electronic Commerce. Proceedings, 1998. IX, 207 pages. 1999. (Subseries LNAI).

Vol. 1572: P. Fischer, H.U. Simon (Eds.), Computational Learning Theory. Proceedings, 1999. X, 301 pages. 1999. (Subseries LNAI).

Vol. 1574: N. Zhong, L. Zhou (Eds.), Methodologies for Knowledge Discovery and Data Mining. Proceedings, 1999. XV, 533 pages. 1999. (Subseries LNAI).

Vol. 1575: S. Jähnichen (Ed.), Compiler Construction. Proceedings, 1999. X, 301 pages. 1999.

Vol. 1576: S.D. Swierstra (Ed.), Programming Languages and Systems. Proceedings, 1999. X, 307 pages. 1999.

Vol. 1577: J.-P. Finance (Ed.), Fundamental Approaches to Software Engineering. Proceedings, 1999. X, 245 pages. 1999.

Vol. 1578: W. Thomas (Ed.), Foundations of Software Science and Computation Structures. Proceedings, 1999. X, 323 pages. 1999.

Vol. 1579: W.R. Cleaveland (Ed.), Tools and Algorithms for the Construction and Analysis of Systems. Proceedings, 1999. XI, 445 pages. 1999.

Vol. 1580: A. Včkovski, K.E. Brassel, H.-J. Schek (Eds.), Interoperating Geographic Information Systems. Proceedings, 1999. XI, 329 pages. 1999.

Vol. 1581: J.-Y. Girard (Ed.), Typed Lambda Calculi and Applications. Proceedings, 1999. VIII, 397 pages. 1999.

Vol. 1582: A. Lecomte, F. Lamarche, G. Perrier (Eds.), Logical Aspects of Computational Linguistics. Proceedings, 1997. XI, 251 pages. 1999. (Subseries LNAI).

Vol. 1584: G. Gottlob, E. Grandjean, K. Seyr (Eds.), Computer Science Logic. Proceedings, 1998. X, 431 pages. 1999.

Vol. 1586: J. Rolim et al. (Eds.), Parallel and Distributed Processing. Proceedings, 1999. XVII, 1443 pages. 1999.

Vol. 1587: J. Pieprzyk, R. Safavi-Naini, J. Seberry (Eds.), Information Security and Privacy. Proceedings, 1999. XI, 327 pages. 1999.

Vol. 1590: P. Atzeni, A. Mendelzon, G. Mecca (Eds.), The World Wide Web and Databases. Proceedings, 1998. VIII, 213 pages. 1999.

Vol. 1592: J. Stern (Ed.), Advances in Cryptology – EUROCRYPT '99. Proceedings, 1999. XII, 475 pages. 1999.

Vol. 1593: P. Sloot, M. Bubak, A. Hoekstra, B. Hertzberger (Eds.), High-Performance Computing and Networking. Proceedings, 1999. XXIII, 1318 pages. 1999.

Vol. 1594: P. Ciancarini, A.L. Wolf (Eds.), Coordination Languages and Models. Proceedings, 1999. IX, 420 pages. 1999.

Vol. 1596: R. Poli, H.-M. Voigt, S. Cagnoni, D. Corne, G.D. Smith, T.C. Fogarty (Eds.), Evolutionary Image Analysis, Signal Processing and Telecommunications. Proceedings, 1999. X, 225 pages. 1999.

Vol. 1597: H. Zuidweg, M. Campolargo, J. Delgado, A. Mullery (Eds.), Intelligence in Services and Networks. Proceedings, 1999. XII, 552 pages. 1999.

Vol. 1599: T. Ishida (Ed.), Multiagent Platforms. Proceedings, 1998. VIII, 187 pages. 1999. (Subseries LNAI).

Vol. 1602: A. Sivasubramaniam, M. Lauria (Eds.), Network-Based Parallel Computing. Proceedings, 1999. VIII, 225 pages. 1999.

Vol. 1605: J. Billington, M. Diaz, G. Rozenberg (Eds.), Application of Petri Nets to Communication Networks. IX, 303 pages. 1999.

Vol. 1609: Z. W. Raś, A. Skowron (Eds.), Foundations of Intelligent Systems. Proceedings, 1999. XII, 676 pages. 1999. (Subseries LNAI).

Contents