LNAI 3662

Chitta Baral
Gianluigi Greco
Nicola Leone
Giorgio Terracina (Eds.)

Logic Programming and Nonmonotonic Reasoning

8th International Conference, LPNMR 2005
Diamante, Italy, September 2005
Proceedings

Springer

Chitta Baral   Gianluigi Greco
Nicola Leone   Giorgio Terracina (Eds.)

Logic Programming and Nonmonotonic Reasoning

8th International Conference, LPNMR 2005
Diamante, Italy, September 5-8, 2005
Proceedings

Springer

Series Editors

Jaime G. Carbonell, Carnegie Mellon University, Pittsburgh, PA, USA
Jörg Siekmann, University of Saarland, Saarbrücken, Germany

Volume Editors

Chitta Baral
Arizona State University
Department of Computer Science and Engineering
Fulton School of Engineering
Brickyard Suite 572, 699 S. Mill Avenue, Tempe, AZ 85281, USA
E-mail: chitta@asu.edu

Gianluigi Greco
Nicola Leone
Giorgio Terracina
Università della Calabria
Dipartimento di Matematica
via P. Bucci 30B, I-87030 Rende (CS), Italy
E-mail: {ggreco, leone, terracina}@mat.unical.it

Library of Congress Control Number: 2005931127

CR Subject Classification (1998): I.2.3, I.2, F.4.1, D.1.6

ISSN 0302-9743
ISBN-10 3-540-28538-5 Springer Berlin Heidelberg New York
ISBN-13 978-3-540-28538-0 Springer Berlin Heidelberg New York

Springer is a part of Springer Science+Business Media

springeronline.com

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India
Printed on acid-free paper SPIN: 11546207 06/3142 5 4 3 2 1 0

Lecture Notes in Artificial Intelligence 3662

Edited by J. G. Carbonell and J. Siekmann

Subseries of Lecture Notes in Computer Science

Preface

These are the proceedings of the *8th International Conference on Logic Programming and Nonmonotonic Reasoning* (LPNMR 2005). Following the previous ones held in Washington, DC, USA (1991), Lisbon, Portugal (1993), Lexington, KY, USA (1995), Dagstuhl, Germany (1997), El Paso, TX, USA (1999), Vienna, Austria (2001) and Ft. Lauderdale, FL, USA (2004), the eighth conference was held in Diamante, Italy, from 5th to 8th of September 2005.

The aim of the LPNMR conferences is to bring together and facilitate interactions between active researchers interested in all aspects concerning declarative logic programming, nonmonotonic reasoning, knowledge representation, and the design of logic-based systems and database systems. LPNMR strives to encompass theoretical and experimental studies that lead to the implementation of practical systems for declarative programming and knowledge representation.

The technical program of LPNMR 2005 comprised three invited talks that were given by Jürgen Angele, Thomas Eiter and Michael Kifer. All papers presented at the conference and published in these proceedings went through a rigorous review process which selected 25 research papers and 16 papers for the system and application tracks.

Many individuals worked for the success of the conference. Special thanks are due to all members of the Program Committee and to additional reviewers for their efforts to produce fair and thorough evaluations of submitted papers. A special thanks is due to the University of Calabria Organizing Committee which made this event possible. Last, but not least, we thank the sponsoring institutions for their generosity.

June 2005

Chitta Baral and Nicola Leone
Program Co-chairs
LPNMR'05

Organization

LPNMR 2005 was organized by the Department of Mathematics at the University of Calabria, Italy.

Program Co-chairs

Chitta Baral (Arizona State University, USA)
Nicola Leone (University of Calabria, Italy)

Program Committee

Jose Alferes (New University of Lisbon, Portugal)
Leopoldo Bertossi (Carleton University, Canada)
Pedro Cabalar (Corunna University, Spain)
Gerhard Brewka (University of Leipzig, Germany)
Juergen Dix (Technical University of Clausthal, Germany)
Wolfgang Faber (University of Calabria, Italy)
Norman Foo (University of New South Wales, Australia)
Michael Gelfond (Texas Tech University, USA)
Antonis Kakas (University of Cyprus, Cyprus)
Katsumi Inoue (Kobe University, Japan)
Vladimir Lifschitz (University of Texas at Austin, USA)
Fangzhen Lin (Hong Kong University of Science and Technology, China)
Ilkka Niemelä (Helsinki University of Technology, Finland)
David Pearce (Rey Juan Carlos University, Spain)
Alessandro Provetti (University of Messina, Italy)
Francesco Scarcello (University of Calabria, Italy)
Torsten Schaub (University of Potsdam, Germany)
Hans Tompits (Vienna University of Technology, Austria)
Francesca Toni (Imperial College London, UK)
Mirek Truszczynski (University of Kentucky, USA)
Marina de Vos (University of Bath, United Kingdom)

Committee for the System and Application Tracks

Gerald Pfeifer (SUSE/Novell) - [**chair**]
Marcello Balduccini (Texas Tech University)
Yuliya Lierler (Universität Erlangen)
Ilkka Niemelä (Helsinki University of Technology)
Yuting Zhao (Istituto Trentino di Cultura)

Publicity Chair

Gianluigi Greco (University of Calabria, Italy)

Organizing Co-chairs

Giovambattista Ianni (University of Calabria, Italy)
Giorgio Terracina (University of Calabria, Italy)

Organizing Committee

Francesco Calimeri
Mina Catalano
Chiara Cumbo
Tina Dell'Armi
Stefania Galizia
Gianluigi Greco
Giuseppe Ielpa
Vincenzino Lio
Simona Perri
Veronica Policicchio
Francesco Ricca
Paola Sdao

External Referees

João Alcântara
Christian Anger
Marcello Balduccini
Federico Banti
Andrea Bracciali
Martin Brain
Loreto Bravo
Andreas Bruening
Francesco Buccafurri
Stefania Costantini
Carlos Damasio
Yannis Dimopoulos
Phan Minh Dung
Ulle Endriss
Esra Erdem
Paolo Ferraris
Michael Fink
Martin Gebser
Gianluigi Greco
Keijo Heljanko
Koji Iwanuma
Tomi Janhunen
Joohyung Lee
João Leite
Lengning Liu
Massimo Marchi
Victor Marek
Yves Martin
Thomas Meyer
Rob Miller
Emilia Oikarinen
Philip Reiser
Fariba Sadri
Chiaki Sakama
Ken Satoh
Juan Manuel Serrano
Tommi Syrjänen
Giorgio Terracina
Satoshi Tojo
Richard Watson
Gregory Wheeler
Stefan Woltran
Ester Zumpano

Lecture Notes in Artificial Intelligence (LNAI)

Vol. 3662: C. Baral, G. Greco, N. Leone, G. Terracina (Eds.), Logic Programming and Nonmonotonic Reasoning. XIII, 454 pages. 2005.

Vol. 3632: R. Nieuwenhuis (Ed.), Automated Deduction – CADE-20. XIII, 459 pages. 2005.

Vol. 3626: B. Ganter, G. Stumme, R. Wille (Eds.), Formal Concept Analysis. X, 349 pages. 2005.

Vol. 3625: S. Kramer, B. Pfahringer (Eds.), Inductive Logic Programming. XIII, 427 pages. 2005.

Vol. 3620: H. Muñoz-Avila, F. Ricci (Eds.), Case-Based Reasoning Research and Development. XV, 654 pages. 2005.

Vol. 3614: L. Wang, Y. Jin (Eds.), Fuzzy Systems and Knowledge Discovery, Part II. XLI, 1314 pages. 2005.

Vol. 3613: L. Wang, Y. Jin (Eds.), Fuzzy Systems and Knowledge Discovery, Part I. XLI, 1334 pages. 2005.

Vol. 3607: J.-D. Zucker, L. Saitta (Eds.), Abstraction, Reformulation and Approximation. XII, 376 pages. 2005.

Vol. 3596: F. Dau, M.-L. Mugnier, G. Stumme (Eds.), Conceptual Structures: Common Semantics for Sharing Knowledge. XI, 467 pages. 2005.

Vol. 3593: V. Mařík, R. W. Brennan, M. Pěchouček (Eds.), Holonic and Multi-Agent Systems for Manufacturing. XI, 269 pages. 2005.

Vol. 3587: P. Perner, A. Imiya (Eds.), Machine Learning and Data Mining in Pattern Recognition. XVII, 695 pages. 2005.

Vol. 3584: X. Li, S. Wang, Z.Y. Dong (Eds.), Advanced Data Mining and Applications. XIX, 835 pages. 2005.

Vol. 3581: S. Miksch, J. Hunter, E. Keravnou (Eds.), Artificial Intelligence in Medicine. XVII, 547 pages. 2005.

Vol. 3577: R. Falcone, S. Barber, J. Sabater-Mir, M.P. Singh (Eds.), Trusting Agents for Trusting Electronic Societies. VIII, 235 pages. 2005.

Vol. 3575: S. Wermter, G. Palm, M. Elshaw (Eds.), Biomimetic Neural Learning for Intelligent Robots. IX, 383 pages. 2005.

Vol. 3571: L. Godo (Ed.), Symbolic and Quantitative Approaches to Reasoning with Uncertainty. XVI, 1028 pages. 2005.

Vol. 3559: P. Auer, R. Meir (Eds.), Learning Theory. XI, 692 pages. 2005.

Vol. 3558: V. Torra, Y. Narukawa, S. Miyamoto (Eds.), Modeling Decisions for Artificial Intelligence. XII, 470 pages. 2005.

Vol. 3554: A. Dey, B. Kokinov, D. Leake, R. Turner (Eds.), Modeling and Using Context. XIV, 572 pages. 2005.

Vol. 3539: K. Morik, J.-F. Boulicaut, A. Siebes (Eds.), Local Pattern Detection. XI, 233 pages. 2005.

Vol. 3538: L. Ardissono, P. Brna, A. Mitrovic (Eds.), User Modeling 2005. XVI, 533 pages. 2005.

Vol. 3533: M. Ali, F. Esposito (Eds.), Innovations in Applied Artificial Intelligence. XX, 858 pages. 2005.

Vol. 3528: P.S. Szczepaniak, J. Kacprzyk, A. Niewiadomski (Eds.), Advances in Web Intelligence. XVII, 513 pages. 2005.

Vol. 3518: T.B. Ho, D. Cheung, H. Liu (Eds.), Advances in Knowledge Discovery and Data Mining. XXI, 864 pages. 2005.

Vol. 3508: P. Bresciani, P. Giorgini, B. Henderson-Sellers, G. Low, M. Winikoff (Eds.), Agent-Oriented Information Systems II. X, 227 pages. 2005.

Vol. 3505: V. Gorodetsky, J. Liu, V. A. Skormin (Eds.), Autonomous Intelligent Systems: Agents and Data Mining. XIII, 303 pages. 2005.

Vol. 3501: B. Kégl, G. Lapalme (Eds.), Advances in Artificial Intelligence. XV, 458 pages. 2005.

Vol. 3492: P. Blache, E. Stabler, J. Busquets, R. Moot (Eds.), Logical Aspects of Computational Linguistics. X, 363 pages. 2005.

Vol. 3488: M.-S. Hacid, N.V. Murray, Z.W. Raś, S. Tsumoto (Eds.), Foundations of Intelligent Systems. XIII, 700 pages. 2005.

Vol. 3487: J. Leite, P. Torroni (Eds.), Computational Logic in Multi-Agent Systems. XII, 281 pages. 2005.

Vol. 3476: J. Leite, A. Omicini, P. Torroni, P. Yolum (Eds.), Declarative Agent Languages and Technologies II. XII, 289 pages. 2005.

Vol. 3464: S.A. Brueckner, G.D.M. Serugendo, A. Karageorgos, R. Nagpal (Eds.), Engineering Self-Organising Systems. XIII, 299 pages. 2005.

Vol. 3452: F. Baader, A. Voronkov (Eds.), Logic for Programming, Artificial Intelligence, and Reasoning. XI, 562 pages. 2005.

Vol. 3451: M.-P. Gleizes, A. Omicini, F. Zambonelli (Eds.), Engineering Societies in the Agents World V. XIII, 349 pages. 2005.

Vol. 3446: T. Ishida, L. Gasser, H. Nakashima (Eds.), Massively Multi-Agent Systems I. XI, 349 pages. 2005.

Vol. 3445: G. Chollet, A. Esposito, M. Faundez-Zanuy, M. Marinaro (Eds.), Nonlinear Speech Modeling and Applications. XIII, 433 pages. 2005.

Vol. 3438: H. Christiansen, P.R. Skadhauge, J. Villadsen (Eds.), Constraint Solving and Language Processing. VIII, 205 pages. 2005.

Vol. 3430: S. Tsumoto, T. Yamaguchi, M. Numao, H. Motoda (Eds.), Active Mining. XII, 349 pages. 2005.

Lecture Notes in Artificial Intelligence (LNAI)

Vol. 3662: C. Baral, G. Greco, N. Leone, G. Terracina (Eds.), Logic Programming and Nonmonotonic Reasoning. XIII, 454 pages. 2005.

Vol. 3632: R. Nieuwenhuis (Ed.), Automated Deduction - CADE-20. XIII, 459 pages. 2005.

Vol. 3626: B. Ganter, G. Stumme, R. Wille (Eds.), Formal Concept Analysis. X, 349 pages. 2005.

Vol. 3625: S. Kramer, B. Pfahringer (Eds.), Inductive Logic Programming. XIII, 427 pages. 2005.

Vol. 3620: H. Muñoz-Avila, F. Ricci (Eds.), Case-Based Reasoning Research and Development. XV, 654 pages. 2005.

Vol. 3614: L. Wang, Y. Jin (Eds.), Fuzzy Systems and Knowledge Discovery, Part II. XLI, 1314 pages. 2005.

Vol. 3613: L. Wang, Y. Jin (Eds.), Fuzzy Systems and Knowledge Discovery, Part I. XLI, 1334 pages. 2005.

Vol. 3607: J.-D. Zucker, L. Saitta (Eds.), Abstraction, Reformulation and Approximation. XII, 376 pages. 2005.

Vol. 3596: F. Dau, M.-L. Mugnier, G. Stumme (Eds.), Conceptual Structures: Common Semantics for Sharing Knowledge. XI, 467 pages. 2005.

Vol. 3593: V. Mařík, R. W. Brennan, M. Pěchouček (Eds.), Holonic and Multi-Agent Systems for Manufacturing. XI, 269 pages. 2005.

Vol. 3587: P. Perner, A. Imiya (Eds.), Machine Learning and Data Mining in Pattern Recognition. XVII, 695 pages. 2005.

Vol. 3584: X. Li, S. Wang, Z.Y. Dong (Eds.), Advanced Data Mining and Applications. XIX, 835 pages. 2005.

Vol. 3581: S. Miksch, J. Hunter, E. Keravnou (Eds.), Artificial Intelligence in Medicine. XVII, 547 pages. 2005.

Vol. 3577: R. Falcone, S. Barber, J. Sabater-Mir, M.P. Singh (Eds.), Trusting Agents for Trusting Electronic Societies. VIII, 235 pages. 2005.

Vol. 3575: S. Wermter, G. Palm, M. Elshaw (Eds.), Biomimetic Neural Learning for Intelligent Robots. IX, 383 pages. 2005.

Vol. 3571: L. Godo (Ed.), Symbolic and Quantitative Approaches to Reasoning with Uncertainty. XVI, 1028 pages. 2005.

Vol. 3559: P. Auer, R. Meir (Eds.), Learning Theory. XI, 692 pages. 2005.

Vol. 3558: V. Torra, Y. Narukawa, S. Miyamoto (Eds.), Modeling Decisions for Artificial Intelligence. XII, 470 pages. 2005.

Vol. 3554: A. Dey, B. Kokinov, D. Leake, R. Turner (Eds.), Modeling and Using Context. XIV, 572 pages. 2005.

Vol. 3539: K. Morik, J.-F. Boulicaut, A. Siebes (Eds.), Local Pattern Detection. XI, 233 pages. 2005.

Vol. 3538: L. Ardissono, P. Brna, A. Mitrovic (Eds.), User Modeling 2005. XVI, 533 pages. 2005.

Vol. 3533: M. Ali, F. Esposito (Eds.), Innovations in Applied Artificial Intelligence. XX, 858 pages. 2005.

Vol. 3528: P.S. Szczepaniak, J. Kacprzyk, A. Niewiadomski (Eds.), Advances in Web Intelligence. XVII, 513 pages. 2005.

Vol. 3518: T.B. Ho, D. Cheung, H. Liu (Eds.), Advances in Knowledge Discovery and Data Mining. XXI, 864 pages. 2005.

Vol. 3508: P. Bresciani, P. Giorgini, B. Henderson-Sellers, G. Low, M. Winikoff (Eds.), Agent-Oriented Information Systems II. X, 227 pages. 2005.

Vol. 3505: V. Gorodetsky, J. Liu, V. A. Skormin (Eds.), Autonomous Intelligent Systems: Agents and Data Mining. XIII, 303 pages. 2005.

Vol. 3501: B. Kégl, G. Lapalme (Eds.), Advances in Artificial Intelligence. XV, 458 pages. 2005.

Vol. 3492: P. Blache, E. Stabler, J. Busquets, R. Moot (Eds.), Logical Aspects of Computational Linguistics. X, 363 pages. 2005.

Vol. 3488: M.-S. Hacid, N.V. Murray, Z.W. Raś, S. Tsumoto (Eds.), Foundations of Intelligent Systems. XIII, 700 pages. 2005.

Vol. 3487: J. Leite, P. Torroni (Eds.), Computational Logic in Multi-Agent Systems. XII, 281 pages. 2005.

Vol. 3476: J. Leite, A. Omicini, P. Torroni, P. Yolum (Eds.), Declarative Agent Languages and Technologies II. XII, 289 pages. 2005.

Vol. 3464: S.A. Brueckner, G.D.M. Serugendo, A. Karageorgos, R. Nagpal (Eds.), Engineering Self-Organising Systems. XIII, 299 pages. 2005.

Vol. 3452: F. Baader, A. Voronkov (Eds.), Logic for Programming, Artificial Intelligence, and Reasoning. XI, 562 pages. 2005.

Vol. 3451: M.-P. Gleizes, A. Omicini, F. Zambonelli (Eds.), Engineering Societies in the Agents World V. XIII, 349 pages. 2005.

Vol. 3446: T. Ishida, L. Gasser, H. Nakashima (Eds.), Massively Multi-Agent Systems I. XI, 349 pages. 2005.

Vol. 3445: G. Chollet, A. Esposito, M. Faundez-Zanuy, M. Marinaro (Eds.), Nonlinear Speech Modeling and Applications. XIII, 433 pages. 2005.

Vol. 3438: H. Christiansen, P.R. Skadhauge, J. Villadsen (Eds.), Constraint Solving and Language Processing. VIII, 205 pages. 2005.

Vol. 3430: S. Tsumoto, T. Yamaguchi, M. Numao, H. Motoda (Eds.), Active Mining. XII, 349 pages. 2005.

¥566.40元

Table of Contents

Invited Papers

ASP Foundations

ASP Extensions

Applications

Actions and Causations

Algorithms and Computation

Foundations

Semantics

Application Track

System Track

Nonmonotonic Reasoning in FLORA-2*

Michael Kifer

Department of Computer Science,
State University of New Your at Stony Brook,
Stong Brook, NY 11794, USA
kifer@cs.stonybrook.edu

Abstract. FLORA-2 is an advanced knowledge representation system that integrates F-logic, HiLog, and Transaction Logic. In this paper we give an overview of the theoretical foundations of the system and of some of the aspects of nonmonotonic reasoning in FLORA-2. These include scoped default negation, behavioral inheritance, and nonmonotonicity that stems from database dynamics.

1 Introduction

FLORA-2 is a knowledge base engine and a complete environment for developing knowledge-intensive applications. It integrates F-logic with other novel formalisms such as HiLog and Transaction Logic. FLORA-2 is freely available on the Internet[1] and is in use by a growing research community. Many of the features of FLORA-2 have been adopted by the recently proposed languages in the Semantic Web Services domain: WSML-Rule[2] and SWSL-Rules.[3]

One of the main foundational ingredients of FLORA-2, F-logic [20], extends classical predicate calculus with the concepts of objects, classes, and types, which are adapted from object-oriented programming. In this way, F-logic integrates the paradigms of logic programming and deductive databases with the object-oriented programming paradigm. Most of the applications of F-logic have been in intelligent information systems, but more recently it has been used to represent ontologies and other forms of Semantic Web reasoning [14,12,27,1,11,2,19].

HiLog [8] is an extension of the standard predicate calculus with higher-order syntax. Yet the semantics of HiLog remains first-order and tractable. In FLORA-2, HiLog is the basis for simple and natural querying of term structures and for reification (or objectification) of logical statements, which is an important requirement for a Semantic Web language. Transaction Logic [6] provides the basis for declarative programming of "procedural knowledge" that is often embedded in intelligent agents or Semantic Web services.

In this paper we first survey the main features of FLORA-2 and then discuss three forms of nonmonotonic reasoning provided by the system.

---

* This work was supported in part by NSF grant CCR-0311512 and by U.S. Army Medical Research Institute under a subcontract through Brookhaven National Lab.

1 http://flora.sourceforge.net/

2 http://www.w3.org/Submission/WSML/

3 http://www.daml.org/services/swsl-rules/1.0/

C. Baral et al. (Eds.): LPNMR 2005, LNAI 3662, pp. 1–12, 2005.

2 Overview of F-Logic

F-logic extends predicate calculus both syntactically and semantically. It has a monotonic logical entailment relationship, and its proof theory is sound and complete with respect to the semantics. F-logic comes in two flavors: the first-order flavor and the logic programming flavor. The first-order flavor of F-logic can be viewed as a syntactic variant of classical logic [20]. The logic programming flavor uses a subset of the syntax of F-logic, but gives it a different, non-first-order semantics by interpreting the negation operator as negation-as-failure.

The relationship between the first-order variant of F-logic and its logic programming variant is similar to the relationship between predicate calculus and standard logic programming [23]: object-oriented logic programming is built on the rule-based subset of F-logic by adding the appropriate non-monotonic extensions [32,33,24]. These extensions are intended to capture the semantics of negation-as-failure (like in standard logic programming [28]) and the semantics of multiple inheritance with overriding (which is not found in standard logic programming).

F-logic uses first-order variable-free terms to represent *object identity* (abbr., OID); for instance, `John` and `father(Mary)` are possible Ids of objects. Objects can have attributes. For instance,

```
Mary[spouse->John, children->{Alice,Nancy}].
Mary[children->Jack].
```

Such formulas are called F-logic *molecules*. The first formula says that object `Mary` has an attribute `spouse` whose value is the OID `John`. It also says that the attribute `children` is set-valued and its value is a set that *contains* two OIDs: `Alice` and `Nancy`. We emphasize "contains" because sets do not need to be specified all at once. For instance, the second formula above says that `Mary` has an additional child, `Jack`.

In earlier versions of F-logic, set-valued attributes were denoted with `->>` instead of `->`. However, subsequently the syntax was modernized and simplified. Instead of using different arrows, cardinality constraints (to be explained shortly) were introduced to indicate that an attribute is single-valued.

While some attributes of an object are specified explicitly, as facts, other attributes can be defined using deductive rules. For instance, we can derive `John[children->{Alice,Nancy,Jack}]` using the following deductive rule:

```
?X[children->{?C}] :- ?Y[spouse->?X, children->{?C}].
```

In the new and simplified syntax, alphanumeric symbols prefixed with the ?-sign denote variables and unprefixed alphanumeric symbols denote constants (i.e., OIDs). The earlier versions of FLORA-2 used Prolog conventions where variables were capitalized alphanumeric symbols.

F-logic objects can also have *methods*, which are functions that take arguments. For instance,

```
John[grade(cs305,fall2004)->100, courses(fall2004)->{cs305,cs306}].
```

says that John has a method, grade, whose value on the arguments cs305 (a course identifier) and fall2004 (a semester designation) is 100; it also has a set-valued method courses, whose value on the argument fall2004 is a set of OIDs that contains course identifiers cs305 and cs306. Like attributes, methods can be defined using deductive rules.

The F-logic syntax for *class membership* is John:student and for *subclass relationship* it is student::person. Classes are treated as objects and it is possible for the same object to play the role of a class in one formula and of an object in another. For instance, in the formula student:class, the symbol student plays the role of an object, while in student::person it appears in the role of a class.

F-logic also provides means for specifying schema information through *signature* formulas. For instance, person[spouse {0:1}=>person, name {0:1}=> string, child=> person] is a signature formula that says that class person has three attributes: single-valued attributes spouse and name (single-valuedness is indicated by the cardinality constraint 0:1) and a set-valued attribute child. It further says that the first attribute returns objects of type person, the second of type string, and the last returns sets of objects such that each object in the set is of type person.

3 HiLog and Meta-information

F-logic provides simple and natural means for exploring the structure of object data. Both the schema information associated with classes and the structure of individual objects can be queried by simply putting variables in the appropriate syntactic positions. For instance, to find the set-valued methods that are defined in the *schema* of class student and return objects of type person, one can ask the following query:

```
?- student[?M=>person].
```

The next query is about the type of the results of the attribute name in class student. This query also returns all the superclasses of class student.

```
?- student::?C and student[name=>?T].
```

The above are schema-level meta-queries: they involve the subclass relationship and the type information. One can also pose meta-queries that involve object data (rather than schema). The following queries return the methods that have a known value for the object John:

```
?- John[?Meth->?SomeValue].
```

However, the meta-query facilities of F-logic are not complete. For instance, there is no way in such queries to separate method names from their arguments. Thus, if we had a fact of the form

```
John[age(2005)-> 20].
```