

Michael Moortgat (Ed.)

LNAI 2014

Logical Aspects of Computational Linguistics

**Third International Conference, LACL'98
Grenoble, France, December 1998
Selected Papers**



Springer

7-53
141
998

Michael Moortgat (Ed.)

Logical Aspects of Computational Linguistics

Third International Conference, LACL'98
Grenoble, France, December 14-16, 1998
Selected Papers



E200401781



Springer

Series Editors

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

Volume Editor

Michael Moortgat
Utrecht Institute of Linguistics OTS
Trans 10, 3512 JK Utrecht, The Netherlands
E-mail: Michael.Moortgat@let.uu.nl

Cataloging-in-Publication Data applied for

Die Deutsche Bibliothek - CIP-Einheitsaufnahme

Logical aspects of computational linguistics : third international conference ; selected papers / LACL '98, Grenoble, France, December 14 - 16, 1998. Michael Moortgat (ed.). - Berlin ; Heidelberg ; New York ; Barcelona ; Hong Kong ; London ; Milan ; Paris ; Singapore ; Tokyo : Springer, 2001 (Lecture notes in computer science ; Vol. 2014 : Lecture notes in artificial intelligence)
ISBN 3-540-42251-X

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

ISBN 3-540-42251-X Springer-Verlag Berlin Heidelberg New York

This work is subject to copyright. All rights are reserved, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, re-use of illustrations, recitation, broadcasting, reproduction on microfilms or in any other way, and storage in data banks. Duplication of this publication or parts thereof is permitted only under the provisions of the German Copyright Law of September 9, 1965, in its current version, and permission for use must always be obtained from Springer-Verlag. Violations are liable for prosecution under the German Copyright Law.

Springer-Verlag Berlin Heidelberg New York
a member of BertelsmannSpringer Science+Business Media GmbH

<http://www.springer.de>

© Springer-Verlag Berlin Heidelberg 2001
Printed in Germany

Typesetting: Camera-ready by author, data conversion by PTP-Berlin, Stefan Sossna
Printed on acid-free paper SPIN: 10782159 06/3142 5 4 3 2 1 0

Lecture Notes in Artificial Intelligence 2014

Subseries of Lecture Notes in Computer Science

Edited by J. G. Carbonell and J. Siekmann

Lecture Notes in Computer Science

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

Springer

Berlin

Heidelberg

New York

Barcelona

Hong Kong

London

Milan

Paris

Singapore

Tokyo

Preface

The conference series Logical Aspects of Computational Linguistics (LACL) aims at providing a forum for the presentation and discussion of current research in all the formal and logical aspects of computational linguistics. The LACL initiative started with a workshop held in Nancy (France) in 1995. Selected papers from this event have appeared as a special issue of the *Journal of Logic Language and Information*, Volume 7(4), 1998. In 1996, LACL shifted to the format of an international conference. LACL'96 and '97 were both held in Nancy (France). The proceedings appeared as volumes 1328 and 1582 of the Springer Lecture Notes in Artificial Intelligence.

This volume contains selected papers of the third international conference on Logical Aspects of Computational Linguistics (LACL'98), held in Grenoble, France, from December 14 to 16, 1998. The conference was organized by the University Pierre Mendès-France (Grenoble 2) together with LORIA (Laboratoire Lorrain d'Informatique et Applications, Nancy). On the basis of 33 submitted 4-page abstracts, the Program Committee selected 19 contributions for presentation. In addition to the selected papers, the program featured three invited talks, by Maarten de Rijke (ILLC, Amsterdam), Makoto Kanazawa (Chiba University, Japan), and Fernando Pereira (AT&T Labs). After the conference, the contributors were invited to submit a full paper for the conference proceedings.

I thank the members of the Program Committee for their conscientious work in refereeing the submitted abstracts and the full papers, and the authors of the invited papers and the regular presentations for their stimulating contributions to the conference. Special thanks go to the Organizing Committee, in particular Alain Lecomte, whose efficiency contributed greatly to the success of LACL'98.

April 2001

Michael Moortgat

Organization

Program Committee

Chair Michael Moortgat (OTS Utrecht)
Johan van Benthem (ILLC Amsterdam)
Gosse Bouma (Groningen)
Vijay Shanker (Delaware)
Erhard Hinrichs (Tübingen)
Mary Dalrymple (Xerox, Palo Alto)
Ruy de Queiroz (Recife)
François Lamarche (LORIA Nancy)
Christian Retoré (IRISA Rennes)
Uwe Reyle (IMS Stuttgart)

Organizing Committee

Chair Alain Lecomte (Université Grenoble 2 & Project Calligramme, LORIA)
Denis Vernant (Groupe de Recherches Philosophie, Langage et Cognition)
Jean-Michel Adam (Département Informatique & Mathématiques)

The local Organizing Committee thanks Catherine Finkel (Université Pierre Mendès-France), Claude Jeannin and Amélie de Paoli (UFR Sciences de l'Homme et de la Société) for their logistic support during the workshop, and François Lamarche (LORIA) and Christian Retoré (IRISA) for passing on their experience gained during the organization of previous LACL conferences.

Sponsoring Institutions

La Ville de Grenoble
France Télécom – CNET
Xerox Research Center Europe
Université Pierre Mendès-France Grenoble 2
INRIA Rhône-Alpes

Lecture Notes in Artificial Intelligence (LNAI)

- Vol. 1865: K.R. Apt, A.C. Kakas, E. Monfroy, F. Rossi (Eds.), *New Trends Constraints*. Proceedings, 1999. X, 339 pages. 2000.
- Vol. 1866: J. Cussens, A. Frisch (Eds.), *Inductive Logic Programming*. Proceedings, 2000. X, 265 pages. 2000.
- Vol. 1867: B. Ganter, G.W. Mineau (Eds.), *Conceptual Structures: Logical, Linguistic, and Computational Issues*. Proceedings, 2000. XI, 569 pages. 2000.
- Vol. 1881: C. Zhang, V.-W. So (Eds.), *Design and Applications of Intelligent Agents*. Proceedings, 2000. X, 183 pages. 2000.
- Vol. 1886: R. Mizoguchi, J. Slaney (Eds.), *PRICAI 2000: Topics in Artificial Intelligence*. Proceedings, 2000. XX, 835 pages. 2000.
- Vol. 1898: E. Blanzieri, L. Portinale (Eds.), *Advances in Case-Based Reasoning*. Proceedings, 2000. XII, 530 pages. 2000.
- Vol. 1889: M. Anderson, P. Cheng, V. Haarslev (Eds.), *Theory and Application of Diagrams*. Proceedings, 2000. XII, 504 pages. 2000.
- Vol. 1891: A.L. Oliveira (Ed.), *Grammatical Inference: Algorithms and Applications*. Proceedings, 2000. VIII, 313 pages. 2000.
- Vol. 1902: P. Sojka, I. Kopeček, K. Pala (Eds.), *Text, Speech and Dialogue*. Proceedings, 2000. XIII, 463 pages. 2000.
- Vol. 1904: S.A. Cerri, D. Dochev (Eds.), *Artificial Intelligence: Methodology, Systems, and Applications*. Proceedings, 2000. XII, 366 pages. 2000.
- Vol. 1910: D.A. Zighed, J. Komorowski, J. Żytkow (Eds.), *Principles of Data Mining and Knowledge Discovery*. Proceedings, 2000. XV, 701 pages. 2000.
- Vol. 1916: F. Dignum, M. Greaves (Eds.), *Issues in Agent Communication*. X, 351 pages. 2000.
- Vol. 1919: M. Ojeda-Aciego, I.P. de Guzman, G. Brewka, L. Moniz Pereira (Eds.), *Logics in Artificial Intelligence*. Proceedings, 2000. XI, 407 pages. 2000.
- Vol. 1925: J. Cussens, S. Džeroski (Eds.), *Learning Language in Logic*. X, 301 pages. 2000.
- Vol. 1930: J.A. Campbell, E. Roanes-Lozano (Eds.), *Artificial Intelligence and Symbolic Computation*. Proceedings, 2000. X, 253 pages. 2001.
- Vol. 1932: Z.W. Raś, S. Ohsuga (Eds.), *Foundations of Intelligent Systems*. Proceedings, 2000. XII, 646 pages.
- Vol. 1934: J.S. White (Ed.), *Envisioning Machine Translation in the Information Future*. Proceedings, 2000. XV, 254 pages. 2000.
- Vol. 1937: R. Dieng, O. Corby (Eds.), *Knowledge Engineering and Knowledge Management*. Proceedings, 2000. XIII, 457 pages. 2000.
- Vol. 1952: M.C. Monard, J. Simão Sichman (Eds.), *Advances in Artificial Intelligence*. Proceedings, 2000. XV, 498 pages. 2000.
- Vol. 1955: M. Parigot, A. Voronkov (Eds.), *Logic for Programming and Automated Reasoning*. Proceedings, 2000. XIII, 487 pages. 2000.
- Vol. 1967: S. Arikawa, S. Morishita (Eds.), *Discovery Science*. Proceedings, 2000. XII, 332 pages. 2000.
- Vol. 1968: H. Arimura, S. Jain, A. Sharma (Eds.), *Algorithmic Learning Theory*. Proceedings, 2000. XI, 335 pages. 2000.
- Vol. 1972: A. Omicini, R. Tolksdorf, F. Zambonelli (Eds.), *Engineering Societies in the Agents World*. Proceedings, 2000. IX, 143 pages. 2000.
- Vol. 1979: S. Moss, P. Davidsson (Eds.), *Multi-Agent-Based Simulation*. Proceedings, 2000. VIII, 267 pages. 2001.
- Vol. 1991: F. Dignum, C. Sierra (Eds.), *Agent Mediated Electronic Commerce*. VIII, 241 pages. 2001.
- Vol. 1994: J. Lind, *Iterative Software Engineering for Multiagent Systems*. XVII, 286 pages. 2001.
- Vol. 2003: F. Dignum, U. Cortés (Eds.), *Agent-Mediated Electronic Commerce III*. XII, 193 pages. 2001.
- Vol. 2007: J.F. Roddick, K. Hornsby (Eds.), *Temporal, Spatial, and Spatio-Temporal Data Mining*. Proceedings, 2000. VII, 165 pages. 2001.
- Vol. 2014: M. Moortgat (Ed.), *Logical Aspects of Computational Linguistics*. Proceedings, 1998. X, 287 pages. 2001.
- Vol. 2019: P. Stone, T. Balch, G. Kraetzschmar (Eds.), *RoboCup 2000: Robot Soccer World Cup IV*. XVII, 658 pages. 2001.
- Vol. 2033: J. Liu, Y. Ye (Eds.), *E-Commerce Agents*. VI, 347 pages. 2001.
- Vol. 2035: D. Cheung, G.J. Williams, Q. Li (Eds.), *Advances in Knowledge Discovery and Data Mining – PAKDD 2001*. Proceedings, 2001. XVIII, 596 pages. 2001.
- Vol. 2039: M. Schumacher, *Objective Coordination in Multi-Agent System Engineering*. XIV, 149 pages. 2001.
- Vol. 2056: E. Stroulia, S. Matwin (Eds.), *Advances in Artificial Intelligence*. Proceedings, 2001. XII, 366 pages. 2001.
- Vol. 2062: A. Nareyek, *Constraint-Based Agents*. XIV, 178 pages. 2001.
- Vol. 2070: L. Monostori, J. Váncza, M. Ali (Eds.), *Engineering of Intelligent Systems*. Proceedings, 2001. XVIII, 951 pages. 2001.

Lecture Notes in Computer Science

- Vol. 2033: J. Liu, Y. Ye (Eds.), E-Commerce Agents. VI, 347 pages. 2001. (Subseries LNAI).
- Vol. 2034: M.D. Di Benedetto, A. Sangiovanni-Vincentelli (Eds.), Hybrid Systems: Computation and Control. Proceedings, 2001. XIV, 516 pages. 2001.
- Vol. 2035: D. Cheung, G.J. Williams, Q. Li (Eds.), Advances in Knowledge Discovery and Data Mining – PAKDD 2001. Proceedings, 2001. XVIII, 596 pages. 2001. (Subseries LNAI).
- Vol. 2037: E.J.W. Boers et al. (Eds.), Applications of Evolutionary Computing. Proceedings, 2001. XIII, 516 pages. 2001.
- Vol. 2038: J. Miller, M. Tomassini, P.L. Lanzi, C. Ryan, A.G.B. Tettamanzi, W.B. Langdon (Eds.), Genetic Programming. Proceedings, 2001. XI, 384 pages. 2001.
- Vol. 2039: M. Schumacher, Objective Coordination in Multi-Agent System Engineering. XIV, 149 pages. 2001. (Subseries LNAI).
- Vol. 2040: W. Kou, Y. Yesha, C.J. Tan (Eds.), Electronic Commerce Technologies. Proceedings, 2001. X, 187 pages. 2001.
- Vol. 2041: I. Attali, T. Jensen (Eds.), Java on Smart Cards: Programming and Security. Proceedings, 2000. X, 163 pages. 2001.
- Vol. 2042: K.-K. Lau (Ed.), Logic Based Program Synthesis and Transformation. Proceedings, 2000. VIII, 183 pages. 2001.
- Vol. 2043: D. Craeynest, A. Strohmeier (Eds.), Reliable Software Technologies – Ada-Europe 2001. Proceedings, 2001. XV, 405 pages. 2001.
- Vol. 2044: S. Abramsky (Ed.), Typed Lambda Calculi and Applications. Proceedings, 2001. XI, 431 pages. 2001.
- Vol. 2045: B. Pfitzmann (Ed.), Advances in Cryptology – EUROCRYPT 2001. Proceedings, 2001. XII, 545 pages. 2001.
- Vol. 2047: R. Dumke, C. Rautenstrauch, A. Schmietenendorf, A. Scholz (Eds.), Performance Engineering. XIV, 349 pages. 2001.
- Vol. 2048: J. Pauli, Learning Based Robot Vision. IX, 288 pages. 2001.
- Vol. 2051: A. Middeldorp (Ed.), Rewriting Techniques and Applications. Proceedings, 2001. XII, 363 pages. 2001.
- Vol. 2052: V.I. Gorodetski, V.A. Skormin, L.J. Popyack (Eds.), Information Assurance in Computer Networks. Proceedings, 2001. XIII, 313 pages. 2001.
- Vol. 2053: O. Danvy, A. Filinski (Eds.), Programs as Data Objects. Proceedings, 2001. VIII, 279 pages. 2001.
- Vol. 2054: A. Condon, G. Rozenberg (Eds.), DNA Computing. Proceedings, 2000. X, 271 pages. 2001.
- Vol. 2055: M. Margenstern, Y. Rogozhin (Eds.), Machines, Computations, and Universality. Proceedings, 2001. VIII, 321 pages. 2001.
- Vol. 2056: E. Stroulia, S. Matwin (Eds.), Advances in Artificial Intelligence. Proceedings, 2001. XII, 366 pages. 2001. (Subseries LNAI).
- Vol. 2057: M. Dwyer (Ed.), Model Checking Software. Proceedings, 2001. X, 313 pages. 2001.
- Vol. 2059: C. Arcelli, L.P. Cordella, G. Sanniti di Baja (Eds.), Visual Form 2001. Proceedings, 2001. XIV, 799 pages. 2001.
- Vol. 2062: A. Nareyek, Constraint-Based Agents. XIV, 178 pages. 2001. (Subseries LNAI).
- Vol. 2064: J. Blanck, V. Brattka, P. Hertling (Eds.), Computability and Complexity in Analysis. Proceedings, 2000. VIII, 395 pages. 2001.
- Vol. 2066: O. Gascuel, M.-F. Sagot (Eds.), Computational Biology. Proceedings, 2000. X, 165 pages. 2001.
- Vol. 2068: K.R. Dittrich, A. Geppert, M.C. Norrie (Eds.), Advanced Information Systems Engineering. Proceedings, 2001. XII, 484 pages. 2001.
- Vol. 2070: L. Monostori, J. Vánca, M. Ali (Eds.), Engineering of Intelligent Systems. Proceedings, 2001. XVIII, 951 pages. 2001. (Subseries LNAI).
- Vol. 2072: J. Lindskov Knudsen (Ed.), ECOOP 2001 – Object-Oriented Programming. Proceedings, 2001. XIII, 429 pages. 2001.
- Vol. 2073: V.N. Alexandrov, J.J. Dongarra, B.A. Julianio, R.S. Renner, C.J.K. Tan (Eds.), Computational Science – ICCS 2001. Part I. Proceedings, 2001. XXVIII, 1306 pages. 2001.
- Vol. 2074: V.N. Alexandrov, J.J. Dongarra, B.A. Julianio, R.S. Renner, C.J.K. Tan (Eds.), Computational Science – ICCS 2001. Part II. Proceedings, 2001. XXVIII, 1076 pages. 2001.
- Vol. 2081: K. Aardal, B. Gerards (Eds.), Integer Programming and Combinatorial Optimization. Proceedings, 2001. XI, 423 pages. 2001.
- Vol. 2082: M.F. Insana, R.M. Leahy (Eds.), Information Processing in Medical Imaging. Proceedings, 2001. XVI, 537 pages. 2001.
- Vol. 2091: J. Bigun, F. Smeraldi (Eds.), Audio- and Video-Based Biometric Person Authentication. Proceedings, 2001. XIII, 374 pages. 2001.
- Vol. 2092: L. Wolf, D. Hutchison, R. Steinmetz (Eds.), Quality of Service – IWQoS 2001. Proceedings, 2001. XII, 435 pages. 2001.

Table of Contents

Invited Paper

Deductions with Meaning	1
<i>Christof Monz, Maarten de Rijke (ILLC, Amsterdam)</i>	

Contributed Papers

Computational Solutions for Structural Constraints (Learning Structural Permissions in Categorical Grammar)	11
<i>Marcelo Finger (University of São Paulo)</i>	
Hypothetical Reasoning and Basic Non-constituent Coordination in Type-Logical Grammar	31
<i>Nissim Francez (Technion-IIT, Haifa)</i>	
Computational and Structural Aspects of Openly Specified Type Hierarchies	48
<i>Stephen J. Hegner (Umeå University)</i>	
Anaphora and Quantification in Categorical Grammar	70
<i>Gerhard Jäger (ZAS, Berlin)</i>	
An LTAG Perspective on Categorical Inference	90
<i>Aravind K. Joshi, Seth Kulick, Natasha Kurtonina (University of Pennsylvania)</i>	
Dominance Constraints: Algorithms and Complexity	106
<i>Alexander Koller, Joachim Niehren (Universität des Saarlandes), Ralf Treinen (Université Paris-Sud)</i>	
Strict Compositionality and Literal Movement Grammars	126
<i>Marcus Kracht (Freie Universität Berlin)</i>	
Categorical Minimalism	143
<i>Alain Lecomte (Université Pierre Mendès France, Grenoble 2 & LORIA, Nancy)</i>	
Sequential Construction of Logical Forms	159
<i>Wilfried Meyer-Viol (King's College, London)</i>	
Derivational Minimalism Is Mildly Context-Sensitive	179
<i>Jens Michaelis (Universität Potsdam)</i>	

Dominance Constraints in Context Unification	199
<i>Joachim Niehren, Alexander Koller (Universität des Saarlandes)</i>	
A Revision System of Circular Objects and Its Applications to Dynamic Semantics of Dialogues	219
<i>Norihiro Ogata (Osaka University)</i>	
Lexicalized Proof-Nets and TAGs	230
<i>Sylvain Pogodalla (XRCE, Meylan)</i>	
Lambek Calculus Proofs and Tree Automata	251
<i>Hans-Joerg Tiede (Illinois Wesleyan University)</i>	
Grammars with Composite Storages	266
<i>Christian Wartena (Universität Potsdam)</i>	
Author Index	287

Deductions with Meaning

Christof Monz and Maarten de Rijke

Institute for Logic, Language and Computation (ILLC)
University of Amsterdam, Plantage Muidergracht 24,
1018 TV Amsterdam, the Netherlands.
{christof, mdr}@wins.uva.nl

Abstract. In this paper, we consider some of the problems that arise if automated reasoning methods are applied to natural language semantics. It turns that out that the problem of ambiguity has a strong impact on the feasibility of any theorem prover for computational semantics. We briefly investigate the different aspects of ambiguity and review some of the solutions that have been proposed to tackle this problem.

1 Introduction

One of the concluding slogans of the FraCaS project on Frameworks for Computational Semantics is that ‘[t]here can be no semantics without logic’ [11]. We take this to mean that formalisms for semantic representation should be developed hand-in-hand with inference methods for performing reasoning tasks with representations and algorithms for representation construction.

Clearly, to be usable in the first place, representation formalisms need to come equipped with construction methods, and this explains the need for algorithmic tools. But what about the need for inference methods? At least three types of reasons can be identified. For *cognitive* purposes one may want to test the truth conditions of a representation against (a model of) speakers’ intuitions—this amounts to a model checking or theorem proving task. Also, the whole issue of what it is to understand a discourse may be phrased as a model generation task. *Computationally*, we need various reasoning tasks and AI-heuristics to help resolve quantifier scope ambiguity, or to resolve anaphoric relations in information extraction and natural language queries. And last, but not least, the very *construction* of semantic representations may require inference tools to be used in checking for consistency and informativity. At the end of the day, the main purpose of a semantic representation is that we can *do* something with it, both algorithmically and in terms of inference tasks.

Now, the present times are exciting ones for anyone with an interest in inference for natural language semantics. On the one hand, there is work in semantics that has little or no attention for inferential aspects. This is certainly the case for a lot of work in dynamic semantics and underspecified representation, and in the recent *Handbook of Logic and Language* [6] inferential methods for semantic representations are largely absent, despite the fact that a substantial part of the book is devoted to representational matters.

At the same time, there is a growing body of work aimed at developing inference methods and tools for natural language semantics, fed by a growing realization that these are ‘the heart of the enterprise’ [7, page viii]. This is manifested not only by various research initiatives (see below), but also by the fact that a number of textbooks and monographs on natural language semantics and its inferential and algorithmic aspects are in preparation [7,11], and by a recent initiative to set up a special interest group on Computational Semantics (see <http://www.coli.uni-sb.de/~patrick/SIGICS.html> for details).

In this note we survey some of the ongoing work on inference and natural language semantics; we identify commonalities, as well as possibilities and the main logical challenges we are confronted with in the field.

2 Putting Semantics to Work

2.1 Lines of Attack

It has often been claimed that classical reasoning based on first-order logic (FOL) is not appropriate as an inference method for natural language semantics. We are pragmatic in this matter: try to stick to existing formats and tools and see how far they get you, and only if they fail, one should develop novel formats and tools. Traditional inference tools (such as theorem provers and model builders) are reaching new levels of sophistication, and they are now widely and easily available. Blackburn and Bos [7] show that the ‘conservative’ strategy of using first-order tools can actually achieve a lot. In particular, they use first-order theorem proving techniques for implementing van der Sandt’s approach to presupposition. We refer the reader to the DORIS system, which is accessible on the internet at <http://www.coli.uni-sb.de/~bos/atp/doris.html>.

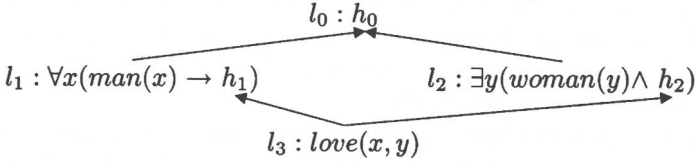
Although one may want to stick to first-order based tools as much as possible, for reasons of efficiency, or simply to get ‘natural representations’ it may pay to move away from the traditional first-order realm. Such a move may be particularly appropriate in two of the areas that currently pose the biggest challenges for computational semantics: *ambiguity* and *dynamics* [11, Chapter 8]. Let us consider some samples of deductive approaches in each of these two areas.

Reasoning with Quantifier Ambiguity. While the problem of ambiguity and underspecification has recently enjoyed a considerable increase in attention from computational linguists and computer scientists, the focus has mostly been on semantic aspects, and ‘reasoning with ambiguous sentences is still in its infancy’ [13]. Lexical ambiguities can be represented pretty straightforwardly by putting the different readings into a disjunction. It is also possible to express quantificational ambiguities by a disjunction, but quite often this involves much more structure than in the case of lexical ambiguities, because quantificational ambiguities are not tied to a particular atomic expression. For instance, the only way to represent the ambiguity of (1.a) in a disjunctive manner is (2).

- (1) Every man loves a woman.
 (2) $\forall x (man(x) \rightarrow \exists y (woman(y) \wedge love(x, y)))$
 $\vee \exists y (woman(y) \wedge \forall x (man(x) \rightarrow love(x, y)))$

Obviously, there seems to be some redundancy, because some subparts appear twice. Underspecified approaches such as the Core language Engine (CLE, [3]) or Underspecified Discourse Representation Theory (UDRT, [35]) allow us to represent quantifier ambiguities in a non-redundant way. The corresponding underspecified representation for (1) is given in (3).¹

(3)



This concise representation of the possible readings should allow us to avoid the state explosion problem. For representing the semantics of a natural language sentence this can be seen immediately, but to which extent theorem proving profits from underspecified representations is not easily determined. Up to now there is no proof theory which can directly work with underspecified representation. All of the approaches we are aware of rely, to some extent, on disambiguation. That is: first disambiguate an underspecified representation and then apply the rules of your proof theory. Once disambiguation has been carried out, this amounts, more or less, to classical proof theory, see, for instance, [17].

In [29] we have proposed a tableau calculus that interleaves disambiguation steps with deduction steps so that the advantages of an underspecified representation can, at least partially, be retained.

In addition, it is sometimes not necessary to compute all disambiguations, because there exists a strongest (or weakest) disambiguation. If there exists such a strongest (or weakest) disambiguation it suffices to verify (or falsify) this one, because it entails (or is entailed by) all other disambiguations. E.g., (4) has six readings which are listed in (5). For each reading we put the order of the quantifiers and negation sign as a shorthand in front of it.

(4) Every boy didn't see a movie

- (5) $(\forall\exists\neg) \forall x(boy(x) \rightarrow \exists y(movie(y) \wedge \neg see(x, y)))$
 $(\forall\neg\exists) \forall x(boy(x) \rightarrow \neg\exists y(movie(y) \wedge see(x, y)))$
 $(\neg\forall\exists) \neg\forall x(boy(x) \rightarrow \exists y(movie(y) \wedge see(x, y)))$
 $(\exists\forall\neg) \exists y(movie(y) \wedge \forall x(boy(x) \rightarrow \neg see(x, y)))$
 $(\exists\neg\forall) \exists y(movie(y) \wedge \neg\forall x(boy(x) \rightarrow see(x, y)))$
 $(\neg\exists\forall) \neg\exists y(movie(y) \wedge \forall x(boy(x) \rightarrow see(x, y)))$

¹ Actually, the underspecified representation in (3) differs slightly from the way underspecified representations are defined in [35], where the holes are not explicitly mentioned. Our representation is a bit closer to [9], but the differences between the frameworks are mainly notational.

In (6), we give the corresponding entailment graph which has as its elements the readings in (5). Two readings φ and ψ are connected by \Rightarrow if $\varphi \models \psi$.

$$(6) \quad \begin{array}{ccc} (\exists\forall\neg) & (\forall\neg\exists) & (\neg\forall\exists) (\exists\neg\forall) \\ \downarrow & & \downarrow \\ (\forall\exists\neg) & & (\neg\exists\forall) \end{array}$$

Unfortunately, this graph is not very dense. There are only two pairs of readings that stand in the entailment relation. Nevertheless, it allows for some improvement of the calculus, as it allows us to filter out some readings. $(\exists\forall\neg)$ and $(\forall\exists\neg)$ are two of the readings of (4), where $(\exists\forall\neg)$ entails $(\forall\exists\neg)$. On the other hand, if we are able to derive a contradiction for $(\forall\exists\neg)$, then we know that $(\exists\forall\neg)$ is contradictory, too. In [29] we have shown how the subset of readings which is sufficient can be identified.

But there is more to reasoning with quantificational ambiguity than just developing a calculus for it. In the presence of multiple readings of premises and conclusions, fundamental logical notions such as *entailment* receive new dimensions. Should *all* possible readings of the conclusion follow (in the traditional sense) from *all* possible readings of the premises for the ambiguous conclusion to qualify as a consequence of an ambiguous premise? Basic research in this direction has been carried out by a number of people [12,36,17,20]. Ultimately, the aim here is to obtain insights into the development and implementation of theorem provers for underspecified representations.

Reasoning with Pronoun Ambiguity. A number of calculi have been proposed for reasoning with dynamic semantics. [21,37,40] present natural deduction style calculi for Discourse Representation Theory, and [42] presents a tableau calculus. In the area of Dynamic Predicate Logic (DPL, [19]) and its many variations, [43] presents some ground-tableau calculi and [15] a sequent calculus. All of these approaches presuppose that pronouns are already resolved to some antecedent. Therefore, the problem of pronoun ambiguity does not arise within the calculus but the construction algorithm of the semantic representations. In order to employ the aforementioned calculi it is necessary that the semantic representation is disambiguated, but this might result in a huge number of readings, where the advantage of underspecified representation is lost. Again, it seems reasonable to interleave disambiguation and deduction steps, where disambiguation is only carried out if this is demanded by the deduction method.

The *resolution method* [39] has become quite popular in automated theorem proving, because it is very efficient and it is easily augmentable by lots of strategies which restrict the search space, see e.g., [23]. On the other hand, the resolution method has the disadvantage of presupposing that its input has to be in *clause form*, where clause form is the same as CNF but a disjunction is displayed as a set of literals (the *clause*) and the conjunction of disjunctions is a set of clauses. Probably the most attractive feature of resolution is that it has only one single inference rule, the resolution rule.

Applying the classical resolution method to a dynamic semantics introduces a problem: transforming formulas to clause form causes a loss of structural information. Therefore, it is sometimes impossible to distinguish between variables that can serve as antecedents for a pronoun and variables that cannot. [27,28] provide a resolution calculus that uses *labels* to encode the information about accessible variables. Each pronoun is annotated with a label that indicates the set of accessible antecedents.

There is a further problem with resolution calculus as it was presented in [27,28] is that it requires backtracking in order to be complete. Unfortunately, backtracking is hard to implement efficiently and it spoils some of the appeal of preferring resolution over tableau methods.

A tableau calculus for pronoun ambiguity has been introduced in [30]. This tableau calculus has a number of advantages over a resolution-based approach to pronoun resolution, as mentioned above. First of all, it is possible to interleave the computation of accessible variables with deduction, since preservation of structure is guaranteed in our signed tableau method. This is not possible in resolution, because it is assumed that the input is in conjunctive normal form, which destroys all structural information needed for pronoun binding. There, accessible antecedents can only be computed by a preprocessing step, cf. [27,28].

But the major advantage is that no backtracking is needed if the choice of an antecedent for a pronoun does not allow us to close all open branches; we simply apply pronoun resolution again, choosing a different antecedent.

2.2 Lessons Learned

The brief sketches of recent work on inference and natural language semantics given above show a number of things. First, all traditional computational reasoning tasks (theorem proving, model checking, model generation) are needed, but often in novel settings that work on more complex data structures. Dealing with ambiguity is one of the most difficult tasks for theorem provers, and we have seen in the previous section how we can tackle this problem. On the other hand, so far we have only looked at theorem proving for quantifier and pronoun ambiguity, separately; but what kind of problems arise if one tries to devise a theorem prover for a language containing both kinds of ambiguity? We will have a closer look at this later on.

Second, there are novel logical concerns both at a fundamental and at an architectural level. The former is illustrated by the proliferation of notions of entailment and by the need for incremental, structure preserving proof procedures. As to the latter, to move forward we need to develop methods for integrating specialized inference engines, possibly operating on different kinds of information, with other computational tools such as statistical packages, parsers, and various interfaces. We propose to use combinations of small specialized modules rather than large baroque systems. Of course, similar strategies in design and architecture have gained considerable attention in both computer science [8], and in other areas of applied logic and automated reasoning [10].

Combining Ambiguities. What happens if we combine both kinds of ambiguity and try to reason efficiently with formulas that contain ambiguous quantifier scopings *and* unresolved pronouns? Especially, which of the proof strategies that we used for dealing with the respective ambiguities can be adopted, and which ones raise problems?

When combining different kinds of ambiguity, ambiguities do not simply multiply, but they also interfere. Below, a short example is given, where (7) and its two readings in (8), an instance of quantifier ambiguity, is followed by (9), which contains an unresolved pronoun.

(7) Every man loves a woman.

(8) a. $\forall x (man(x) \rightarrow \exists y (woman(y) \wedge love(x, y)))$

b. $\exists y (woman(y) \wedge \forall x (man(x) \rightarrow love(x, y)))$

(9) But she is already married.

Here, (9) allows us to resolve the quantifier ambiguity of (7). Therefore, an appropriate calculus has to account for this. (9) filters out (8.a), because it does not provide an antecedent for the pronoun *she* in (9). This is easily seen, as (7) was uttered in the empty context, and (8.a) does not provide any antecedents. This implies that (8.a) cannot be a possible reading.

The preceding discussion so far hints at another problem that occurs if we try to reason in a combined framework. Considering only quantifier ambiguity, it was possible to neglect a reading φ if it entailed another reading ψ . Is this still possible if there are pronouns occurring in the proof which remain to be resolved? Reconsidering (7), the reading (8.b) entails (8.a), and it is sufficient to use only (8.a) in the proof. But if (7) is followed by (9), then (8.a) does not provide any antecedents for the pronoun in (9) and the pronoun remains unresolved. In fact, according to the discussion above, (8.a) would be filtered out, just because it cannot provide any antecedent; but then, no reading is left. (8.b) is filtered out, because it is stronger than (8.a), and (8.a) is filtered out for the reasons just given.

An obvious way out is to prefer weaker readings over stronger ones without throwing the stronger reading away. Only if the weaker reading does not cause any unresolvedness of pronouns, one can fully dispense with the stronger reading. For a longer discussion of this problem and some ways to solve this, the reader is referred to [33].

Incrementality. Implementations in computational semantics that employ the *theorem provers* normally state the inference tasks in a non-incremental way. For instance, DORIS filters out those readings of a natural language discourse that do not obey local informativity or local consistency constraints. In this process of filtering out readings, the system is often faced with very similar reasoning tasks involving very similar sets of premises and conclusions. In DORIS, these tasks are treated independently of each other, and every inference task is started from scratch. The set of formulas which are treated multiple times grows with the length of the discourse. Of course, this redundancy significantly decreases