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.
E-mail: {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' [CFvGG98]. 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* [vBtM97] 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' [BB98, 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 [BB98,CFvGG98], 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 [BB98] 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* [CFvGG98, 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' [vDP96]. 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) $\forall x\,(man(x) \to \exists y\,(woman(y) \wedge love(x,y)))$
$\vee\ \exists y\,(woman(y) \wedge \forall x\,(man(x) \to love(x,y)))$

Obviously, there seems to be some redundancy, because some subparts appear twice. Underspecified approaches such as the Core language Engine (CLE, [Als92]) or Underspecified Discourse Representation Theory (UDRT, [Rey93]) allow us to represent quantifier ambiguities in a non-redundant way. The corresponding underspecified representation for (1) is given in (3).[1]

(3)
$$\begin{tikzcd}
 & l_0 : h_0 & \\
l_1 : \forall x(man(x) \to h_1) \arrow[ur] & & l_2 : \exists y(woman(y) \wedge\ h_2) \arrow[ul] \\
 & l_3 : love(x,y) \arrow[ul] \arrow[ur] &
\end{tikzcd}$$

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, [vEJ96].

In [MdR98c] 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 reading 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)
$(\forall\exists\neg)$ $\forall x(boy(x) \to \exists y(movie(y) \wedge \neg see(x,y)))$
$(\forall\neg\exists)$ $\forall x(boy(x) \to \neg\exists y(movie(y) \wedge see(x,y)))$
$(\neg\forall\exists)$ $\neg\forall x(boy(x) \to \exists y(movie(y) \wedge see(x,y)))$
$(\exists\forall\neg)$ $\exists y(movie(y) \wedge \forall x(boy(x) \to \neg see(x,y)))$
$(\exists\neg\forall)$ $\exists y(movie(y) \wedge \neg\forall x(boy(x) \to see(x,y)))$
$(\neg\exists\forall)$ $\neg\exists y(movie(y) \wedge \forall x(boy(x) \to see(x,y)))$

[1] Actually, the underspecified representation in (3) differs slightly from the way underspecified representations are defined in [Rey93], where the holes are not explicitly mentioned. Our representation is a bit closer to [Bos96], 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 $\varphi \models \psi$.

$$
\text{(6)} \qquad
\begin{array}{cccc}
(\exists\forall\neg) & (\forall\neg\exists) & (\neg\forall\exists) & (\exists\neg\forall) \\
\Downarrow & & \Downarrow & \\
(\forall\exists\neg) & & (\neg\exists\forall) &
\end{array}
$$

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 [MdR98c] 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 [vD96,Rey95,vEJ96,Jas97]. 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. [KR96,RG94,Sau90] present natural deduction style calculi for Discourse Representation Theory, and [SE86] presents a tableau calculus. In the area of Dynamic Predicate Logic (DPL, [GS91]) and its many variations, [Vel97] presents some ground-tableau calculi and [vE98a] 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* [Rob65] 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., [Lov78]. 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 than can not. [MdR98a,MdR98b] 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 [MdR98a,MdR98b] 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 [MdR99]. 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. [MdR98a,MdR98b].

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 [Boo91], and in other areas of applied logic and automated reasoning [BS96].

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 [Mon99c].

Incrementality. Implementations in computational semantics that employ 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 the efficiency of the implementation, and it will prevent the system from scaling up.

[Mon99a,Mon99b] introduce a way of stating these inference tasks such that redundant applications of inference rules can be avoided. This is accomplished by taking context and the way contextual information is threaded through a discourse explicitly into account. The approach in [Mon99a,Mon99b] is based on formal theories of context, see, e.g., [AS94a,AS94b,MB97].

3 Further Directions and Challenges

The findings of the previous sections are supported by a number of further and novel developments in more applied areas adjacent to natural language semantics. We will restrict ourselves to three examples.

First, in syntactic analysis, partial or underspecified approaches to parsing are becoming increasingly popular [Abn96]. Just like underspecified representations in semantics, a partial parse fully processes certain phrases, but leaves some ambiguities such as modifier attachment underspecified. Given this similarity, it is natural to ask whether underspecified semantics can somehow be combined with partial parsing. An ongoing project at ILLC studies to which extent one can, for instance, use semantic information into account to resolve syntactic ambiguities; see `http://www.illc.uva.nl/~mdr/Projects/Derive/` for details. Note that combinations of underspecified representation and packed syntactic trees (parse forests) have been considered before [Sch96,Dör97], but no methods for using semantic information to resolve syntactic ambiguities are reported there.

Second, assuming that underspecified representations can usefully be combined with partial parsing, we may be able to improve methods in Information Extraction (IE). Common approaches to IE suffer from the fact that they either give only a very shallow analysis of text documents, as in approaches using word vectors, or that they are domain dependent, as in the case of template filling. More general techniques using some kind of logical representation could circumvent these disadvantages. Now, IE techniques provide the right data structures, but to access the information one needs the right retrieval algorithms. Logic-based Information Retrieval (IR) has been around, at least theoretically, since the mid 1980's [vR86]. An ongoing project at ILLC investigates to which extent underspecified reasoning and representation can be used for IR; again, see `http://www.illc.uva.nl/~mdr/Projects/Derive/` for details. We do not believe that these techniques can compete with IR methods for very large data collections, where logic-based techniques seem to be intractable, but we are confident about substantial quality improvements for smaller domains.

In this context, it seems interesting to investigate to which extent Description Logics can be employed to represent the content of a document. [MG92] consider a fragment of Montague Semantics ([Mon73]) that can be expressed in Description Logics. Formulas belonging to this fragment have to be quantifier-free, meaning that they do not contain any lambda abstractions. For instance, (10.b), which is the semantic representation of (10.a) belongs to the fragment, but (11.b), representing (11.a), does not.

(10)a. Mary read a book.
   b. (`Mary read` (`some book`))
(11)a. Mary read a book that John bought.
   b. (`Mary read` (`some` (λx (x `book` $\wedge$ (`John` (`bought` x))))))

For this quantifier-free fragment, [MG92] provides an inference procedure which decides satisfiability in polynomial time. More generally, Description Logics are concerned representations and inference algorithms for fragments of first- and higher-order logics in which quantification is of a restricted, or *guarded* nature; see, for instance, [Fra93,Küs98] for further uses of Description Logics in computational semantics. One of the important advantages of using Description Logics is that very efficient inference tools are available, such as DLP [PS98].

Finally, and coming from a completely different direction, there is work on the use of dynamic semantics to explain the meaning of programs in hybrid programming languages such as Alma-0 [ABPS] that combine the imperative and declarative programming paradigms. [vE98b] shows how dynamic predicate logic provides an adequate semantics for a non-trivial fragment of Alma-0, and how inference tools for dynamic predicate logic become verification tools for the hybrid programming language.

4 Conclusions

In this note we have identified some of the main concerns of doing inference for natural language semantics. One of the most difficult tasks in this context is the problem of reasoning with ambiguity. We have seen that it is possible to devise calculi which can deal with a particular kind of ambiguity, but that things get much more complicated if one tries to devise a calculus which can deal with different kinds of ambiguity. We have illustrated these concerns by means of samples from ongoing research initiatives, and, in addition, we have listed what we take to be some of the main challenges and most promising research directions in the area.

Acknowledgments

Christof Monz was supported by the Physical Sciences Council with financial support from the Netherlands Organization for Scientific Research (NWO), project 612-13-001. Maarten de Rijke was supported by the Spinoza Project 'Logic in Action' at ILLC, the University of Amsterdam.

References

[Abn96] S. Abney. Tagging and partial parsing. In *Corpus-Based Methods in Language and Speech*. Kluwer Academic Publishers, 1996.

[ABPS] K.R Apt, J. Brunekreef, V. Partington, and A. Schaerf. Alma-0: An imperative language that supports declarative programming. *ACM Toplas*, 1998. In press.

[Als92] H. Alshawi, editor. *The Core Language Engine*. MIT Press, Cambridge, Massachusetts, 1992.

[AS94a] G. Attardi and M. Simi. Building proofs in context. In *Proc. of Meta '94*, LNCS 883, pages 410–424. Springer, 1994.

[AS94b] G. Attardi and M. Simi. Proofs in context. In J. Doyle and P. Torasso, editors, *Principles of Knowledge Representation and Reasoning: Proc. of the Fourth International Conference*. Morgan Kaufmann, 1994.

[BB98] P. Blackburn and J. Bos. *Representation and Inference for Natural Language*. Available from `http://www.coli.uni-sb.de/~bos/comsem/`, 1998.

[Boo91] G. Booch. *Object-Oriented Design*. Benjamin/Cummings, 1991.

[Bos96] J. Bos. Predicate logic unplugged. In P. Dekker and M. Stokhof, editors, *Proceedings of the 10th Amsterdam Colloquium*, pages 133–142. ILLC, University of Amsterdam, 1996.

[BS96] F. Baader and K. Schulz, editors. *Frontiers of Combining Systems*, Applied Logic Series. Kluwer Academic Publishers, 1996.

[CFvGG98] D. Crouch, C. Fox, J. van Genabith, and S. Glasbey, editors. *The FraCaS Book*. Unpublished manuscript, 1998.

[Dör97] J. Dörre. Efficient construction of underspecified semantics under massive ambiguity. In *Proc. of the 35th Annual Meeting of the ACL*, 1997.

[Fra93] E. Franconi. A treatment of plurals and plural quantifications based on a theory of collections. *Minds and Machines*, 3(4):453–474, 1993.

[GS91] J. Groenendijk and M. Stokhof. Dynamic Predicate Logic. *Linguistics and Philosophy*, 14:39–100, 1991.

[Jas97] J. Jaspars. Minimal logics for reasoning with ambiguous expressions. CLAUS-report 94, Universität des Saarlandes, 1997.

[KR96] H. Kamp and U. Reyle. A calculus for first order Discourse Representation Structures. *Journal of Logic, Language and Information*, 5:297–348, 1996.

[Küs98] U. Küssner. Description logic unplugged. In E. Franconi et al., editor, *Proceedings of the 1998 International Description Logics Workshop (DL'98)*, 1998.

[Lov78] D.W. Loveland. *Automated Theorem Proving: A Logical Bases*. North-Holland, Amsterdam, 1978.

[MB97] J. McCarthy and S. Buvač. Formalizing context (expanded notes). In A. Aliseda, R. van Glabbeek, and D. Westerståhl, editors, *Computing Natural Language*, pages 13–50. Stanford University, 1997.

[MdR98a] C. Monz and M. de Rijke. Labeled resolution for discourse semantics. In *Proc. LD'98*. Dept. of Computer Science, Univ. of Freiburg, 1998.

[MdR98b] C. Monz and M. de Rijke. A resolution calculus for dynamic semantics. In J. Dix et al, editor, *Proc. JELIA'98*, LNAI 1489, pages 184–198. Springer, 1998.

[MdR98c] C. Monz and M. de Rijke. A tableaux calculus for ambiguous quantification. In H. de Swart, editor, *Automated Reasoning with Analytic Tableaux and Related Methods,* TABLEAUX'98, LNAI 1397, pages 232–246. Springer, 1998.

[MdR99] C. Monz and M. de Rijke. A tableau calculus for pronoun resolution. In N.V. Murray, editor, *Automated Reasoning with Analytic Tableaux and Related Methods,* TABLEAUX'99, LNAI. Springer, 1999.

[MG92] D.A. McAllester and R. Givan. Natural language syntax and first-order inference. *Artificial Intelligence*, 56(1):1–20, 1992.

[Mon73] R. Montague. The proper treatment of quantification in ordinary english. In J. Hintikka, J. Moravcsik, and P. Suppes, editors, *Approaches to Natural Language. Proceedings of the 1970 Stanford Workshop on Grammar and Semantics*. Reidel, Dordrecht, 1973.

[Mon99a] C. Monz. Computing presuppositions by contextual reasoning. In *Proceedings of the AAAI-99 Workshop on Reasoning in Context for AI Applications*, 1999.

[Mon99b] C. Monz. Contextual inference in computational semantics. Submitted for publication, 1999.

[Mon99c] C. Monz. Underspecified theorem proving with different kinds of ambiguity. Institute for Computational Linguistics, University of Stuttgart, 1999.

[PS98] P. Patel-Schneider. DLP system description. In E. Franconi et al., editor, *Proceedings of the 1998 International Description Logics Workshop (DL'98)*, pages 87–89, 1998.

[Rey93] U. Reyle. Dealing with ambiguities by underspecification: Construction, representation, and deduction. *Journal of Semantics*, 10(2):123–179, 1993.

[Rey95] U. Reyle. On reasoning with ambiguities. In *Proceedings of the 6th Meeting of the EACL*, pages 1–8, Dublin, 1995.

[RG94] U. Reyle and D. Gabbay. Direct deductive computation on Discourse Representation Structures. *Linguistics and Philosophy*, 17(4):343–390, 1994.

[Rob65] J.A. Robinson. A machine oriented logic based on the resolution principle. *Journal of the ACM*, 12(1):23–41, 1965.

[Sau90] W. Saurer. A natural deduction system for Discourse Representation Theory. *Journal of Philosophical Logic*, 22:249–302, 1990.

[Sch96] M. Schiehlen. Semantic construction from parse forests. In *Proc. COLING'96*, 1996.

[SE86] C. Sedogbo and M. Eytan. A tableau calculus for DRT. *Logique et Analyse*, 31:379–402, 1986.

[vBtM97] J. van Benthem and A. ter Meulen, editors. *Handbook of Logic and Language*. North-Holland, 1997.

[vD96] K. van Deemter. Towards a logic of ambiguous expressions. In van Deemter and Peters [vDP96].

[vDP96] K. van Deemter and S. Peters, editors. *Semantic Ambiguity and Underspecification*. CSLI Publications, 1996.

[vE98a] J. van Eijck. Axiomatizing dynamic logics for anaphora, 1998. CWI, Amsterdam.

[vE98b] J. van Eijck. Programming with dynamic logic. Manuscript, CWI, Amsterdam, 1998.

[vEJ96] J. van Eijck and J. Jaspars. Ambiguity and reasoning. Technical Report CS-R9616, CWI, Amsterdam, 1996.

[Vel97] F. Veltman. Dynamic predicate logic, 1997. ILLC, Amsterdam.

[vR86] K. van Rijsbergen. A non-classical logic for information retrieval. *The Computer Journal*, 29:481–485, 1986.