# Chapter 16

# Dynamic Semantics and Ambiguity

CHRISTOF MONZ

ABSTRACT. Ambiguity is one of the most difficult problems in natural language processing. We present a dynamic semantics which gives a formal account of the behavior of ambiguous expressions occurring in a sequence of sentences. Ambiguous sentences are represented in an underspecified way and the update potential of underspecified representations is defined. Updating information states with underspecified semantic representations is interleaved with partial disambiguation, helping to avoid the problem of state explosion. In general, it is considered how ambiguity survives throughout a discourse and how contextual information can restrict ambiguity.

## 1 Introduction

Natural language expressions can be highly ambiguous, and this ambiguity may have various faces. Well-known phenomena include lexical and syntactic ambiguities. In this paper we focus on constructing a discourse semantic for quantificational ambiguities and ambiguities evolving in pronoun resolution, as exemplified in (16.1.a) and (16.1.b), respectively.

(16.1) (a) Every man loves a woman.
(b) A man saw a boy in the park. He whistled.

The different readings of (16.1.a) correspond to the two logical representations in (16.2) and the pronoun in (16.1.b) can be resolved in two ways as illustrated in (16.3)

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

(16.3) (a) A man$_i$ saw a boy in the park. He$_i$ whistled.



*Proceedings of the Third ESSLLI Student Session*
Ivana Kruijff-Korbayová (editor).

(b) A man saw a boy$_i$ in the park. He$_i$ whistled.

We refer the reader to [vDP96] for extensive discussions of these and other examples of quantificational ambiguity. Often ambiguous sentences have a preferred reading, cf. [KM93, JK96] for more details, but adding preference selection would result in a nonmonotonic framework. For instance, assume that a wide scope reading for the universal quantifier is preferred in (16.1), then (16.2.a) would be an appropriate semantic representation, but if (16.1) is followed by

(16.4) But she is already married.

(16.2.b) would be more appropriate. If we want to process a discourse in a monotone fashion, we must initially allow for both possibilities.

The problem of ambiguity and underspecification has recently enjoyed a considerable increase in attention from computational linguists, computer scientists and logicians (see, for instance, [vDP96]). However, the focus has mostly been on giving a semantic representation for single sentences. Although some of the theories are couched in a dynamic setting (cf. for example [Bos95, Rey93]), most of them do not give a formal account of the behavior of ambiguous expressions in a discourse, but see [Fer97, Mon98]. Roughly, two things are lacking: First, there is no detailed account of pronoun resolution. In [Bos95] coindexation is assumed and in UDRT ([Rey93, Rey95]) only one way of binding is considered, but not different ways of binding in parallel. Second, there is no notion of an ambiguous context and ambiguous updating. In other words, the update potential of ambiguous sentences is not given.

The aim of this paper is to give a dynamic semantics for a language that includes ambiguous expressions, where ambiguity is represented in a non-redundant, underspecified way, as it is done in [Rey93, Bos95].

The rest of this paper is structured as follows. Section 2 discusses representational issues of ambiguity and the notion of underspecification is introduced. In Section 3 a dynamic semantics for a language including underspecified representations is given and some of its properties are discussed. Finally, conclusions and prospects for further work are provided in Section 4.

## 2 Representing Ambiguity

It is possible to represent quantificationally ambiguous formulas in a compact way as upper semi-lattices, cf. [Rey93]:

(16.5)
$$\begin{tikzcd} & l_0 : h_0 & \\ l_1 : \forall x(man(x) \rightarrow h_1) \arrow[ur] & & l_2 : \exists y(woman(y) \wedge h_2) \arrow[ul] \\ & l_3 : love(x, y) \arrow[ul] \arrow[ur] & \end{tikzcd}$$

**1 Definition (Underspecified Representation)** An underspecified representation (*UR*) is a quadruple $\langle LHF, L, H, C \rangle$, where *LHF* is the set of labeled h-formulas, and $C$ is a partial order on the labels ($L$) and holes ($H$) occurring in *LHF*. A *hole* indicates that the concrete form of this subformula is not yet known. For instance, we do not know what the scope of the universal quantifier is, so the succedent of the implication is a hole. *h-formulas* are the formulas possibly containing holes. In (16.5) $LHF = \{l_0 : h_0, l_1 : \forall x(man(x) \rightarrow h_1), l_2 : \exists y(woman(y) \wedge h_2), l_3 : love(x, y)\}$ and $C = \{l_1 \leq h_0, l_2 \leq h_0, l_3 \leq h_1, l_3 \leq h_2\}$.

There are two possible sets of instantiations, $\iota_1$ and $\iota_2$, of the holes $h_0$, $h_1$, $h_2$, $h_3$ in (2) which obey the constraints:

$$\iota_1 = \{h_0 := l_1, h_1 := l_2, h_2 := l_3\}$$
$$\iota_2 = \{h_0 := l_2, h_2 := l_1, h_1 := l_3\}$$

For each instantiation of the holes there is a corresponding substitution $\sigma(\iota)$ which is like $\iota$ but $h := \varphi \in \sigma(\iota)$ if there is a $l$, such that $l : \varphi \in LHF$ and $h := l \in \iota$.

An obvious question at this point is, how does one associate a *UR* with a given natural language expression? We will not address this issue here, but we will assume that there exists some mechanism for arriving at *UR*'s, see for example [Kön94].

The next step is to define an extension of the language of first-order logic, $\mathcal{L}$, in which both standard (unambiguous) expressions occur side by side with the above underspecified representations. The resulting language of the language of underspecified logic, or $\mathcal{L}^u$ for short, is the language in which we will use for constructing semantic representations.

**2 Definition (Underspecified Logic)** A formula $\varphi$ is a formula of our *underspecified logic* $\mathcal{L}^u$, or a *u-formula*, that is, a formula possibly containing underspecified representations, if it is built up from underspecified representations and the usual atomic formulas from standard first-order logic using the familiar boolean connectives and quantifiers.

**3 Example** As an example of a more complex u-formula consider the semantic representation of *if every boy didn't sleep and John is a boy, then John didn't sleep.*

$$\left( \left( \begin{tikzcd} & l_0 : h_0 & \\ l_1 : \neg\, h_1 \arrow[ur] & & l_2 : \forall x\,(boy(x) \rightarrow h_2) \arrow[ul] \\ & l_3 : sleep(x) \arrow[ul] \arrow[ur] & \end{tikzcd} \right) \wedge boy(j) \right) \rightarrow \neg sleep(j)$$

**4 Definition (Total Disambiguations)** Then, by $\delta(\varphi)$ we denote the set of total disambiguations of the u-formula $\varphi$, where for all $d \in \delta(\varphi)$, $d \in \mathcal{L}$. For complex u-formulas $\delta$ is defined recursively:

1. $\delta(\langle LHF, L, H, C\rangle)$ = the set of $LHF\sigma(\iota)$ such that
   (i) $\iota$ is an instantiation and $\sigma(\iota)$ is the corresponding substitution
   (ii) $H\iota = L$
   (iii) for all $l, l' \in L$, $l \leq l' \in closure(C\iota)$ or $l' \leq l \in closure(C\iota)$
2. $\delta(\neg\varphi) = \{\ \neg d \mid d \in \delta(\varphi)\ \}$
3. $\delta(\varphi \circ \psi) = \{\ d \circ d' \mid d \in \delta(\varphi),\ d' \in \delta(\psi)\ \}$, where $\circ \in \{\wedge, \vee, \rightarrow\}$
4. $\delta(\mathcal{Q}x\varphi) = \{\ \mathcal{Q}xd \mid d \in \delta(\varphi)\ \}$, where and $\mathcal{Q} \in \{\forall, \exists\}$.

To see whether an underspecified representation holds in a model or not we check whether this holds for all its total disambiguations.

**5 Definition (Validity and Falsity)** Given an arbitrary model $M$ and a u-formula $\varphi$

$M \models \varphi$ iff $M \models d$ , for all $d \in \delta(\varphi)$

$M \models \neg\varphi$ iff $M \not\models d$ , for all $d \in \delta(\varphi)$

## 3 A Dynamic Semantics for Ambiguity

To combine underspecified representation and update semantics we take Dekker's *EDPL* [Dek93] as a basis and adapt it in a way such that it fits our purposes. We chose *EDPL* because it is both, eliminative *and* distributive and the notion of an information state is explicit.

The adaption proceeds mainly along two lines. First, contrary to approaches in the tradition of dynamic predicate logic (DPL), see [GS91], we do not take resolution of pronouns as granted; i.e. coindexation of pronouns and their antecedents is not pressuposed. We dispense with this — admittedly quite handy — restriction on our input, because it is unrealistic and anaphora resolution is an important source of ambiguity in natural language texts, which we would like to account for. If we do not assume coindexation, it is necessary to deal with pronouns in a different way. We do so by binding pronominal arguments by three new operators $\mathsf{he}$, $\mathsf{she}$ and $\mathsf{it}$, similar to [Bea93], that roughly say that its arguments have to be identified with an accessible antecedent in a given context. Second, whereas in *EDPL* contexts are identified with states, we have to think about the notion of an ambiguous context. In this paper we define an *ambiguous context*, $\sigma$, simply by a set of states, where each state in a set of states represents an unambiguous reading of the context. Of course the notion of an update has to be redefined appropriately. A context is said to be *unambiguous* if it contains exactly one state.

## 3.1 EDPL

Before adapting *EDPL*, we briefly repeats its semantic definitions.

**6 Definition (Semantics of EDPL)**

$$
\begin{array}{lll}
s[\![R(x_1..x_n)]\!] & = \{i \in s \mid \langle i(x_1)..i(x_n)\rangle \in F(R)\} & \text{if } x_1..x_n \in D(s) \\
s[\![x = y]\!] & = \{i \in s \mid i(x) = i(y)\} & \text{if } x, y \in D(s) \\
s[\![\neg\varphi]\!] & = s - s[\![\varphi]\!] & \\
s[\![\exists x\varphi]\!] & = s[x]\,[\![\varphi]\!] & \text{if } x \notin D(s) \\
s[\![\varphi \wedge \psi]\!] & = s[\![\varphi]\!][\![\psi]\!] &
\end{array}
$$

where $s - s' = s \backslash \{i \restriction_{D(i)} \mid i \in s'\}$ and
$s[x] = \{j \mid \exists i \in s : i \leq_{\{x\}} j\}$ , such that $i \leq_X j$ iff
$D(j) = D(i) \cup X$ and $i = j \restriction_{D(i)}$

The other connectives, $\forall$, $\vee$, $\rightarrow$, can be defined in terms of the definitions given above.

## 3.2 Pronoun Resolution

The first difference between *EDPL* and our dynamic predicate logic with ambiguity (*DPLA*), concerns the treatment of pronouns. In *EDPL* it is assumed that pronouns are resolved and therefore pronouns and there antecedents use the same variable. In our approach pronouns are bound by a pronominal operator $\mathsf{P}$, $\mathsf{P} \in \{\mathsf{he}, \mathsf{she}, \mathsf{it}\}$, so $\mathsf{she}\, x\, \varphi$ means that the argument $x$ in $\varphi$ is pronominal and its antecedent must be female.[1] Because pronom-

[1] This is of course a simplified picture of the whole issue of grammatical and natural gender and the way it restricts pronoun resolution.

inal resolution might be ambiguous, updating a context $\sigma$ with $[\![\mathsf{P}x\varphi]\!]$ does not have to return a single update state for each $s \in \sigma$, but a set of them.

**7 Definition**

$$s[\![\mathsf{P}x\varphi]\!] = \{s[\![\varphi[x/y]]\!] \mid y \in D(s)\ ,\ s[\![\varphi[x/y]]\!] \neq \emptyset\}, \quad \text{where } \mathsf{P} \in \{\mathsf{he}, \mathsf{she}, \mathsf{it}\}$$

The restriction $y \in D(s)$ ensures that only accessible antecedents are considered.

To make the return value of an update to be consumable by the next update function we have to generalize our notion of an update: an update function is a partial function from sets of states to sets of states. We will call sets of states also *ambiguous contexts*.

**8 Definition (Pronoun Resolution)** Let $\sigma$ be an ambiguous state.

$$\sigma[\![\mathsf{P}x\varphi]\!] = \bigcup_{s\in\sigma} s[\![\mathsf{P}x\varphi]\!], \qquad \text{where } \mathsf{P} \in \{\mathsf{he}, \mathsf{she}, \mathsf{it}\}$$

It is obvious that this way of resolving pronouns is very inefficient, but we want to give a monotone semantics for ambiguity, and of course it is possible to augment our theory with some heuristics that allow to restrict the set of readings or ways of resolving pronouns, see for instance [JK96].

## 3.3 Quantificational Ambiguities

Next we consider quantificational ambiguities, for which a compact representation was given earlier. The semantics of underspecified representations is defined in terms of total disambiguations see [Rey95], so avoiding redundancy on the the syntactic side is possible, but it returns on the semantic side. Often this is not a problem, because one can consider the syntactic side, only. For instance given a complete and sound calculus, reasoning in this system amounts to syntactic manipulation. But in an ambiguous setting things are a bit different: underspecified representations are compact representations of several formulas, i.e. they are syntactically ambiguous, and at the moment it is not obvious how to reason with ambiguous formulas without disambiguating them at least partially, cf. [Jas97, MdR98].

The question is how can we restrict the massive branching of contexts when updating with ambiguous information. If we do not update with total disambiguations but instead interleave updating and disambiguation we might gain a decrease in complexity because further disambiguation steps depend on earlier successful updates. Updates of complex formulas are computed in a top-down manner. The first step is to identify the possible outermost operators in a quantificationally ambiguous expression and start computing the updates until we face a *hole*. Then we make a choice again and check what are the operators which may have the next wider scope respectively, continue updating, and so on. Possibly some of these first steps exclude some disambiguations because they are not compatible with the given input context.

Partially updating with quantificationally ambiguous material changes also the set of ordering constraints. This makes it necessary to make contexts more complex. We will say that an ambiguous context $\sigma$ is a set of triples of the form $\langle s, C, LHF \rangle$. Updating $\sigma$ proceeds in a top-down manner:

**9 Definition** Let $UR = \langle LHF, L, C \rangle$ be an underspecified representation. $\mathsf{top}(UR)$ returns the formula of the top node and $\mathsf{dt}(h)$ are the possible immediate subformulas of a hole $h$.

$$\begin{array}{lll} \sigma[\![UR]\!] & = & \{\langle s, C, LHF \rangle \mid \exists C', LHF' : \langle s, C', LHF' \rangle \in \sigma\}[\![\mathsf{top}(UR)]\!] \\ \sigma[\![h]\!] & = & \bigcup\limits_{l_j:\varphi_j \in \mathsf{dt}(h)} \{\langle s, C \cup \{l_k \leq h'\}, LHF \rangle \mid \langle s, C, LHF \rangle \in \sigma, l_k : \psi \in \mathsf{dt}(h)\}[\![\varphi_j[h']]\!] \\ \sigma[\![h]\!] & = & \sigma[\![\varphi_j]\!], \text{ if } \mathsf{dt}(h) = \{l_j : \varphi_j\} \text{ and } \varphi_j \text{ does not contain a hole} \end{array}$$

Having defined the update function of underspecified representations and expressions containing pronouns, we adapt the update conditions of the kind of expressions already considered in *EDPL*. First we have to adapt the update definition of unambiguous contexts from simple states to triples of the form $\langle s, C, LHF \rangle$.

**10 Definition** if $\varphi$ does not contain holes or underspecified representations:

$$\langle s, C, LHF \rangle[\![\varphi]\!] = \begin{cases} \langle s[\![\varphi]\!], C, LHF \rangle, & \text{if } s[\![\varphi]\!] \neq \emptyset \text{ and defined and } C \not\vdash \bot \\ \text{undefined}, & \text{otherwise} \end{cases}$$

**11 Definition (Semantics of DPLA)** The update potentials of the logical connectives are those given in Definition 3 plus the following ones, where $s^c$ is of the form $\langle s, C, LHF \rangle$.

$$\begin{array}{ll} \sigma[\![R(x_1 \ldots x_n)]\!] & = \{s^c[\![R(x_1 \ldots x_n)]\!] \mid s^c \in \sigma, s^c[\![R(x_1 \ldots x_n)]\!] \text{ is defined}\} \\ \sigma[\![x = y]\!] & = \{s^c[\![x = y]\!] \mid s^c \in \sigma, s^c[\![x = y]\!] \text{ is defined}\} \\ \sigma[\![\neg\varphi]\!] & = \{s^c - t^c | s^c \in \sigma, t^c \in \{s^c\}[\![\varphi]\!]\} \\ \sigma[\![\exists x\varphi]\!] & = \{s^c[\![\exists x\varphi]\!] \mid s^c \in \sigma, s^c[\![\exists x\varphi]\!] \text{ is defined}\} \\ \sigma[\![\varphi \wedge \psi]\!] & = \sigma[\![\varphi]\!][\![\psi]\!] \\ \sigma[\![\mathsf{P}x\varphi]\!] & = \bigcup\limits_{s^c \in \sigma} s^c[\![\mathsf{P}x\varphi]\!], \qquad \text{if defined} \\ \sigma[\![UR]\!] & = \{\langle s, C, LHF \rangle \mid \exists C', LHF' : \langle s, C', LHF' \rangle \in \sigma\}[\![\mathsf{top}(UR)]\!] \end{array}$$

$$\sigma[\![h]\!] = \bigcup_{l_j:\varphi_j \in \mathsf{dt}(h)} \{\langle s, C \cup \{l_k \leq h'\}, LHF\rangle \mid \langle s, C, LHF\rangle \in \sigma,$$
$$l_k : \psi \in \mathsf{dt}(h)\}[\![\varphi_j[h']\,]\!]$$
$$\sigma[\![h]\!] = \sigma[\![\varphi_j]\!], \text{ if } \mathsf{dt}(h) = \{l_j : \varphi_j\} \text{ and } \varphi_j \text{ does not contain a hole}$$

where $\langle s, C, LHF\rangle - \langle t, C, LHF\rangle = \langle s - t, C, LHF\rangle$

The next example illustrates the advantage of partial updating over regular updating with total disambiguations.

**12 Example** Consider a situation $\sigma$, where nothing about a male person was said and somebody utters *every boy didn't see a movie he knows*. First of all we have to give an underspecified representation *UR* of the sentence:

$l_0 : h_0$

$l_1 : \exists y(m(y) \wedge \mathsf{he}z(k(z,y)\wedge\ h_1\ )$ $\quad l_2 : \forall x(b(x) \to h_2\ )$ $\quad l_3 : \neg\ h_3$

$l_4 : see(x,y)$

Updating $\sigma$ with *UR* means updating $\sigma$ with the top node of *UR*:

1. $\sigma[\![\,UR]\!] = \sigma[\![h_0]\!]^C$

Then we have to consider the three operators which can have widest scope:

2. $\{\langle s, C \cup \{l_2, l_3 \leq h_1\}, LHF\rangle \mid \langle s, C, LHF\rangle \in \sigma\}[\![\exists y(m(y) \wedge \mathsf{he}z(k(z,y)) \wedge h_2)]\!]$
$\cup \{\langle s, C \cup \{l_1, l_3 \leq h_2\}, LHF\rangle \mid \langle s, C, LHF\rangle \in \sigma\}[\![\forall x(b(x) \to h_1]\!]$
$\cup \{\langle s, C \cup \{l_1, l_2 \leq h_3\}, LHF\rangle \mid \langle s, C, LHF\rangle \in \sigma\}[\![\neg h_3]\!]$

If we focus on updating the first one, we see that the pronoun cannot be resolved properly.

3. $\{\langle s[y], C \cup \{l_2, l_3 \leq h_1\}, LHF\rangle \mid \langle s, C, LHF\rangle \in \sigma\}[\![m(y) \wedge \mathsf{he}z(k(z,y)) \wedge h_2]\!]$
$= \{\langle \{j \mid \exists i \in s : i \leq_{\{y\}} j\}, C \cup \{l_2, l_3 \leq h_1\}, LHF\rangle \mid \langle s, C, LHF\rangle \in \sigma\} = \sigma_1$
$= \sigma_1[\![m(y) \wedge \mathsf{he}z(k(z,y)) \wedge h_2]\!] = \sigma_1[\![m(y)]\!][\![\mathsf{he}z(k(z,y)) \wedge h_2]\!]$
$= \{\langle \{j \mid \exists i \in s : i \leq_{\{y\}} j, j(y) \in F(m)\}, C_1, LHF\rangle \mid \langle s, C_1, LHF\rangle \in \sigma_1\} = \sigma_2$
$= \sigma_2[\![\mathsf{he}z(k(z,y)) \wedge h_2]\!]$

Because $F(m) \cap \mathsf{he} = \emptyset$ the next update results in

4. $= \{\langle \emptyset, C_1, LHF\rangle \mid \langle s, C_1, LHF\rangle \in \sigma_2\}$

Therefore the update of the whole formula reduces to

5. $\{\langle s, C \cup \{l_1, l_3 \leq h_2\}, LHF\rangle \mid \langle s, C, LHF\rangle \in \sigma\}[\![\forall x(b(x) \to h_1]\!]$
$\cup \{\langle s, C \cup \{l_1, l_2 \leq h_3\}, LHF\rangle \mid \langle s, C, LHF\rangle \in \sigma\}[\![\neg h_3]\!]$

Note, that we are able to rule out the wide scope reading of the existential quantifier by leaving the scoping relation of the universal quantifier and the negation still underspecified.

## 3.4 Some Properties of DPLA

*EDPL* is distributive and eliminative; but what about *DPLA*? It is easy to see that distributivity holds, because in none of our definitions the update depends on the whole context. Combining this with the fact that *EDPL* is distributive we get:

**13 Observation** $\sigma[\![\varphi]\!] = \bigcup_{s^c \in \sigma} \{s^c\}[\![\varphi]\!]$ , if defined

Although updates of ambiguous formulas possibly contain more readings than the original context, each of these reading eliminates some (possibly null) assignment functions.

**14 Observation** *DPLA* is eliminative.

For the fragment which does not involve ambiguities the properties of *EDPL* generally hold for *DPLA*, too.

**15 Definition (Entailment)** Following [Dek93], where $s \leq s'$ iff $\forall i \in s \exists j \in s' : i \subseteq j$ we say that $\sigma \leq \sigma'$ iff $\forall\langle s, C, LHF\rangle\forall\langle s', C', LHF'\rangle : s \leq s'$. The entailment relation can now be defined as:

$$\varphi_1 \ldots \varphi_n \models \psi \text{ iff } \sigma[\![\varphi_1]\!] \ldots [\![\varphi_n]\!] \leq \sigma[\![\psi]\!]$$

This is just one of many possible definitions of the entailment relation. Two things should be pointed out: first, the dynamic force of the existential quantifier does not exceed the entailment relation. So anaphora occurring in the conclusion cannot be bound by an existential quantifier in the premises, cf. [Ben96, Gro95] for a comparison of dynamic entailment relations. Second, our notion of entailment is very weak, because it presupposes that all readings in the premises entail all readings in the conclusion, but for dealing with natural language semantics it seems to be an appropriate definition, see [vD96, MdR98, Rey95] for further discussion and refinements.

# 4 Conclusion

Our formalism is monotone and it inherits the properties of eliminativity and distributivity from *EDPL*. It is more general than *DPL* based approaches, because it gives an account for pronoun resolution without relying on coindexation and it does so by considering different possibilities in parallel. Contrary to [Bos95, Rey93] the update potential of ambiguous sentences is defined and it is possible to compute the update potential of a sequence of ambiguous sentences.

*DPLA* allows us to give a general picture of ambiguous updating, which can be extended in several directions by adding some heuristics to yield a more efficient formalism. We would like to point out, that we did not try to model how human speakers deal with ambiguous information, but we did try to give an account that is general enough to be specified in a way to do so.

It might be interesting to see how our framework interacts with approaches on deduction in an ambiguous setting, cf. [Jas97, MdR98]. Jaspars noticed that the notion of relative ambiguity is necessary in order to get a well behaved calculus for ambiguity. This notion is part of our framework and of dynamic semantics in general, because updates depend on the context to which they are applied, and so do ambiguous updates. The next step will be to see how a calculus for reasoning in ambiguous dynamic semantics can be developed.

## Acknowledgments

The research in this paper was supported by the Spinoza project 'Logic in Action' at the University of Amsterdam.

## References

[Bea93] D. Beaver. Kinematic Montague Grammar. In H. Kamp, editor, *Presupposition*, DYANA-2 Report R.2.2a. University of Amsterdam, 1993.

[Ben96] J. van Benthem. *Exploring Logical Dynamics.* Studies in Logic, Language and Information, CSLI Publications, 1996.

[Bos95] J. Bos. Predicate logic unplugged. In P. Dekker and M. Stokhof, editors, *Proc. 10th Amsterdam Colloquium.* ILLC, University of Amsterdam, 1995.

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

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

[Dek93] P. Dekker. *Transsentential Meditations.* PhD thesis, ILLC / Department of Philosophy, University of Amsterdam, Amsterdam, 1993.

[Fer97] T. Fernando. Ambiguity under changing contexts. *Linguistics and Philosophy*, 20:575–606, 1997.

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

[Gro95] W. Groeneveld. *Logical Investigations into Dynamic Semantics.* PhD Thesis, Institute for Logic, Language and Computation, University of Amsterdam, 1995.

[Jas97] J. Jaspars. Minimal logics for reasoning with ambiguous expressions. CLAUS-Report 94, University of Saarbrücken, 1997.

[JK96] J. Jaspars and M. Kameyama. Preferences in dynamic semantics. In M. Stokhof and P. Dekker, editors, *Proceedings of the 10th Amsterdam Colloquium*, pages 445–464. ILLC, Amsterdam, 1996.

[KM93] H. S. Kurtzman and M. C. MacDonald. Resolution of quantifier scope ambiguities. *Cognition*, 48:243–279, 1993.

[Kön94] E. König. A study in grammar design. Arbeitspapier des Sonderforschungsbereich 340 no. 54, Institut für Maschinelle Sprachverarbeitung, 1994.

[KR96] E. König and U. Reyle. A general reasoning scheme for underspecified representations. In H.-J. Ohlbach and U. Reyle, editors, *Logic and Its Applications. Festschrift for Dov Gabbay*. Kluwer Academic Publishers, 1996.

[MdR98] 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.

[Mon98] C. Monz. Dynamic semantics and underspecification. In H. Prade, editor, *Proceedings of the 13$^{th}$ European Conference on Artificial Intelligence (ECAI'98)*, pages 201–202. John Wiley & Sons, 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 Association for Computational Linguistics, European Chapter*, pages 1–8, Dublin, 1995.