# Dynamic Semantics and Underspecification

**Christof Monz**[1]

**Abstract.** Most work on ambiguity in natural language focuses on the semantic representation of single ambiguous sentences. In this paper we present a dynamic semantics which gives a formal account of the behavior of ambiguous expressions occurring in a sequence of sentences. It is considered how partial disambiguation and dynamic updating can be interleaved to restrict ambiguity in an efficient way.

## 1 Introduction

Natural Language Processing (NLP) has a long tradition in Artificial Intelligence, but it still remains one of the hardest problems in AI. One reason herefore is the ambiguity of natural language expressions, which makes semantic construction and automated deduction very inefficient. Although there are approaches which allow to represent ambiguous expressions efficiently, e.g. [7], they only consider single sentences. In this paper we focus on semantic construction of sequences of sentences involving quantificational ambiguities, as illustrated by the notorious example in (1)

(1) Every man loves a woman.

which has two readings:

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

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

Often ambiguous sentences have a preferred reading, cf. [5, 3] 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 (1), then (2.a) would be an appropriate semantic representation, but if (1) is followed by

(3) But she is already married.

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

Quantificationally ambiguous sentences as (1) can be represented in a compact way as upper semi-lattices, cf. [7]:

(4)
$$l_0 : h_0$$
$$l_1 : \forall x(man(x) \rightarrow h_1) \qquad l_2 : \exists y(woman(y) \wedge h_2)$$
$$l_3 : love(x,y)$$

An underspecified representation ($UR$) is a pair $\langle V, C\rangle$, where $V$ is the set of nodes, and $C$ is a partial order on the labels and holes occurring in $V$. A *hole* indicates that the concrete form of this subformula is not known yet. For instance, we do not know what the scope of the universal quantifier is, therefore the succedent of the implication is a hole. In (4) $V = \{l_0 : h_0, l_1 : \forall x(man(x) \rightarrow h_1), l_2 : \exists y(woman(y) \wedge h_2), l_3 : love(x,y)\}$ and $C = \{l_1 \leq h_0, l_2 \leq h_0, l_3 \leq h_1, l_3 \leq h_2\}$. We neglect how a mapping from natural language sentences to underspecified representations can be defined, but see [4] for a detailed account.

## 2 Underspecified Dynamic Semantics

To combine underspecified representations and dynamic semantics we take Dekker's *EDPL* [1] as a basis and adapt it in a way such that it fits our purposes. In *EDPL* states are sets of partial variable assignment functions and the semantic update conditions are:

**Definition 1 (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.

Whereas in *EDPL* contexts are identifed with states, we have to think about the notion of an ambiguous context. Here, we represent *ambiguous contexts*, $\sigma$, simply by sets of states, where each state in a set of states represents an unambiguous reading of the context. The notion of an update has to be redefined appropriately.

### 2.1 DPLA

Normally the semantics of underspecified representations is defined in terms of total disambiguations, see [8]. Avoiding redundancy is possible if we consider single sentences, but it returns as soon as we try to update contexts with ambiguous expressions, where we have to look at the update potential of each disambiguation.

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 sucessful updates.

[1] Institute for Logic, Language and Computation (ILLC), University of Amsterdam


ECAI 98. *13th European Conference on Artificial Intelligence Young Researcher Paper*
Edited by Henri Prade
Published in 1998 by John Wiley & Sons, Ltd.

**Definition 2** Let $UR$ be an underspecified representation where $UR = \langle V, C\rangle$. $\mathsf{top}(UR)$ returns the formula of the top node and $\mathsf{dt}(h)$ are the possible immediate subformulas of a hole $h$:

1. $\mathsf{top}(UR)$ = the $\varphi$ such that $l : \varphi \in V$ and for all $l'$ if $l' : \psi \in V$ then $C \vdash l' \leq l$
2. $\mathsf{dt}(h) = \{l : \varphi \in V \mid C \vdash l \leq h, \text{for no } l' : C \vdash l < l' < h\}$

$\sigma[\![UR]\!] = \{\langle s, C\rangle \mid \exists C' : \langle s, C'\rangle \in \sigma\}[\![\mathsf{top}(UR)]\!]$

$\sigma[\![h]\!] \;=$

$\bigcup_{l_j : \varphi_j \in \mathsf{dt}(h)} \{\langle s, C \cup \{l_k \leq h'\}\rangle \mid \langle s, C\rangle \in \sigma, l_k : \psi \in \mathsf{dt}(h)\}[\![\varphi_j[h']]\!]$

$\sigma[\![h]\!] \;= \sigma[\![\varphi_j]\!]$, if $\mathsf{dt}(h) = \{l_j : \varphi_j\}$ and no a hole in $\varphi_j$

The new thing is that the update is now parameterized with a set of ordering constraints on labels. We will say that $\sigma[\![\varphi]\!] = \emptyset$ if $C \vdash \bot$ .

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 is a set of pairs: $\langle s, C\rangle$.

Having defined the update function of underspecified representations, it is easy to 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 pairs of the form $\langle s, C\rangle$.

**Definition 3** if $\varphi$ does not contain holes or underspecified representations

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

This means that an expression which does not contain holes or underspecified representations does not effect the ordering constraints of the context to which it is applied.

**Definition 4 (Semantics of DPLA)** The update potentials of the logical connectives are those given in Definition 2 plus the following ones, where $s^c$ is of the form $\langle s, C\rangle$.

$$\begin{array}{ll}
\sigma[\![R(x_1..x_n)]\!] & = \{s^c[\![R(x_1..x_n)]\!] \mid s^c \in \sigma, s^c[\![R(x_1..x_n)]\!]\text{defined}\} \\
\sigma[\![x = y]\!] & = \{s^c[\![x = y]\!] \mid s^c \in \sigma, s^c[\![x = y]\!]\text{defined}\} \\
\sigma[\![\neg\varphi]\!] & = \{s^c - t^c \mid s^c \in \sigma, t^c \in \{s^c\}[\![\varphi]\!]\} \\
\sigma[\![\exists x\varphi]\!] & = \{s^c[\![\exists x\varphi]\!] \mid s^c \in \sigma, s^c[\![\exists x\varphi]\!]\text{defined}\} \\
\sigma[\![\varphi \wedge \psi]\!] & = \sigma[\![\varphi]\!][\![\psi]\!]
\end{array}$$

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

## 2.2 Properties of DPLA

*EDPL* is distributive and eliminative, but what about *DPLA*? Distributivity holds because in none of our definitions the update depends on the context as a whole. Combining this with the fact that *EDPL* is distributive we get:

**Observation 1** $\sigma[\![\varphi]\!] = \bigcup_{s^c \in \sigma} \{s^c\}[\![\varphi]\!]$ , 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.

**Observation 2** *DPLA* is eliminative.

**Definition 5 (Entailment)** Following [1], where $s \leq s'$ iff $\forall i \in s \exists j \in s' : i \subseteq j$ we say that $\sigma \leq \sigma'$ iff $\forall\langle s, C\rangle\forall\langle s', C'\rangle : s \leq s'$. 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. Therefore anaphora occuring in the conclusion cannot be bound by an existential quantifier in the premisses. 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 [6, 8] for further discussion and refinements.

## 3 Conclusion

Our formalism 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 to deduction in an ambiguous setting, cf. [2, 6]. Here, the notion of relative ambiguity is necessary in order to get a well behaved calculus for ambiguity, see [2]. This notion is part of our framework because updates depend on the context to which they are applied and so do ambiguous updates, too. The next step will be to see how a calculus for reasoning in ambiguous dynamic semantics can be developed.


## ACKNOWLEDGEMENTS

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


## REFERENCES


[1] P. Dekker, *Transsentential Meditations*, Ph.D. dissertation, ILLC / Department of Philosophy, University of Amsterdam, Amsterdam, 1993.

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

[3] J. Jaspars and M. Kameyama, 'Preferences in dynamic semantics', in *Proceedings of the 10th Amsterdam Colloquium*, eds., M. Stokhof and P. Dekker, pp. 445–464. ILLC, Amsterdam, (1996).

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

[5] H. S. Kurtzman and M. C. MacDonald, 'Resolution of quantifier scope ambiguities', *Cognition*, **48**, 243–279, (1993).

[6] C. Monz and M. de Rijke, 'A tableaux calculus for ambiguous quantification', in *Automated Reasoning with Analytic Tableaux and Related Methods*, TABLEAUX'98, ed., H. de Swart, LNAI 1397, pp. 232–246. Springer, (1998).

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

[8] U. Reyle, 'On reasoning with ambiguities', in *Proceedings of the 6th Meeting of the Association for Computational Linguistics, European Chapter*, pp. 1–8, Dublin, (1995).