Dynamic Semantics and Underspecification

Christof Monz

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 quantification and choice, as illustrated by the notorious example in (1)

(1) Every man loves a woman.

which has two readings:

(a) ∀x (man(x) → ∃y (woman(y) ∧ love(x, y))).

(b) ∃y (woman(y) ∧ ∀x (man(x) → 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 (2a) would be an appropriate semantic representation, but if (1) is followed by

(3) But she is already married.

(2a) 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],

\[
\begin{array}{ccc}
\emptyset & : & l_0 \\
\text{l}_1 & : & l_0 \\
\text{l}_2 & : & \exists y (\text{woman}(y) \land l_0) \\
\text{l}_3 & : & \text{love}(x, y) \\
\end{array}
\]

An underspecified representation (UR) is a pair \( (V, C) \), where \( V \) is the set of nodes, and \( C \) is a partial order on the labels \( l_0 \) and \( l_1 \) 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)

\[
V = \{ l_0, l_1, l_2, l_3 : \exists x (\text{man}(x) \rightarrow l_1), l_2 : \exists y (\text{woman}(y) \land l_2), l_3 : \text{love}(x, y) \} \text{ and } C = \{ l_1 \leq l_0, l_2 \leq l_0, l_3 \leq l_1, l_3 \leq l_2 \}.
\]

We neglect how a mapping from natural language sentences to underspecified representations can be defined, but the 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)**

\[
\begin{align*}
\text{s}[\text{if}(x_1, x_n)] &= \{ i \in s \mid (i(x_1), \ldots, i(x_n)) \in F(R) \} \text{ if } x_1, x_n \in D(s) \\
\text{s}[x = y] &= \{ i \in s \mid i(x) = i(y) \} \text{ if } x, y \in D(s) \\
\text{s}[\text{if} \varphi] &= s - s[i] \\
\text{s}[\exists x \varphi] &= s[i] \sqcup s \\
\text{s}[\varphi \land \psi] &= s[i] \sqcup s[i]
\end{align*}
\]

where \( s - s' = s \setminus \{ i \mid D(i) \in s' \} \) and \( s[i] = \{ j \mid \exists i \in s : i \leq s(i) \} \), such that \( i \leq x \) iff \( D(i) = D(x) \cup X \) and \( i = j \mid D(i) \).

The other connectives, \( \forall, \lor, \rightarrow \), can be defined in terms of the definitions given above.

Whereas in EDPL contexts are identified 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 successful updates.
Definition 2 Let \( UR \) be an underspecified representation where \( UR = (V, C) \), \( \text{top}(UR) \) returns the formula of the top
node \( \psi \) and \( \text{dt}(h) \) the immediate subformulas of a hole \( k \):

1. \( \text{top}(UR) = \varphi \) such that \( l : \varphi \in V \) and for all \( l' \) if
   \( l' : \varphi \in V \) then \( C \vdash l' \leq l \)
2. \( \text{dt}(h) = \{ l : \varphi \in V \ | \ C \vdash l < h, \text{for no } l' : C \vdash l < l' < h \} \)
3. \( \sigma[UR] = \{ (s, C) \ | \ \exists \sigma' : (s', C) \epsilon \sigma \}\text{top}(UR) \)
4. \( \sigma[h] = \bigcup_{k \in \sigma} \text{dt}(h) \)
5. \( \sigma[\varphi] \) if \( \text{dt}(h) = \{ l : \varphi \} \) and no a hole in \( \varphi \)

The new thing is that the update is now contextualized 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 dis-
ambiguities because they are not compatible with the given
input context. Partially updating with quantificationally am-
biguous 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 \( (s, C, \varphi) \).

Having defined the update function of underspecified repre-
sentations, 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 \((s, C)\).

Definition 3 if \( \varphi \) does not contain holes or underspecified representations
\( (s, C) [\varphi] = \{ \begin{array}{ll}
\emptyset [\varphi], & \text{if } s[\varphi] \neq \emptyset \text{ and defined, } C \vdash \bot \\
\emptyset, & \text{otherwise}
\end{array} \}

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 \( s' \) is of the form \( (s, C) \):

\[
\begin{align*}
\sigma[R(x_1\ldots x_n)] &= \{ s'[R(x_1\ldots x_n)] \ | \ s' \in \sigma \}, [R(x_1\ldots x_n)\text{defined}] \\
\sigma[x = y] &= \{ s'[x = y] \ | \ s' \in \sigma, [s'[x = y]\text{defined}] \\
\sigma[\neg \varphi] &= \{ s' \ | \ s' \in \sigma, t' \in \{ s' \}[\varphi] \} \\
\sigma[\exists x \varphi] &= \{ s' \exists x \varphi \ | \ s' \in \sigma, [s'[\exists x \varphi]\text{defined}] \\
\sigma[\varphi \land \psi] &= \sigma[\varphi][\psi]
\end{align*}
\]

where \( (s, C) \rightarrow (s', C) \rightarrow (s - t, C) \)

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

Observation 1 \( \sigma[\varphi] = \bigcup_{s' \in \sigma} \{ s'[\varphi] \} \), 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
\( \forall i \in \{ j \ | \ j \leq i \} \) we say that \( \sigma \leq \sigma' \) iff \( \forall (s, C) \exists \varphi \ (s, C) : \)
\( s \leq s' \). The entailment relation can now be defined as:

\[ \exists \varphi \ldots \varphi \psi \sigma[\varphi_1] \ldots [\varphi_n] \leq \sigma[\psi] \]

This is just one of many possible definitions of the entailment
relation. Two things should be pointed out: First, the dy-
namic force of the existential quantifier does not exceed the entailment
relation. Therefore anaphora occurring in the conclusion
cannot be bound by an existential quantifier in the premises. 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 ambi-
guous 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 formalism 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
[5] H. S. Kurtzman and M. C. MacDonald, 'Resolution of quanti-
[6] C. Mezz and M. de Rijke, 'A tableau calculus for am-
biguous quantification', in Automated Reasoning with Ar-
tifical Tableaux and Related Methods, TABLEAUX'98, ed.,
[7] U. Reyle, 'Dealing with ambiguities by underspecification:
Construction, representation, and deduction', Journal of Se-