PATRICK BLACKBURN, JOHAN BOS, MICHAEL KOHLHASE AND HANS DE NIVELLE

INFERENCE AND COMPUTATIONAL SEMANTICS

1. Introduction

In this paper we discuss inference in computational semantics. In particular, we argue that state-of-the-art methods in first-order *theorem proving* and *model generation* are of direct relevance to inference for natural language processing. This claim is based on our experience of implementing van der Sandt's approach to presupposition, and much of the paper discusses this application. Incidentally, the reader can experiment with this implementation over the internet: most of what is discussed below is available as part of Johan Bos's DORIS system (Discourse Oriented Representation and Inference System[1]).

This work has its roots in a textbook entitled *Representation and Inference in Natural Language: A First Course in Computational Semantics* (see Blackburn and Bos (2000a) for the latest draft). The goal of this book is straightforward: to present formal semantics from a computational perspective, and equip students with the basic tools required to perform semantic construction computationally. Modularity, reusability, and the use of standard tools is emphasized. Now, as far as *representation* is concerned, it is more or less clear what an introduction to computational semantics should offer: it is obviously sensible to introduce standard semantic representation formalisms such as Discourse Representation Theory (DRT Kamp and Reyle (1993)), to discuss well-known techniques for handling scope ambiguities, and so on. But *inference* is far harder to pin down. What exactly is inference in computational semantics?

Given the present state of knowledge, this is too difficult to answer: "inference" can mean just about anything from issues of architecture design (what information is available for immediate lookup, versus what is to be computed on the fly) to the use of probabilistic techniques. But in spite of this diversity, one topic should arguably play a key role: the use of first-order logic.

Theoretical considerations certainly suggest the importance of first-order inference. Many semantic representation formalisms can be reduced to first-

[1] `http://www.coli.uni-sb.de/~bos/doris`

H. Bunt et al. (eds.), Computing Meaning, Volume 2, 11–28.

order logic (this includes many formalisms which at first glance seem to lie beyond its reach, such as those which make use of partiality, or modal and temporal operators), and even when a full reduction is not possible, first-order logic often provides a useful approximation (a good example is the partial reduction of higher-order logic to first-order logic via generalized models). In particular, as we shall later see, there is a simple reduction from DRT to first-order logic. But first-order inference is not merely of theoretical interest: one of the main points we make in this paper is that it is becoming an increasingly *practical* option.

There is a large and active research community[2] devoted to exploring first-order inference computationally, and a wide range of sophisticated automated theorem provers, model builders, and other tools are now freely available over the internet.

In our view, computational semanticists should take note of these developments; off the shelf tools are now capable of playing a useful role in developing natural language systems with a non-trivial inferential component.[3]

We devote most of this paper to explaining why such tools are relevant to one particular problem: the computational treatment of presupposition. We are going to examine what is arguably one of the most natural (and certainly one of the most empirically successful) approaches to presupposition, namely van der Sandt's DRT based approach (Van der Sandt, 1992). We show how first-order inference techniques can be used to give a simple implementation of van der Sandt's ideas, and suggest that the resulting implementation gives a natural framework for exploring and refining his account. We extract a general lesson from our experiment, and conclude by discussing this.

Restrictions of space force us to assume a certain amount of background knowledge on the part of the reader. In particular, we assume familiarity with the rudiments of DRT (everything the reader needs can be found in the Kamp and Reyle textbook (Kamp and Reyle, 1993), or the first chapters

[2] In our experience, this community is interested in natural language applications, and is often prepared to try and accommodate its special needs. In fact, the proof problems generated by the DORIS system discussed below are currently finding their way into the CADE system competition (Sutcliffe and Suttner, 1997) as challenge problems to the automated theorem proving community.

[3] Of course, the idea of using first-order theorem proving techniques for NLP tasks is not new; it's as old as AI itself, and Allen 1995, for example, contains a good textbook level discussion. Nonetheless, few computational semanticists seem aware of developments in contemporary theorem proving and model generation, or of their potential relevance for computational semantics. We think such tools should be a standard part of the computational semanticist's arsenal.

of Blackburn and Bos (2000b)). Furthermore, while we sketch van der Sandt's method, we're going to focus on the inferential aspect of his work, thus it will be useful to have a copy of his classic article to hand; quite apart from its other merits, it's an excellent introduction to many issues in presupposition that we cannot discuss here.

2. Van der Sandt on Presupposition

Van der Sandt gives an *anaphoric* account of presupposition. That is, in his view presuppositions behave much like anaphoric pronouns—in fact the only difference is that presuppositions have more descriptive content. This simple idea has two important consequences. First, there is no need to give an account of presupposition 'cancellation', for there is no such phenomenon; what other accounts regard as a 'cancellation' is simply a case of a presupposition being successfully resolved to an antecedent. Second, because they have descriptive content, presuppositions are sometimes able to 'repair' the context by creating a suitable antecedent; this process is known as *accommodation*.

Van der Sandt expresses his theory in DRT; strictly speaking this is not necessary, but it is certainly advantageous to do so. DRSs are evolving discourse pictures; they display the previously established context, and grow as more information is added. Van der Sandt lets presuppositions contribute a new picture (that is, a new DRS) to this evolving representation, and demands that the new picture be *sensibly* incorporated into the overall representation. Two incorporation mechanisms are permitted. First, presuppositions can be *resolved*, just like ordinary pronouns. The beautiful point about this option is that it calls for no new apparatus: it simply makes use of familiar DRT mechanisms (such as accessibility) for pronoun resolution. Second, presuppositions can be *accommodated*; that is, they can repair the context by creating their own antecedent. Again, this fits beautifully with central ideas of DRT: because presuppositions are associated with DRSs, accommodation is essentially a matter of enlarging part of the picture.

Let's consider two examples, one illustrating resolution, the other accommodation. First some notation. Van der Sandt represents DRSs containing presupposed information by drawing them with dashed lines; we shall use the computationally more convenient convention of prefixing DRSs containing presupposed information with the symbol α (the mnemonic here is that a DRS marked with an α contains *anaphoric* information). We assume that presupposition triggers in the lexicon (such as the definite article, possessive constructions, and proper names) are associated with an appropriate α-DRS.

For our first example, suppose we have already processed the sentence 'A woman snorts'. That is, we have already built the following DRS:

(1)

y
WOMAN(y) SNORT(y)

Suppose the second sentence is 'The woman collapses'. According to van der Sandt, this is what happens. The second sentence, which contains the presupposition trigger 'the', gives rise to the following DRS:

(2)

<table>
<tr><td></td></tr>
<tr><td>α: <table><tr><td>x</td></tr><tr><td>WOMAN(x)</td></tr></table>
COLLAPSE(x)</td></tr>
</table>

(The best way to view this DRS is as an ordinary DRS—but an ordinary DRS marked as being unresolved with respect to presupposed information.) Next we merge this new DRS with the DRS that represents the previous discourse; note that this merging process takes place while the presuppositions are still unresolved. So after merging we obtain:

(3)

<table>
<tr><td>y</td></tr>
<tr><td>WOMAN(y)
SNORT(y)
α: <table><tr><td>x</td></tr><tr><td>WOMAN(x)</td></tr></table>
COLLAPSE(x)</td></tr>
</table>

Only after merging do we attempt to resolve the presuppositions. We recursively travel through the merged DRS and, for each α-marked DRS we encounter, we try to find a suitable 'anchor' to resolve to. That is, *we try to match the content of the* α*-DRS with that of superordinated DRSs*. Intuitively this is a natural thing to do; after all, presupposed information is supposed to be contextually available.

Let's see how this works. In our example, we only have one elementary presupposition:

(4)

x
WOMAN(x)

Note that if we identify the discourse referents x and y there is a partial match between the outermost DRS and the α-DRS. Carrying out this identification yields:

(5)

y
WOMAN(y) SNORT(y) COLLAPSE(y)

In short, we have successfully dealt with the presupposition induced by 'the', by identifying the discourse referent it introduced with the woman-denoting discourse referent in the preceding context.

That's the basic idea, but things don't always go this smoothly. Sometimes we *can't* find the presupposed information in the preceding context, and resolution is impossible. (Maybe, we missed a bit of a conversation; and anyway, people often have different views about what the assumed context actually is.) To deal with such cases van der Sandt makes use of *accommodation*: if we can't resolve our elementary presuppositions to a suitable element in the context, we don't give up. Instead we simply add the required background information.

Here's an example. Consider the sentence 'If Mia dates Vincent, then her husband is out of town'. Concentrating only on the trigger 'her husband', we get:

(6)

<table>
<tr><td>x y</td></tr>
<tr><td>x=MIA
y=VINCENT
<table><tr><td><table><tr><td></td></tr><tr><td>DATE(x,y)</td></tr></table></td><td>⇒</td><td><table><tr><td></td></tr><tr><td>α: <table><tr><td>z</td></tr><tr><td>HUSBAND(z,x)</td></tr></table>
OUT-OF-TOWN(z)</td></tr></table></td></tr></table>
</td></tr>
</table>

Assuming this is the first DRS we have to process (that is, that the DRS built up so far is still empty), there is no candidate DRS for matching the presupposed information that Mia has a husband, which is coded by the following DRS:

(7)

z
HUSBAND(z,x)

Springer

http://www.springer.com/978-1-4020-0175-8

Computing Meaning
Volume 2
Bunt, H.; Muskens, R.; Thijsse, E. (Eds.)
2001, VI, 306 p., Hardcover
ISBN: 978-1-4020-0175-8