Inference and Computational Semantics

Patrick Blackburn, Johan Bos, Michael Kohlhase
Universität des Saarlandes, Saarbrücken, Germany
{patrick,bos}@coli.uni-sb.de, kohlhase@cs.uni-sb.de

Hans de Nivelle
Centrum voor Wiskunde en Informatica,
Universiteit van Amsterdam, Amsterdam, The Netherlands
Hans.de.Nivelle@cwi.nl

Abstract

This paper discusses inference in computational semantics. We argue that state-of-the-art methods in first-order *theorem proving* and *model building* are of direct relevance to inference for natural language processing. We support our claim by discussing our implementation of van der Sandt's presupposition projection algorithm in Discourse Representation Theory, an algorithm which demands sustained use of powerful inference mechanisms.

Keywords: Computational Semantics, Theorem Proving, Model Building, Presupposition Projection, Discourse Representation Theory

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 building* 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 1998 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

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

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-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 now a large and active research community devoted to exploring first-order inference computationally, and a wide range of sophisticated 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.[2]

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 Chapters 7 and 8 of Blackburn and Bos 1998). 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]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 building, or of their potential relevance for computational semantics. We think such tools should be a standard part of the computational semanticist's arsenal.

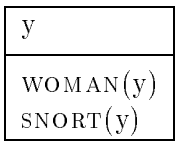
2 Van der Sandt on Presupposition

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*.

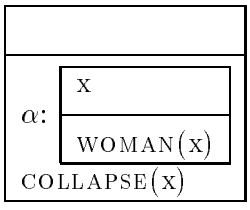
Van der Sandt expresses his theory in DRT; strictly speaking this is not necessary, but it is certainly advantageous to do so. DRSs are evolving discourse pictures; they display the previously established context, and grow as more information is added. Van der Sandt lets presuppositions contribute a 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.

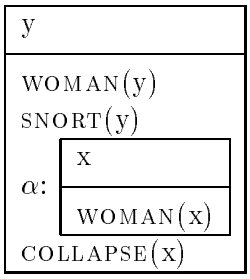
For our first example, suppose we have already processed the sentence 'A woman snorts'. That is, we have already built the following DRS:



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:

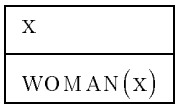


(The best way to view this DRS is as an ordinary DRS—but an ordinary DRS marked as being unresolved with respect to presupposed information.) Next we merge this new DRS with the DRS that represents the previous discourse; note that this merging process takes place while the presuppositions are still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

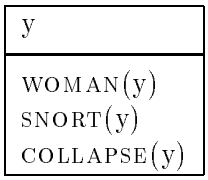


Only after merging do we attempt to resolve the presuppositions. We recursively travel through the merged DRS and, for each α-marked DRS we encounter, we try to find a suitable 'anchor' to resolve to. That is, *we try to match the content of the α-DRS with that of superordinated DRSs.* Intuitively this 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:



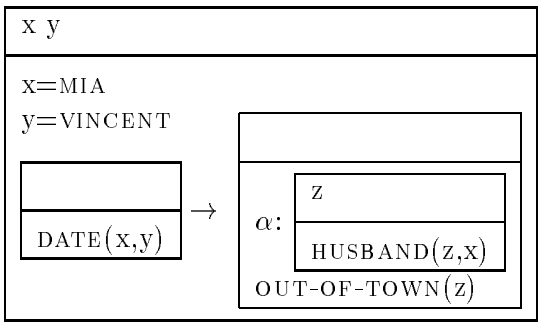
Note that if we identify the discourse referents x and y there is a partial match between the outermost DRS and the α-DRS. Carrying out this identification yields:



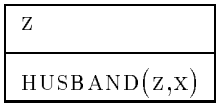
In short, we have successfully dealt with the presupposition induced by 'the', by identifying the discourse referent it introduced with the woman-denoting discourse referent in the preceding context.

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.

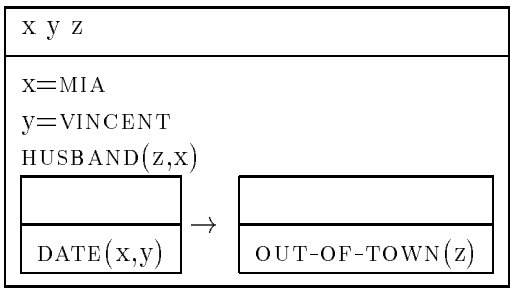
Here's an example. Consider the sentence 'If Mia dates Vincent, then her husband is out of town'. Concentrating only on the trigger 'her husband', we get:



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:



In such cases we accommodate the information to the outermost DRS, and get the following, final, DRS:



In broad terms, that is the way van der Sandt's approach works. But obviously more needs to be said: clearly both resolution and accommodation must be subject to constraints. And indeed they are. A more precise specification of van der Sandt's method is given by the following non-deterministic algorithm:

1. Generate a DRS for the input sentence with all elementary presuppositions given as α-DRSs.
2. Merge this DRS with the DRS of the discourse so far processed.
3. Traverse the DRS, and on encountering an α-DRS A try to
   (a) link the presupposed information to an accessible antecedent,
   (b) or accommodate the information to a superordinated DRS.
4. **Remove those DRSs from the set of potential readings that violate the acceptability constraints.**

Now, implementing the first three steps of this algorithm simply requires a slight modification of the basic DRT pronoun resolution algorithm (for further details, see Chapter 10, Blackburn and Bos 1998). But what of step 4? What are the acceptability constraints we must avoid violating?

One of these constraints, the free variable check, is rather uninteresting: we are not allowed to generate DRSs that contain free variables. But this is just a well-formedness condition on the resulting DRSs; it is easy to implement and we will not bother discussing it further. However there are also a number of far more interesting, essentially *semantic*, acceptability constraints—some of which are fully specified by van der Sandt, some of which are partially specified, and some of which are merely hinted at—which bring us, directly and unavoidably, to non-trivial inference problems. Most of this paper is devoted to discussing these semantic constraints and their implementation. Let's start by considering the two most clearcut constraints that van der Sandt imposes.

First, van der Sandt demands that contributions to a discourse be *consistent*. For example, the following discourses are unacceptable:

(1) Mia is a boxer. Vincent knows all boxers. Vincent doesn't know Mia.

(2) Jody is married. Jody does not have a husband.

Note that the inconsistency of these discourses is *not* a matter of pure logic; it depends on additional background information, namely that men are not women, that women are not men, that Jody is a woman, and that married women have husbands.

Second, van der Sandt demands that contributions to a discourse should be *informative*. That is, every contribution to the discourse should introduce new information. This rules out the following discourses:

(3) Jody is a boxer. Jody is a boxer.

(4) Mia is married. She has a husband.

Note that while the first inference is purely logical, the second inference hinges on our knowledge that Mia is a woman, and that married woman have husbands.[3]

Now, the consistency and informativity constraints are the simplest semantic constraints van der Sandt places on his algorithm (we defer discussion of the more complex *local* constraints till Section 4). But simple as they are to formulate, ensuring that they are met requires non-trivial inferential power: testing for consistency means we need a way of determining whether a given DRS can be embedded in some a model, while testing for newness means we need a way of determining whether one DRS follows from another. Moreover, we need to be able to carry out these tasks in a way that takes background knowledge into account.

What are we to do? One answer, of course, is to develop inference methods for the language of DRSs. Now, this is a theoretically sensible answer, and one that should certainly

[3]The motivation for both the consistency and informativity constraints should be familiar to most readers. Van der Sandt attributes his version of these ideas to Stalnaker.

be further explored.[4] Nonetheless, there are no good reasons for thinking that native DRT approaches will automatically lead to the most efficient implementations. There is a gap—and it is not a trivial one—between the existence of even sophisticated inference methods (for example, sequent calculi) and efficient implementations. *Efficient* theorem proving and model building is not simply a matter of starting with (say) a sequent calculus and applying a few routine programming tricks: it requires a sophisticated analysis of such issues as representation and proof search, and the existence of a complete proof calculus is merely the first step on a long and complex road. First-order inference techniques have had the benefit of extensive exploration by skilled researchers over a long period of time; it is hard for DRT and newer formalisms to compete with that, at least in the short term.

So let's hijack this first-order expertise! That is, let's attempt to make use of the many sophisticated first-order theorem proving and model building tools that are currently available by compiling inference problems involving DRSs into first-order inference problems. We explore this idea in the following section.

3 Exploiting First-Order Inference Tools

If we are to make use of first-order theorem provers and model builders, we have to do two things. First, we have to show how inference problems in DRT can be translated into inference problems in first-order logic. Luckily, as we shall shortly see, this first task is essentially trivial. Second we have to be precise about what the inference problems in DRT that van der Sandt appeals to actually are. For the consistency and informativity constraints introduced so far this is completely straightforward; but as we shall learn in the following section, the ideas underlying what we call van der Sandt's *local* constraints are not nearly so clear cut.

Let's first see how to translate DRSs to first-order formulas. The present implementation of DORIS makes use of the following translation, which is probably the standard one.[5] Here is the clause for boxes:

$$\left(\begin{array}{|c|}\hline \mathrm{x}_1, \ldots, \mathrm{x}_n \\ \hline \gamma_1 \\ \cdot \\ \cdot \\ \gamma_m \\ \hline \end{array}\right)^{fo} \quad = \quad \exists \mathrm{x}_1 \cdots \exists \mathrm{x}_n((\gamma_1)^{fo} \wedge \cdots \wedge (\gamma_m)^{fo})$$

[4] Interesting work already exists; for example (Reyle and Gabbay 1994; Monz and de Rijke 1998) discuss automated theorem proving calculi for DRT and DPL, and the latter is even implemented as a prototype system.

[5] This is the translation given in Kamp and Reyle (1993). A variety of other translations are known (see, for example, the translation given in Van Eijck and De Vries 1992 and Muskens 1996) and it would be interesting to experiment with these as well. One other piece of background information is worth knowing: it is straightforward to give a reverse translation from first-order logic to the language of DRSs (a very simple one is given in Chapter 7 of Blackburn and Bos 1998). This means that the language of DRSs has full first-order strength. An immediate consequence is that the consistency and informativity constraints are, in general, undecidable.

This maps the discourse referents to existentially quantified variables, and recursively translates the conditions. So now we must deal with the conditions. Basic conditions simply map to themselves-viewed-as-first-order-atomic-formulas:

$$\begin{aligned}(R(\mathrm{x}_1,\ldots,\mathrm{x}_n))^{fo} &= R(\mathrm{x}_1,\ldots,\mathrm{x}_n)\\ (\tau_1=\tau_2)^{fo} &= \tau_1=\tau_2\end{aligned}$$

Moreover, complex conditions formed using $\neg$ and $\vee$ are also straightforwardly handled; we simply push the translation function in over the connective, leaving the connective unchanged:

$$\begin{aligned}(\neg B)^{fo} &= \neg(B)^{fo}\\ (B_1 \vee B_2^{fo}) &= (B_1)^{fo} \vee (B_2)^{fo}\end{aligned}$$

Finally, complex conditions formed using $\rightarrow$ are translated as follows.

$$\left(\begin{array}{|l|}\hline \mathrm{x}_1,\ldots,\mathrm{x}_n\\ \hline \gamma_1\\ \cdot\\ \cdot\\ \gamma_m\\ \hline\end{array}\rightarrow B\right)^{fo} = \forall \mathrm{x}_1\cdots\forall \mathrm{x}_n(((\gamma_1)^{fo}\wedge\cdots\wedge(\gamma_m)^{fo})\rightarrow(B)^{fo})$$

There are two key points that need to be made about this translation. The first is semantic: a DRS can be satisfied in a given model using a given assignment if and only if its translation can be satisfied in that same model using the same assignment. It follows that a DRS is *valid*, *consistent* or *inconsistent* if and only if its first-order translation has the same property. In short, we *don't* lose anything of logical importance when we apply this translation. The second point is syntactic: the size of the translation is linear in the size of the input. That is, the computational overhead involved in translation is negligible. For a simple Prolog implementation of this translation, see Chapter 7 of Blackburn and Bos 1998.

So we can get from DRT to first-order logic with ease. What are the ramifications for van der Sandt's consistency and informativity constraints?

In fact what we've already said pretty much gives the answer: to check that a DRS is consistent, simply translate the DRS to first-order logic—call this formula Φ—and then use first-order inference tools to check whether Φ is consistent. Actually, this is an oversimplification: recall that we also want to take background knowledge into account. So let's assume that we have at our disposal a first-order knowledge base which contains the relevant background information; let KB be the conjunction of all the formulas it contains. Our consistency check needs to be performed relative to this background knowledge, which means we should use first-order tools to check whether $KB \wedge \Phi$ is consistent.

So how do we test for informativity? Well, if the new DRS follows from the DRS representing the previous discourse, together with the information stored in the knowledge base, then the new DRS does *not* encode new information. Let Ψ be the first-order formula $(KB \wedge OLD) \rightarrow NEW$, where OLD is the translation of the old DRS and NEW is the

translation of the new DRS. Then the new utterance is informative if and only if Ψ is *not* valid. So, van der Sandt's informativity test on DRSs simply boils down to the following task: use first-order inference tools to check whether Ψ is valid.

Summing up, we can compute van der Sandt's consistency and informativity constraints via our translation if there are practical tools for establishing the consistency and validity of first-order formulas. Of course, the first-order consistency and validity problems are undecidable, so there is no method guaranteed to work on all input—but are there methods which can be expected to work well in practice on the type of formulas typical linguistic examples yield? Our experience suggests that the answer is *yes*. *Sophisticated* theorem provers handle such input well (moreover, unsophisticated theorem provers handle it very badly) and it seems that the use of recent model building techniques can further enhance performance. Let's go into this a little deeper.

A (complete) first-order theorem prover is a tool that, given a valid first-order formula as input, will eventually (given enough memory and time) be able to determine that the formula really is valid; it does this by attempting to prove the formula using some proof calculus, typically resolution or tableaux. If a theorem prover proves a formula this is unequivocal evidence that the formula is valid. (On the other hand, if the theorem prover does not succeed in proving a formula after some finite time, this is does *not* mean that the formula is invalid; it may mean that not enough effort has been devoted to finding a proof.) Thus a theorem prover offers an important *positive* handle on validity; if it says a formula is valid it is correct. Moreover, it also offers a useful *negative* handle on consistency, for a formula is consistent if and only if its negation is valid. Thus if a theorem prover successfully proves $\neg\Phi$, this is unequivocal evidence that Φ is *not* consistent.[6]

However perhaps the most important fact about current theorem provers is the variety that are available and the speed many of them offer. Now, it is hard to say anything general about what is likely to constitute a good choice of theorem prover for natural language (beyond the fact that in general natural language applications will require theorem provers that handle equality, a stumbling block for many tableaux based systems). Indeed, we shall argue below that the best idea is not to choose at all; a better idea is to farm out the inference task to many different theorem provers (and model builders) simultaneously. So for now we'll merely mention that the present version of DORIS makes use of Hans de Nivelle's BLIKSEM, a prover optimized for dealing with the *guarded fragment* of first-order logic (Andréka, Németi, and Van Benthem 1998). An interesting feature of BLIKSEM is that it is a complete first-order theorem prover that actually *decides* the guarded fragment (De Nivelle 1998). In addition, it offers impressive performance on natural language examples. For technical details, including comparison with other theorem provers on this task, see Blackburn, Bos, Kohlhase, and de Nivelle 1998.

But theorem proving is not enough; model building is essential too. A model builder

[6] Actually, some theorem provers offer more than this choice between "Yes, valid!" and "I don't know!"; some can offer conclusive proof of invalidity for some input. For example, some theorem provers blend theorem proving and model building techniques. If such a system says that that a formula is *not* valid, this means it has definitive evidence of invalidity: it has constructed a countermodel. Moreover, some theorem provers — and the BLIKSEM theorem prover discussed below is one of them — are capable of determining that special kinds of input formulas are invalid.

is a tool that, given a first-order formula, attempts to build a model for that formula; if it succeeds, it thereby show that the input formula is consistent. Thus, whereas a theorem prover gives us a direct positive handle on validity, a model builder offers us a *partial* positive handle on satisfiability.[7]

Now this is an important and useful capability. For example, suppose we are using a theorem prover to test Φ for consistency (that is, we instruct it to try and prove $\neg\Phi$). Now, if the theorem prover succeeds, we know that Φ is not consistent, and nothing more needs to be said. But what if the theorem prover returns with the message "I can't prove this"? This *doesn't* mean that Φ is consistent, it merely means that the theorem prover has used up some predetermined quota of resources. Model building offers a partial solution to this problem: as well as calling the theorem prover with input $\neg\Phi$, simultaneously call the model builder with input Φ. In practice, this should successfully deal with many of the formulas the theorem prover can't handle, as is shown in Table 1. Here the top row lists possible responses from the theorem prover to $\neg\Phi$, while the left hand column lists possible responses of the model builder to Φ.

Table 1: Consistency checking for theorem provers and model builders

$\Phi \setminus \neg\Phi$	**valid**	?	**invalid**
satisfiable	–	consistent	consistent
?	inconsistent	?	consistent
not satisfiable	inconsistent	inconsistent	–

As yet, model building has not been incorporated in in DORIS. This is because, until recently, most model builders haven't been able to handle equality, and this is a liability for natural language applications. However model builders have recently appeared which handle first-order logic with equality and constant symbols, but *without* function symbols—exactly the kinds of formulas produced by the above translation to first-order logic; see Bry and Torge 1997; Bry and Torge 1998. Model building will play an increasingly important role in future versions of DORIS.

The current plan for DORIS is to generalize the idea of simultaneous use as far as possible. Don't bother trying to decide which is the best theorem prover for a problem, or whether a problem is best handled using theorem proving or model building—simply call on all available tools.[8] That is, we want to interface DORIS to a piece of middleware which will farm out the inference task to a wide range of theorem provers and model

[7] Be warned, while important, the grip on satisfiability offered by model builders is necessarily partial: the set of satisfiable first-order formulas is *not* recursively enumerable.

[8] This well-known technique is called competitive parallelism in automated theorem proving, and has been shown to provide super-linear speedups. That is, on average, n competing automated theorem provers find a proof of a given theorem in *less* than $1/n^{th}$ of the average proof time. In fact, the winner of the annual CADE automated theorem proving competition (Sutcliffe and Suttner 1997) is the GANDALF system that time-slices several theorem proving strategies in one system, imitating competitive parallelism.

builders over the internet. In effect, DORIS will start a race, and wait for the winner to report back.

4 The Local Constraints

The main message of the previous section is straightforward: there is a simple bridge between DRT and first-order logic, and once we are we have converted a DRS inference problem to a first-order inference problem we can use theorem proving and/or model building to resolve it. But now we encounter a difficulty. Van der Sandt imposes further constraints which we will call *local* constraints. Unfortunately, it is unclear from his description what the relevant DRS inference problem actually is, and until this is resolved we cannot reduce these constraints to first-order logic.

Roughly speaking, van der Sandt's locality constraints are as follows: superordinated DRSs should neither imply a subordinated DRS (we will term this local informativity), nor a negated subordinated DRS (this we call local consistency). The most important effect of the locality constraints is that they filter out certain (mostly global or intermediate) accommodation possibilities. To illustrate this, note that the following example does *not* presuppose that Mia has a husband (in this example, and in the ones that follow, the relevant presupposition trigger is underlined).

(5) If Mia is married, then her husband is out of town.

The local informativity constraints prevent global accommodation of the fact that Mia has a husband, because this information follows from the antecedent DRS.[9]

Van der Sandt does not precisely define the local principles; all he says about them is the following (slightly reformulated):

> The resolved DRS does not give rise to a structure in which (the negation of) some subordinate DRS is entailed by the DRSs which are superordinated to it. (Van der Sandt 1992, p. 367)

The problem lies in the interpretation of "the DRSs which are superordinated to it". To take a concrete example, let's say we have a DRS B_0 containing a conditional formed out of the DRSs B_1 and B_2:

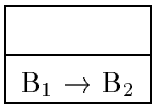
B_0:

$B_1 \rightarrow B_2$

In order to check the condition above, we first need to determine pairs that consist of a set of superordinated DRSs and a subordinated DRS. In the example above, there are

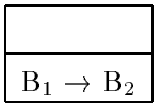
[9] This is nice, but in our view van der Sandt's local informativity constraints are too strong. For example, they rule out discourses such as 'Vincent eats a cheese-burger. If he eats a cheese-burger, he enjoys it.' and 'Vincent eats a burger. Every burger goes with a five dollar shake.' Incidentally, in DORIS local informativity is implemented as a "soft" constraint.

two such pairs: $\langle\{B_0\},B_1\rangle$, and $\langle\{B_0,B_1\},B_2\rangle$. The second assumption we have to make is that "entailed by the DRSs" means entailed by the merge of all these DRSs.[10] What this yields (using $\Rightarrow$ for entailment) is:



$\Rightarrow B_1$

for the pair $\langle\{B_0\},B_1\rangle$, and

$B_1 \oplus$ 

$\Rightarrow B_2$

for the instance $\langle\{B_0,B_1\},B_2\rangle$ (here $\oplus$ denotes merge of DRSs). The problem lies with the second entailment. Translating it into first-order logic, assuming that the respective translations for B_0, B_1, and B_2 are p, q, and, r, yields $q \wedge p \Rightarrow r$, and as p equals $p \rightarrow q$, we end up with $q \wedge (q \rightarrow r) \Rightarrow r$ which is a theorem. That means that the local informativity constraint is useless as it rules out *any* DRS with an implicational condition.

These problems are just a tip of the iceberg, but instead of further discussing the difficulties, let us turn straightaway to our positive proposal. We suggest this: redefine Van der Sandt's local constraints as follows.

Local Informativity
Let B be a DRS containing a subordinated DRS B_i, and B_1,...,B_n be subordinated DRSs contained in B which subordinate B_i. Then B_i is locally informative iff the DRS $B^* \oplus B_1 \oplus ... \oplus B_n$ does not entail B_i, where B^* is B minus the conditions that contain B_1,...,B_n,B_i.

Local Consistency
Let B be a DRS containing a subordinated DRS B_i, and B_1,...,B_n be subordinated DRSs contained in B which subordinate B_i. Then B_i is locally consistent iff the DRS $B^* \oplus B_1 \oplus ... \oplus B_n$ does not entail $\neg B_i$, where B^* is B minus the conditions that contain B_1,...,B_n,B_i.

The crucial change is that we have removed the conditions that contains the subordinated DRS. This enables us to avoid the problems noted above.

In our view, these conditions capture van der Sandt's intentions. Certainly their empirical predictions (as checked by DORIS), seem to coincide with Van der Sandt's expectations. Note that it is immediate which first-order problems these acceptability constraints correspond to: we merely form the required merges, translate, and determine whether the required entailments hold or not. Furthermore, note that computing these constraints places a heavy burden on the inference component: every sub-DRS gives rise to two calls

[10] This is the only interpretation that makes sense from a logical point of view. If it meant that each of the superordinated DRSs entailed the subordinated DRS, problems with free variables would appear immediately.

to the theorem prover and/or model builder. Indeed this is *precisely* why we find van der Sandt's algorithm such an interesting testbed for inference in computational semantics: it makes incessant non-trivial demands of any proposed inference mechanism.

Summing up, now have what we believe is a faithful implementation of van der Sandt's algorithm. In fact, this implementation transformed our views of the importance of first-order inference techniques for natural language processing. Prior to this, we tended to view such techniques as interesting, useful as teaching material, but ultimately as of questionable relevance. This view was swept away the first time we hooked up a theorem prover and saw how well it coped with van der Sandt's ideas. For a start, it was pleasant to be able to calculate van der Sandt's predictions instead of laboriously working them out by hand. More significantly, it swiftly became apparent that we now had a "presupposition laboratory" on our hands: we could explore van der Sandt's ideas to our hearts content, extending, refining, and just plain experimenting, with ease. For example, van der Sandt fleetingly mentions that linking an α-DRS A to a superordinated DRS B is allowed "such that the conditions of B are compatible with the conditions of A" (Van der Sandt 1992, p. 358), under a substitution that maps the discourse referents from A to B. In effect, he wants to allows (semantically justified) *partial matches* between the antecedent DRS and the DRS with the presupposed information. This idea can be implemented in our "presupposition laboratory", yielding a mechanism for coping with with direct bridges.

5 Concluding Remarks

We have argued that state-of-the-art theorem provers and model builders are of direct relevance to computational semantics. Our discussion has been, at best, preliminary: many difficult issues have not been addressed. Perhaps the most significant omission is any discussion of how well these methods will scale up (or, more accurately, what is lacking is discussion of what to do when these methods *fail* to scale up, for fail they certainly will with large enough discourses and knowledge bases). But in spite of this limitation, we believe that there is at least one application in which such methods have a good chance of playing an important role in the near future: the development of computational tools for exploring formal semantics. To close the paper, we'd like to explain why.

Richard Montague's work is arguably the most significant contribution to formal semantics since the pioneering work of Frege. Prior to Montague, formal semantics was based on analogies between natural and formal languages; Montague swept analogy aside and replaced it with the use of precisely specified translation algorithms. In our view, there are clear signs that the study of semantics is on the brink of taking a further step forward. This step (which we view as the natural fulfillment of Montague's vision) could be described as moving from Montague's method of fragments to what we like to call the *Method of Architectures*.

Over the next few years, progress in semantics is likely to depend on integrating the insights from diverse areas of inquiry. To put it another way, synthesis, not further analysis, is likely to be the key to significant progress. But with every passing day it seems clearer that the widespread availability of massive computing power and the internet will make

it possible to develop computational tools which can help semanticists to "think in bigger chunks". We certainly hope so, for in our view such tools could transform the study of semantics.

Now, such tools will probably develop by plugging together the best available components—from graphical interfaces to parsers—into ever more usable and flexible systems. Moreover, a key component of any such tool will be an inference mechanism (or inference mechanisms). Of course, it is impossible to predict what sort of inference mechanism might ultimately be favored; nonetheless, first-order inference is a likely candidate: it fits in well theoretically, and even if we assume the worst about the problem of scalability, this is far less likely to be crippling in a piece of software designed to help semanticists than it would in a more down to earth application.

Well—we'd certainly like to see things turn out this way and perhaps they will. In the meantime, let's start experimenting.

Acknowledgments

We are grateful to Claire Gardent and Karsten Konrad for discussion and pointers to the literature. We would also like to thank two anonymous reviewers for their comments on the first version of this paper. One of them offered us a theorem prover capable of model building. We are interested in finding out more—please unmask yourself!

References

Allen, J. (1995). *Natural Language Understanding* (2nd ed.). Benjamin/Cummings.

Andréka, H., I. Németi, and J. Van Benthem (1998). Modal languages and bounded fragments of predicate logic. *Journal of Philosophical Logic 27*(3), 217–274.

Blackburn, P. and J. Bos (1998, July). Representation and Inference for Natural Language. A First Course in Computational Semantics. Draft available at URL: `http://www.coli.uni-sb.de/~bos/comsem/`.

Blackburn, P., J. Bos, M. Kohlhase, and H. de Nivelle (1998). Automated Theorem Proving for Natural Language Understanding. Paper available at URL: `http://www.uni-koblenz.de/~peter/cade-15-ws/`.

Bry, F. and S. Torge (1997). Model Generation for Applications – A Tableaux Method Complete for Finite Satisfiability. Research Report PMS-FB-1997-15, LMU Institut für Informatik, München.

Bry, F. and S. Torge (1998). A deduction method complete for refutation and finite satisfiability. In *Proc. 6th European Workshop on Logics in AI (JELIA)*. Springer LNAI.

De Nivelle, H. (1998). A resolution decision procedure for the guarded fragment. In *CADE 15*. Springer Verlag.

Kamp, H. and U. Reyle (1993). *From Discourse to Logic*. Dordrecht: Kluwer.

Monz, C. and M. de Rijke (1998). A resolution calculus for dynamic semantics. In *Proceedings of the 6th Eurpopean Workshop on Logics in AI (JELIA'98)*, LNAI. Springer.

Muskens, R. (1996). Combining montague semantics and discourse representation. *Linguistics and Philosophy 19*, 143–186.

Reyle, U. and D. M. Gabbay (1994). Direct deductive computation on discourse representation structures. *Linguistics & Philosophy 17*, 343–390.

Sutcliffe, G. and C. Suttner (1997). The results of the cade-13 atp system competition. *Journal of Automated Reasoning 18*(2), 259–264. Special Issue on the CADE-13 Automated Theorem Proving System Competition.

Van der Sandt, R. A. (1992). Presupposition Projection as Anaphora Resolution. *Journal of Semantics 9*, 333–377.

Van Eijck, J. and F.-J. De Vries (1992). Dynamic interpretation and hoare deduction. *Journal of Logic, Language and Information 1*(1), 1–44.

Weidenbach, C., B. Gaede, and G. Rock (1996). Spass & flotter, version 0.42. In M. McRobbie and J. Slaney (Eds.), *Proceedings of the 13th Conference on Automated Deduction*, Number 1104 in LNAI, New Brunswick, NJ, USA. Springer Verlag.