In: H. Bunt & R. Muskens (eds.), Computing Meaning, Studies in Linguistics Philosophy Series, Kluwer, Dordrecht, 1999.

PRESUPPOSITION PROJECTION as PROOF CONSTRUCTION

Emiel Krahmer & Paul Piwek

1 Introduction

Van der Sandt's (1992) anaphoric account of presupposition is generally considered to be the theory which makes the best empirical predictions about presupposition projection (see e.g., Beaver 1997:983). The main insight is that there is an interesting correspondence between the behavior of anaphoric pronouns in discourse and the projection of presuppositions in complex sentences. Van der Sandt proposes to 'resolve' presuppositions just like anaphoric pronouns are resolved in *Discourse Representation Theory* (DRT, Kamp & Reyle 1993). Van der Sandt contends that there is also an important *difference* between pronouns and presuppositions: when there is no antecedent for an anaphoric pronoun, the sentence containing the pronoun cannot be interpreted. However, when there is no antecedent for a presupposition – and the presupposition has sufficient descriptive content – then the presupposition can be accommodated and, as it were, create its own antecedent. This combination of resolution and accommodation constitutes the empirical strength of Van der Sandt's approach.

A problem with Van der Sandt's approach is that it does not take the influence of world knowledge into account (see e.g., Beaver 1995: 64-66). Consider:

(1) a. If John is married, his wife probably walks the dog.

In: H. Bunt & R. Muskens (eds.), Computing Meaning, Studies in Linguistics Philosophy Series, Kluwer, Dordrecht, 1999.

b. If John buys a car, he checks the motor first.

c. If Spaceman Spiff lands on planet X, he will be annoyed by the fact that his weight is higher than it would be on earth. (Beaver 1995)

Example (1.a) contains a definite description, *"his wife"*, which triggers the presupposition that John has a wife. For the correct treatment of this example, a rather trivial piece of world knowledge is needed: if a man is married, he has a wife. But, if we do not take this piece of world knowledge into account, the theory of Van der Sandt (1992) is not able to treat being *"married"* as an 'antecedent' for the presupposition triggered by *"his wife"*. Being married creates an (implied) antecedent for *"his wife"*. A more substantial usage of world knowledge is required for example (1.b), which is an example of the notorious *bridging phenomenon* (Clark 1975). The description *"the motor"* presupposes the existence of a motor. Since there is no proper antecedent for this definite description, the theory of Van der Sandt (1992) predicts that the presupposition is accommodated. But this fails to do justice to the intuition that the mentioning of a car somehow licenses the use of *"the motor"* and that the motor is part of the car which John buys. Example (1.c) also illustrates the need for world knowledge. The *"the fact that S"* construction presupposes S; thus the consequent of (1.c) presupposes that Spaceman Spiff's weight is higher than it would be on earth. Since there is no obvious way to bind this presupposition, Van der Sandt's account predicts that it is accommodated.

The claim that world knowledge has an influence on presupposition projection is hardly revolutionary. For instance, Van der Sandt seems to assume that world knowledge somehow influences presupposition projection (Van der Sandt 1992:375, fn. 20), but he gives no clues on how world knowledge interacts with his theory of presupposition. The central question addressed in this chapter is *how* to account for the influence of world knowledge on presup-

In: H. Bunt & R. Muskens (eds.), Computing Meaning, Studies in Linguistics Philosophy Series, Kluwer, Dordrecht, 1999.

position projection. We argue that employing a class of mathematical formalisms known as *Constructive Type Theories* (CTT, see e.g., Martin-Löf 1984, Barendregt 1992) allows us to answer this question. To do so, we reformulate Van der Sandt's theory in terms of CTT. CTT differs from other proof systems in that for each proposition which is proven, CTT also delivers a proof-object which shows *how* the proposition was proven.[1] As we shall see, the presence of these proof-objects is useful from the presuppositional point of view. Additionally, CTT contexts contain *more* information than is conveyed by the ongoing discourse, and there is a formal interaction between this 'background knowledge' and the representation of the current discourse. This means that the reformulation of Van der Sandt's theory in terms of CTT is not just a nice technical exercise, but actually creates interesting new possibilities where the interaction between presupposition resolution and world knowledge is concerned.[2]

2 Presuppositions as Anaphors

Van der Sandt (1992) proposes to *resolve* presuppositions, just like anaphoric pronouns are resolved in DRT. For this purpose he develops a meta-level resolution algorithm. The input of this algorithm is an underspecified *Discourse Representation Structure*

[1] For us, the *constructive* aspect resides in the explicit construction of proof-objects; we are not committed to an underlying intuitionistic logic.

[2] In spirit, our work is related to Ahn (1994), Beun & Kievit (1995) and Krause (1995). Krause presents a type-theoretical approach to presuppositions. His system not only allows binding of presuppositions, but also has the possibility to globally accommodate them using an abductive inferencing mechanism. One important difference with our approach is that we take the entire theory of Van der Sandt (including intermediate and local accommodation) and rephrase it in terms of CTT. Ahn and Beun & Kievit use CTT for dealing with the resolution of definite expressions. The latter focus on selecting the right referent (which may be found in the linguistic context, but also in the physical context) using concepts such as prominence and agreement.

In: H. Bunt & R. Muskens (eds.), Computing Meaning, Studies in Linguistics Philosophy Series, Kluwer, Dordrecht, 1999.

(DRS), which contains one or more unresolved presuppositions. When all these presuppositions have been resolved, a proper DRS remains, which can be interpreted in the standard way.[3] Consider (2), and its Van der Sandtian representation, (DRS 1):

(2) If a Chihuahua enters the room, the dog snarls.

(DRS 1)

$$\begin{array}{|c|}\hline \begin{array}{|l|}\hline x \\ \hline Chihuahua(x) \\ enter(x) \\ \\ \hline \end{array} \Longrightarrow \begin{array}{|l|}\hline \\ \hline snarl(y) \\ \qquad \begin{array}{|l|}\hline y \\ \hline dog(y) \\ \hline \end{array} \\ \hline \end{array} \\ \hline \end{array}$$

The definite description *the dog* presupposes the existence of a dog. Van der Sandt models this by adding an embedded, presuppositional DRS to the representation of the consequent (expressing that there is a dog). To resolve the presuppositional DRS, we do what we would do to resolve a pronoun: look for a suitable, accessible antecedent. In this case, we find one: the discourse referent x introduced in the antecedent *is* accessible, and suitable since a Chihuahua is a dog. As said above, it is unclear *how* this information can be employed in Van der Sandt's theory, but for now let us simply assume that we can *bind* the presupposition. The presuppositional DRS is removed, and the y in the condition $snarl(y)$ is replaced with the newly found antecedent: x.

(DRS 2)

$$\begin{array}{|c|}\hline \begin{array}{|l|}\hline x \\ \hline Chihuahua(x) \\ enter(x) \\ \hline \end{array} \Longrightarrow \begin{array}{|l|}\hline \\ \hline snarl(x) \\ \\ \hline \end{array} \\ \hline \end{array}$$

[3]In Krahmer (1995), Van der Sandt's theory is combined with a version of DRT with a *partial* interpretation. In this way, DRSs which contain unresolved presuppositions can also be interpreted, which is shown to have several advantages.

In: H. Bunt & R. Muskens (eds.), Computing Meaning, Studies in Linguistics Philosophy Series, Kluwer, Dordrecht, 1999.

Anaphoric pronouns need to be bound. For presuppositions this is different: they can also be accommodated, provided the presupposition contains sufficient descriptive content. Reconsider example (2): on Van der Sandt's approach (globally) *accommodating* the presupposition associated with *the dog* amounts to removing the presuppositional DRS from the consequent DRS and placing it in the main DRS, with (DRS 3) as result.

(DRS 3)

$$\begin{array}{|l|}\hline y \\ \hline dog(y) \\ \begin{array}{|l|}\hline x \\ \hline Chihuahua(x) \\ enter(x) \\ \hline \end{array} \Longrightarrow \begin{array}{|l|}\hline \\ \hline snarl(y) \\ \\ \hline \end{array} \\ \\ \hline \end{array}$$

This DRS represents the 'presuppositional' reading of (2), which may be paraphrased as *"there is a dog and if a Chihuahua enters, the aforementioned dog snarls"*.[4] Now we have *two* ways of dealing with the presupposition in example (2), so the question may arise which of these two is the 'best' one. To answer that question, Van der Sandt (1992:357) gives some general rules for preferences, which may be put informally as follows: *1.* Binding is preferred over accommodation, *2.* Accommodation is preferred as high as possible, *3.* Binding is preferred as low as possible. Thus: according to Van der Sandt (DRS 2) (the 'binding' reading) is preferred over (DRS 3) (the 'accommodation' reading).[5] The second preference rule suggests that there is more than one way to accommodate a presupposition, and indeed there is. Consider:

[4] This DRS (as the previous ones) are presented in the usual 'pictorial' fashion. Below we also use a linear notation which we trust to be self-explanatory. For example, in this linear notation the current DRS looks as follows: $[y|\ dog(y),\ [x \mid Chihuahua(x), enter(x)] \Longrightarrow [\ |snarl(y)]]$.

[5] It has been argued that examples like (2), in which there is a *partial match* between anaphor and antecedent are ambiguous between a binding and an accommodation reading. See e.g., Krahmer & Van Deemter (1997) for an analysis of partial match ambiguities. Here we will ignore this issue.

In: H. Bunt & R. Muskens (eds.), Computing Meaning, Studies in Linguistics Philosophy Series, Kluwer, Dordrecht, 1999.

(3) It is not true that I feed John's Chihuahua, since he doesn't have one!

Here global accommodation of the presupposition triggered by *John's Chihuahua* yields an inconsistent DRS. This is prohibited by one of Van der Sandt (1992:367)'s conditions on accommodation. Therefore the presupposition is accommodated *locally*, i.e., within the scope of the negation.

In the next section, we discuss CTT and show how Van der Sandt's approach can be rephrased in terms of it. In the Section thereafter, we will see how the examples in (1), which are problematic for Van der Sandt's approach as it stands, can be dealt with. We believe that the CTT approach leads to better results than adding a proof-system to DRT, as done in e.g., Saurer (1993). The main advantage of CTT is that it is a standard proof system developed in mathematics with well-understood meta-theoretical properties (see Ahn & Kolb (1990) for discussion on the advantages of reformulating DRT in CTT). Moreover, the presence of explicit proof-objects turns out to have some additional advantages for our present purposes.

3 The Deductive Perspective

We introduce CTT by comparing it with DRT; this comparison is based on Ahn & Kolb (1990), who present a formal translation of DRSs into CTT expressions. In CTT, a context is modelled as an ordered sequence of introductions. Introductions are of the form $V:T$, where V is a variable and T is the type of the variable. Consider example (4.a) and its DRT representation (4.b) (in the linear notation, cf. footnote 4).

(4) a. A dog snarls.

b. $[x \mid dog(x), snarl(x)]$

A discourse referent can be modelled in CTT as a variable. A referent is added to the context by means of an introduction which not only adds the variable but also fixes its type. We choose *entity* as the type of discourse referents. Thus, we add $x : entity$ to the context. *entity* itself also requires introduction. Since *entity* is a type, we write: $entity : type$.

In general, a type T can only be used after the type of T itself (or the parts of which T has been composed) has been specified in the context with an introduction (e.g., $T : T'$). However, the introduction of the aforementioned type *type* is not carried out in the context; it is taken care of by an axiom which says that $type : \Box$ (where $\Box$ is to be understood as the 'mother' of all types) can be derived in the empty context ($\epsilon \vdash type : \Box$).

DRT's conditions correspond to introductions $V : T$, where T is of the type *prop* (short for proposition, which comes with the following axiom: $\epsilon \vdash prop : \Box$). For instance, the introduction $y : (dog \cdot x)$ corresponds to the condition $dog(x)$. The type $dog \cdot x$ (of type *prop*) is obtained by applying the type *dog* to the object x. Therefore, it depends on the introductions of x and *dog*. Since $dog \cdot x$ should be of the type *prop*, *dog* must be a (function) type from the set of entities into propositions, i.e., $dog : entity \rightarrow prop$.

The introduction $y : dog \cdot x$ involves the variable y (of the type $dog \cdot x$). The variable y is said to be an inhabitant of $dog \cdot x$. Curry and Feys (1958) came up with the idea that propositions can be seen as classifying proofs (this is known as the *propositions as types – proofs as objects* interpretation). This means that the aforementioned introduction states that there is a proof y for the proposition $dog \cdot x$. The second condition of (4.b), $snarl(x)$, can be dealt with along the same lines (this yields $z : snarl \cdot x$). Thus, the CTT counterpart to the DRS (4.b) contains the following three introductions: $x : entity, y : dog \cdot x, z : snarl \cdot x$.

In: H. Bunt & R. Muskens (eds.), Computing Meaning, Studies in Linguistics Philosophy Series, Kluwer, Dordrecht, 1999.

Dependent Function Types In DRT, the proposition *"Everything sucks"* is translated into A DRS containing the implicative condition $[x \mid thing(x)] \Longrightarrow [\,\mid suck(x)]$. In CTT, this proposition corresponds to the type $(\Pi x\!:\!entity.suck\cdot x)$, which is a dependent function type. It describes functions from the type *entity* into the type $suck\cdot x$. The range of such a function $(suck\cdot x)$ depends on the object x to which it is applied. Suppose that we have an inhabitant f of this function type, i.e., $f : (\Pi x : entity.suck\cdot x)$. Then we have a function which, when it is applied to an arbitrary object y, yields an inhabitant of the proposition $suck\cdot y$. Thus, f is a constructive proof for the proposition that *Everything sucks.*

Of course, function types can be nested. Consider the predicate *"snarl"*. We suggested to introduce it as a function from entities to propositions. One could, however, argue that *"snarl"* is a predicate which only applies to dogs. In that case, it would have to be introduced as a function from entities to another function, i.e., the function from a proof that the entity is a dog to a proposition, that is $snarl\!:\!(\Pi x\!:\!entity.(\Pi p\!:\!dog\cdot x.prop))$. We will abbreviate this as $snarl\!:\!([x\!:\!entity, p\!:\!dog\cdot x] \Rightarrow prop)$.

Inference The core of CTT consists of a set of derivation rules with which one can determine the type of an object in a given context. These rules are also suited for searching for an object belonging to a particular type. There is, for instance, a rule which is similar to modus ponens in propositional logic (in the rule below, $T[x := a]$ stands for a T such that all free occurrences of x in T have been substituted by a. Furthermore, $\Gamma \vdash E\!:\!T$ means that in context Γ, the statement $E\!:\!T$ holds):

$$\frac{\Gamma \vdash F\!:\!(\Pi x\!:\!A.B) \quad \Gamma \vdash a\!:\!A}{\Gamma \vdash F\cdot a\!:\!B[x := a]}$$

For instance, if a context Γ contains the introductions $b : entity$ and $g : (\Pi y : entity.suck\cdot y)$ (Everything sucks), then we can use

In: H. Bunt & R. Muskens (eds.), Computing Meaning, Studies in Linguistics Philosophy Series, Kluwer, Dordrecht, 1999.

this rule to find an inhabitant of the type $suck \cdot b$. In other words, our goal is to find a substitution S such that $\Gamma \vdash P : suck{\cdot}b[S]$. The substitution S should assign a value to P. P is a so-called *gap*. A CTT expression with a gap is an underspecified representation of a proper CTT expression: if the gap is filled, then a proper CTT expression is obtained. The deduction rule tells us that $(g \cdot b)$ can be substituted for P, if $\Gamma \vdash g : (\Pi y : entity.suck \cdot y)$ and $\Gamma \vdash b : entity$. Both so-called judgements are valid, because we assumed that $g : (\Pi y : entity.suck{\cdot}y)$ and $b : entity$ are members of Γ. Thus, we can conclude that $\Gamma \vdash (g{\cdot}b) : suck{\cdot}b$.

Presuppositions as Gaps A DRS is the end product of the interpretation of a sentence with respect to a main DRS. Ahn & Kolb (1990) show that this end product can be translated into a corresponding CTT context. Van der Sandt's presuppositional DRSs can be seen as a kind of proto DRSs of which the presuppositional representations have not yet been resolved. Only after binding and/or accommodation of the presuppositional representations a proper DRS is produced. Analogously, in CTT terms, a construction algorithm could translate a sentence into a proto type before a proper type (of the type *prop*) is returned.[6] This proper type (i.e., proposition) can then be added to the main context by introducing a fresh proof for it. For example, this is the appropriate proto type for example (2):[7]

(5) $[x : entity, y : chihuahua{\cdot}x, z : enters{\cdot}x] \Rightarrow$
$(snarl{\cdot}Y)_{[Y:entity, P:dog\cdot Y]}$

[6] We assume that one sentence translates into one type. The attentive reader may wonder how this agrees with our earlier translation of (4.a). In fact, it corresponds to the following single introduction: $g : (\Sigma x : entity.(\Sigma y : dogx.snarlx))$, given some appropriate, standard derivation rules (e.g., Martin-Löf 1984, Ranta 1994).

[7] Recall that this abbreviates $\Pi x : entity.(\Pi y : chihuahua \cdot x.(\Pi z : enters \cdot x.(snarl{\cdot}Y_{[Y:entity,P:dog\cdot Y]})))$.

In: H. Bunt & R. Muskens (eds.), Computing Meaning, Studies in Linguistics Philosophy Series, Kluwer, Dordrecht, 1999.

Thus, if x is an entity, y is a proof that x is a Chihuahua and z is a proof that x enters, then there exists a proof that Y snarls, where Y is a gap to be filled by an entity for which we can prove that it is a dog.[8] The presuppositional annotation consists of a sequence of introductions with gaps.[9]

Filling the Gaps Before we can evaluate the CTT representation (5) given some context Γ, we first have to resolve the presupposition by filling the gaps. For this purpose, we have developed an algorithm (sketched in the appendix) which can be seen as a re-implementation of Van der Sandt's resolution algorithm, but now operating on CTT expressions. The first thing we do after starting the resolution process, is try to fill the gap by 'binding' it. The question whether we can bind the presupposition triggered by *"the dog"* in example (2) can be phrased in CTT as follows: is

[8] The notion of gaps can also be applied to the analysis of questions in CTT (Piwek 1997). A question introduces gaps, which can be filled by extending the context of interpretation with the answer provided by the dialogue participant. A question is answered, when the associated gaps can be filled.

[9] To be complete, let us give the *syntactic* definition of proto types. For that we need the definition of a proper type:

$$T \quad :== \quad V \mid type \mid prop \mid \square \mid (\Pi V : T.T) \mid (\lambda V : T.T) \mid (T \cdot T).$$

A proto type T' can be obtained by substituting gaps (G) for one or more of the types of some proper type T. The result is a Type with Gaps (TG). An annotation has to be attached to T (with gaps) for specifying the types of the gaps. A TG with one or more annotations (A) is a Proto Type (PT).

$$\begin{array}{lll} TG & :== & G \mid V \mid type \mid prop \mid \square \mid (\Pi V : TG.TG) \mid (\lambda V : TG.TG) \mid \\ & & (TG \cdot TG) \\ A & :== & TG : TG \mid A \otimes A \\ PT & :== & TG \mid PT_A \mid (\Pi V : PT.PT) \mid (\lambda V : PT.PT) \mid (PT \cdot PT) \end{array}$$

$P \otimes Q$ represents the concatenation of sequences P and Q (often written as P, Q). Notice that the definition permits annotation of expressions which are already annotated. This is required for representing embedded presuppositions.

In: H. Bunt & R. Muskens (eds.), Computing Meaning, Studies in Linguistics Philosophy Series, Kluwer, Dordrecht, 1999.

there a substitution S such that the following can be proven?[10,11]

(6) $\Gamma, x\!:\!entity, y\!:\!chihuahua\cdot x, z\!:\!enter\cdot x \vdash_{\Delta}$
$(Y\!:\!entity, P\!:\!dog\cdot Y)[S]$

In words: is it possible to prove the existence of a dog from the global context Γ extended with the local context (the antecedent of the conditional)? The answer is: that depends on Γ. Suppose for the sake of argument that Γ itself does not introduce any dogs, but that it does contain the information that a Chihuahua is a dog. Technically, this means that (7) is a member of Γ:

(7) $f\!:\!([a\!:\!entity, b\!:\!chihuahua\cdot a] \Rightarrow (dog\cdot a))$

Given this function, we find a substitution S for (6), mapping Y to x and P to $(f\cdot x\cdot y)$ (which is the result of applying the aforementioned function f to x and y).[12] So we fill the gaps using the substitution S, remove the annotations (which have done their job) and continue with the result:

(8) $[x\!:\!entity, y\!:\!chihuahua\cdot x, z\!:\!enter\cdot x] \Rightarrow (snarl\cdot x)$

Thus, intuitively, if an interpreter knows that a Chihuahua is a dog, she will be able to bind the presupposition triggered by the definite *"the dog"* in (2). Now suppose the interpreter does not know that a Chihuahua is a dog or is of the opinion that Chihuahuas simply are not 'proper' dogs. That is, Γ does not contain a function mapping Chihuahuas to dogs. Then, still under the assumption that Γ does not introduce any dogs, the interpreter will not be able to prove the existence of a dog. She can then try to

[10] In general: $\Gamma \vdash_{\Delta} C_1, \ldots, C_n$ abbreviates $\Gamma \vdash C_1, \ldots, \Gamma \vdash C_n$.

[11] Interestingly, Zeevat (1992) compares the Van der Sandtian resolution of a presupposition with answering a 'query' in PROLOG, requiring the instantiation of a variable.

[12] The $\cdot$ (representing function application) is left-associative, thus $f\cdot x\cdot y$ should be read as $(f\cdot x)\cdot y$.

accommodate the existence of a dog by replacing the gaps Y and P with fresh variables, say y' and p', and extending the context Γ with $y' : entity, p' : dog \cdot y'$. Of course, it has to be checked whether this move is *adequate*, whether the result of accommodation is consistent and informative.[13] For more details on the resolution algorithm (also of intermediate[14] and – our alternative for – local accommodation) the reader is referred to the appendix.

4 Using World Knowledge

Bridging From our perspective, bridging amounts to using world knowledge to fill gaps. Consider example (1.b) again, with its CTT representation given in (9).

(9) $[x : entity, y : car \cdot x, z : buy \cdot x \cdot j] \Rightarrow$
$(check \cdot Y \cdot j)_{[Y:entity, P:motor \cdot Y]}$

Before we can add this expression to some context Γ, we have to resolve the presuppositional expression. We first search for a substitution S such that (10) can be proven:

(10) $\Gamma, x : entity, y : car \cdot x, z : buy \cdot x \cdot j \vdash_{\Delta}$
$(Y : entity, P : motor \cdot Y)[S]$

When can *"the motor"* be understood as a bridging anaphor licensed by the introduction of a car? If the interpreter knows that a car has a motor. Modelling this knowledge could go as follows: Γ contains two functions: one function which maps each car to an entity, $f : ([a : entity, b : car \cdot a] \Rightarrow entity)$, and one function which states that this entity is the car's motor $g : ([a : entity, b : car \cdot a] \Rightarrow$

[13]For more information of the background and formalization of these constraints see Van der Sandt (1992:367-369).

[14]Intermediate accommodation is not entirely uncontroversial. For instance, it has been argued that the 'intermediate readings' are achieved in a different way, e.g., by quantificational restriction (see e.g., Beaver 1995).

$(motor{\cdot}(f{\cdot}a{\cdot}b))$. Using these two functions, we find a substitution S in (10), mapping Y to $f{\cdot}x{\cdot}y$ and P to $g{\cdot}x{\cdot}y$. We can look at the resulting proof-objects as the 'bridge' which has been constructed by the interpreter; it makes the link with the introduction of a car explicit (by using x and y) and indicates which inference steps the user had to make to establish the connection with the motor (by using the functions f and g). So, we can fill the gaps, assuming that the proofs satisfy certain conditions. Of course, they have to satisfy the usual Van der Sandt conditions. Additionally, the bridge itself has to be 'plausible'. What plausibility exactly is, is beyond the scope of this chapter (but see Section 5). We would like to point out, however, that the presence in CTT of explicit proof-objects indicating precisely which pieces of knowledge have been used, facilitates plausibility-checking. For example, we contend that the complexity of the proof-object is inversely proportional to the plausibility of the bridge.[15]

Let us now consider a somewhat more complex example.

(11) John walked into the room. The chandelier shone brightly. (after Clark 1975)

Assume that the first sentence of (11) has already been processed, which means that the context Γ at least contains the following introductions: $x : entity, y : room \cdot x, z : walk\_in \cdot x \cdot j$. Now, we encounter the CTT representation of the second sentence:

(12) $q : shine{\cdot}Y_{[Y:entity, P:chandelier \cdot Y]}$

We want to resolve the presupposition triggered by *"the chandelier"* in the context Γ (assuming that Γ does not introduce any chandeliers). When would an interpreter be able to link *the chandelier* to the room John entered? Of course, it would be easy if

[15]For a given proof-object we can determine which atomic proof-objects from the context have been used and how many times. Thus, in the aforementioned $f{\cdot}x{\cdot}y$ three atomic proofs are used, namely f, x and y.

In: H. Bunt & R. Muskens (eds.), Computing Meaning, Studies in Linguistics Philosophy Series, Kluwer, Dordrecht, 1999.

she had some piece of knowledge to the effect that every room has a chandelier (if her Γ would contain functions which for each room produce a chandelier). However, such knowledge is hardly realistic; many rooms do not have a chandelier.

In a more realistic scenario, the following might happen. The interpreter tries to prove the existence of a chandelier, but fails to do so. However, she knows that a chandelier is a kind of lamp and the existence of a lamp *can* be proven using the room just mentioned and the background knowledge that rooms have lamps. Formally, and analogous to the *motor*-example, Γ contains one function which produces an entity for each room; $f:([a:entity, b:room{\cdot}a] \Rightarrow entity)$, and one which states that this entity is a lamp; $g:([a:entity, b:room \cdot a] \Rightarrow (lamp\cdot(f\cdot a\cdot b)))$. Since the speaker has uttered (11) the interpreter will *assume* that (one of) the lamp(s) in the room is a chandelier.[16] In terms of the CTT approach: the interpreter *infers* that the room which John entered contains an entity which is a lamp (applying the aforementioned piece of knowledge; the functions f and g), and then binds *part* of the presupposition by filling the Y gap with $f{\cdot}x{\cdot}y$ (the inferred lamp). The remaining part of the presupposition (that the lamp is in fact a chandelier) is now *accommodated* in the usual way by filling the P gap with a fresh variable.[17,18]

[16] Notice that according to this picture both the anaphor *and* the antecedent play a role in constructing the bridge (see, for instance, Milward 1996).

[17] Where does bridging fit in with Van der Sandt's preference hierarchy? We hypothesize that rule *1*, mentioned in Section 2, should be restated as, *1.a* Binding to a non-inferred antecedent is preferred to accommodation, and *1.b* Binding to a non-inferred antecedent is preferred to binding to an inferred antecedent. Whether binding to an implied antecedent is preferred over accommodation or vice versa cannot be stated in a general way: this again depends on the 'plausibility'.

[18] It has been observed that binding a pronominal anaphor to an implied antecedent is generally impossible. This follows from our present approach: the descriptive content of a pronoun is *so* small, that there will in general be many inferred objects meeting what little descriptive content there is, thus

In: H. Bunt & R. Muskens (eds.), Computing Meaning, Studies in Linguistics Philosophy Series, Kluwer, Dordrecht, 1999.

Summarizing: if the 'bridge' between would-be anaphor and would-be antecedent is fully derivable using world knowledge, the presupposition can be bound. Thus, binding plays a more substantial rôle than in Van der Sandt's original theory, as presuppositions can be bound to both inferred and non-inferred antecedents. On the other hand, if the 'bridge' between anaphor and antecedent is not fully derivable, the 'missing link' will be accommodated. So, accommodation is still a repair-strategy, as in Van der Sandt's original approach, but now there is generally less to repair. In most cases, accommodation will amount to 'assuming' a more specific description of a deduced object (in this case, that the lamp whose existence has been proven is actually a chandelier).[19] Notice, finally, that our approach to bridging is deliberately *not* lexical.[20]

(13) Yesterday somebody parked a car in front of my door, and the dog howled awfully.

This example can be understood in a bridging-manner given the 'right' background knowledge. Suppose, it is well known between

---

resulting in an 'unresolvable ambiguity'. Notice that this approach does not preclude that sometimes a pronoun *can* refer back to a inferred antecedent. Consider: *"Did you hear that John finally is going to get married? She must be very rich"*. In such cases, one implied antecedent ('John's future wife') seems to be more prominent than all others.

[19]In this respect our approach to bridging is comparable to the one advocated in Hobbs (1987) and, in particular, in Hobbs *et al.* (1993). One important difference between our approach and theirs is that we take the *presuppositionhood* of the bridging anaphor as one of the central characteristics. This separation of presupposed and asserted material enables us to resolve bridging anaphors even in cases where the asserted material is inconsistent with the context. A similar point is made in Asher & Lascarides (1996:19), who argue that rhetorical relations are an important factor for processing bridging NPs.

[20]As opposed to e.g., Bos, Buitelaar & Mineur (1995), where bridging is analyzed by the addition of *qualia-structures* to Van der Sandt's presupposition theory. As Bos, Buitelaar & Mineur put it, a qualia-structure can be seen as a set of lexical entailments. Our main objection to this approach is that not all implied antecedents are *lexical* entailments, as example (13) illustrates.

In: H. Bunt & R. Muskens (eds.), Computing Meaning, Studies in Linguistics Philosophy Series, Kluwer, Dordrecht, 1999.

the speaker and the interpreter that the former lives opposite a dog hotel somewhere in the countryside, and all the cars which stop in front of this hotel (and hence in front of the speaker's door) either drop a dog or pick one up. In this context, the hearer will have no trouble constructing the required bridge (since she has a mental function which produces a dog for each car stopping in front of the speaker's door). For more examples, we refer to Krahmer & Piwek (1997).

Conditionals and Presuppositions One attractive feature of the CTT view on discourse is that we get 'discourse markers' for propositions for free. This is useful, for instance, in the case of propositional presuppositions, of which *the fact that S* construction is an example (cf. (1.c)). According to Stalnaker (1974), a proposition which is presupposed should be part of the context (common background). In terms of CTT, this means that a proof for the proposition should be derivable in the context. The latter interpretation agrees nicely with the dictum of presuppositions as anaphors: the proof of the proposition acts as the required antecedent (cf., Ranta 1994).

In order to make this idea more precise, let us give the proto type for example (1.c). For the sake of simplicity we treat *"annoyed by the fact that"* as a (complex) predicate: *annoyed* is a function which applied to a person, a proposition and a proof for the proposition yields a new proposition, $annoyed : [x : entity, q : prop, r : q] \Rightarrow prop$:

(14) $[p : land \cdot sp \cdot plx] \Rightarrow$
$(annoyed \cdot sp \cdot (weigth\_higher \cdot sp) \cdot P)_{[P : weight\_higher \cdot sp]}$

The basic structure of this proto type is $\Phi \Rightarrow \Psi_\pi$.[21] The algorithm sketched in the appendix proceeds as follows. It first tries

[21] This proto type contains some simplifications: the meaning of some parts of the sentence has not been analysed to the fullest detail: we stipu-

In: H. Bunt & R. Muskens (eds.), Computing Meaning, Studies in Linguistics Philosophy Series, Kluwer, Dordrecht, 1999.

to bind the presupposition, in the context of Γ extended with Φ (the conditional's antecedent). In this case, Φ seems to provide no proper antecedent for the presupposition. World knowledge can, however, change the picture dramatically. Suppose that the interpreter knows that: *"if something lands on planet X, then its weight will be higher than it would be on earth"*, formally $f : ([x : entity, q : land \cdot x \cdot plx] \Rightarrow weight\_higher \cdot x)$. In that case, the presupposition can be bound. The appropriate substitution for the presupposition P, namely $f \cdot sp \cdot p$, is obtained by using world knowledge and the information given in the conditional's antecedent.

Now, suppose there is not sufficient information in the context to find a binder for the presupposition. Then some piece of information will have to be accommodated. First, the algorithm attempts to globally accommodate the presupposition. This results in a rather awkward reading, paraphrasable as *"Spaceman Spiff's weight is higher than it would be on earth, and if he lands on planet X, it will bother him (that his weight is higher than it would be on earth)"*. Beaver explains this awkwardness by pointing out that the sentence will typically be uttered in a situation where Spiff is hanging somewhere in space. Most of us know that in space one is weightless. So for the average interpreter, global accommodation of *"Spiff's weight is higher than it would be on earth"* can be blocked: adding this proposition to a context containing the information that Spiff is weightless will enable the interpreter to derive an inconsistency (given some other fairly common pieces of information, e.g., 'on earth one is not weightless').

If global accommodation is ruled out, there are two possibilities left: intermediate and local accommodation. Here, let us consider the reading involving local accommodation (cf. footnote 14). We

---

late that *"Spiff's weight is higher than it would be on earth"* corresponds to $weight\_higher \cdot sp$. Additionally, some presuppositions are already resolved: *"Spiff"* to the variable sp and *"planet X"* to plx.

model Van der Sandt's local accommodation as follows: given a CTT expression of the form $\Phi \Rightarrow \Psi_\pi$ (as (14)), the algorithm adds $\Phi \Rightarrow \pi$ to the *global* context, i.e.: we model local accommodation as global accommodation of a *conditional* presupposition.[22]

5 Conclusions

We rephrased Van der Sandt's *presuppositions as anaphors* theory in terms of CTT, and showed that this facilitates the formal interaction between world knowledge and presupposition projection. To illustrate this interaction, we applied the CTT version of the presuppositions as anaphors approach to Clark's bridging cases and Beaver's conditional presuppositions. These phenomena, which are beyond the scope of theory presented in Van der Sandt (1992), could be dealt with in a straightforward fashion. An important factor in our analyses is the presence of explicit proof objects, which is one of the characteristic properties of CTT.

There are, however, still a lot of open questions. When is bridge a illformed? Why do listeners prefer one bridge over another? And, why should a listener construct a bridge in the first place? In fact, Clark (1975) already provided part of the answers to these questions. For example, he noted that bridging is a *determinate* process, which has to satisfy certain criteria. Among

[22]The advantage of this alternative can be illustrated using another example from Beaver (1995): *"It is unlikely that if Spaceman Spiff lands on planet X, he will be annoyed by the fact that his weight is higher than it would be on earth."*. Van der Sandt's local accommodation produces the following interpretation for this sentence: *"It is unlikely that if Spaceman Spiff lands on planet X, his weight will be higher than it would be on earth and he will be annoyed by this fact"*. Beaver (1995) remarks that Van der Sandt's reading does not entail that *"if Spaceman Spiff lands on planet X, his weight will be higher than it would be on earth"* (it even suggest the opposite), whereas it intuitively should. According to our re-definition of local accommodation the latter sentence does follow from the (adjusted) global context.

In: H. Bunt & R. Muskens (eds.), Computing Meaning, Studies in Linguistics Philosophy Series, Kluwer, Dordrecht, 1999.

other things, Clark proposes a general *stopping rule* which essentially says that listeners build the shortest possible bridge that is consistent with the context. In Krahmer & Piwek (in prep.) it is argued that the CTT perspective can account for this constraint, as well as 'softer' constraints having to do with *relevance* and *plausibility*, in an elegant manner as conditions on proof-objects.

Acknowledgements

Thanks are due to René Ahn and Kees van Deemter for their comments on earlier versions of this paper, and to the audience of IWCS-II (1997), in particular Jerry Hobbs and Nicholas Asher, for comments and questions. Paul Piwek was partially funded by the co-oporation unit of Brabant universities (SOBU). The authors are mentioned in alphabetical order.

A The Resolution Algorithm

Let Φ be the CTT representation of the current utterance, and Γ the current global context. The following algorithm, written in *Pseudo PROLOG*, tells us how to resolve the presuppositions of Φ (if any) in the context of Γ. C is a variable representing the relevant context, consisting of the Γ extended with temporary assumptions (e.g., antecedents of conditionals). Initially, C is set equal to Γ (i.e., $C := \Gamma$). The *basic* clause goes as follows:

$$\mathtt{resolve}(\Phi, C, \Phi) \text{ :- } \mathtt{atomic}(\Phi).$$

If Φ is atomic, i.e., not of the form $\Pi V : \Phi . \Psi$ (also abbreviated as $[V : \Phi] \Rightarrow \Psi$) and not containing presuppositional annotations, then the resolution of Φ in the context of C is Φ. Here is the recursive clause, which deals with Π-expressions ($\otimes$ stands for concatenation).

$$\begin{array}{ll} \mathtt{resolve}(\Pi V : \Phi . \Psi, C, \Pi V : \Phi' . \Psi') \text{ :- } & \mathtt{resolve}(\Phi, C, \Phi'), \\ & C' := C \otimes (\Gamma - C), \\ & \mathtt{resolve}(\Psi, \mathrm{C}' \otimes V : \Phi', \Psi'). \end{array}$$

In: H. Bunt & R. Muskens (eds.), Computing Meaning, Studies in Linguistics Philosophy Series, Kluwer, Dordrecht, 1999.

In words: when the resolution algorithm encounters an expression of the form $\Pi V:\Phi.\Psi$ in context C then it first resolves all the presuppositions in Φ, and when Φ is totally devoid of presuppositional annotations the algorithm resolves the presuppositions of Ψ with respect to the modified context (the original C possibly extended with the accommodation of presuppositions which arose in Φ[23]) and $V : \Phi'$.[24] The first clause to deal with resolution proper is the one for *binding*.[25]

$$\begin{array}{ll}\texttt{resolve}(\Phi_\chi, C, \Phi') \text{ :- } & \texttt{binder}(\chi, C, S), \\ & \texttt{resolve}(\Phi[S], C, \Phi').\end{array}$$

Where `binder` is defined as follows: $\texttt{binder}(\chi, C, S)$:- $S \in \{S^*|C \vdash \chi[S^*]\}, \texttt{preferred}(S)$. When there is more than one possible binding, it is determined which is the most preferred one (where preference is defined in terms of the number of intervening introductions, the complexity of proof-objects, etc.). If there are two equally preferred bindings, an unresolvable ambiguity results. If there is no 'binder' for a presupposition, we try to *globally accommodate* it.

$$\begin{array}{ll}\texttt{resolve}(\Phi_\chi, C, \Phi') \text{ :- } & \texttt{adequate}(\chi[S'], C), \\ & \texttt{add}(\chi[S'], \Gamma), \\ & \texttt{resolve}(\Phi[S'], C \otimes \chi[S'], \Phi').\end{array}$$

Here and elsewhere S' is the assignment which maps any gaps in Φ to Γ-fresh variables of the right type. Thus: if it is possible to accommodate the presupposition, then we may add it to the context Γ, and go on resolving any remaining presuppositions in Φ with respect to the new, extended context. `adequate` checks whether the result of accommodation in a given context meets the Van der Sandtian conditions, i.e., is

[23] $\Gamma - C$ gives those introductions which are present in Γ but not in C, i.e., have been added to the global context Γ since the beginning of resolution.

[24] $V:\Phi'$ is temporarily added ('assumed') to the context in order to resolve any presuppositions in Ψ.

[25] We have decided to code the preferences (binding over accommodation, etc.) into the algorithm itself. This choice is not forced upon us, it is just more efficient than calculating all possible resolutions, and order them afterwards.

In: H. Bunt & R. Muskens (eds.), Computing Meaning, Studies in Linguistics Philosophy Series, Kluwer, Dordrecht, 1999.

consistent and informative.[26] If binding *and* global accommodation are not possible, we try *intermediate accommodation*:

$$\begin{array}{ll}\texttt{resolve}(\Phi_\chi, C, \Phi') \text{ :- } & \texttt{not empty}(C - \Gamma),\\ & \texttt{adequate}((\chi \Rightarrow \Phi)[S'], C),\\ & \texttt{resolve}((\chi \Rightarrow \Phi)[S'], C, \Phi').\end{array}$$

Thus: if we are in an embedded configuration (that is: there is a difference between Γ – the global context – and C – the extension of the global context with a local context –), and the result of intermediate accommodation is adequate, then we use intermediate accommodation. Finally, here is our version for *local accommodation*.[27]

$$\begin{array}{ll}\texttt{resolve}(\Phi_\chi, C, \chi' \otimes \Phi') \text{ :- } & \texttt{empty}(C - \Gamma),\\ & \chi' := \chi[S']\\ & \texttt{resolve}(\Phi[S'], C \otimes \chi', \Phi').\end{array}$$

$$\begin{array}{ll}\texttt{resolve}(\Phi_\chi, C, \Phi') \text{ :- } & \texttt{not empty}(C - \Gamma),\\ & \Delta := C - \Gamma,\\ & \texttt{adequate}(f : (\Delta \Rightarrow \chi)[S'], \Gamma),\\ & \texttt{add}(f : (\Delta \Rightarrow \chi)[S'], \Gamma),\\ & \texttt{resolve}(\Phi[S'], C \otimes f : (\Delta \Rightarrow \chi)[S'], \Phi').\end{array}$$

[26] $V\!:\!T$ is *consistent* in the context of Γ if it is not the case that there is an E such that $\Gamma, V:T \vdash E:\bot$ (that is, adding $V:T$ to Γ makes $\bot$ provable). $V:T$ is *informative* in the context of Γ if it is not the case that there is an E such that $\Gamma \vdash E\!:\!T$ (i.e., T does not follow from Γ already). A *sequence* of introductions is informative if it contains an informative introduction. Notice that adequacy is tested w.r.t. to C while the presupposition is added to Γ. This is done to capture the 'sub-DRSs clause' of Van der Sandt (1992: 367 (iii)). Notice moreover, that Van der Sandt's *trapping-condition* (which states that no variable may end up being free after resolution) is encoded in the CTT framework itself: a variable cannot occur in a context where its type is not declared.

[27] Since χ may consist of a number of introductions $a_1\!:\!b_1, \ldots, a_n\!:\!b_n$ we use an abbreviation here. For instance: $g:([x:entity, p:car\cdot x] \Rightarrow [a_1:entity, a_2:motor\cdot a1])$ is an abbreviation of $g_1 : ([x : entity, p : car \cdot x] \Rightarrow entity)$ and $g_2\!:\!([x\!:\!entity, p\!:\!car\cdot x] \Rightarrow (motor\cdot g_1\cdot x\cdot p))$.

We distinguish two cases: $\Phi_\chi \Rightarrow \Psi$ and $\Phi \Rightarrow \Psi_\chi$. Notice that Van der Sandt's local accommodation of χ in $\Phi \Rightarrow \Psi_\chi$ is modelled as global (!) accommodation of a function $f : \Phi \Rightarrow \chi$ (where f is Γ-fresh).

References

Ahn, R. (1994) Communicating Contexts: a Pragmatic Approach to Information Exchange. In: *Proceedings of the BRA workshop: Types of Proofs and Programs.* Sweden:Baastad.

Ahn, R. & H-P. Kolb (1990) Discourse Representation meets Constructive Mathematics. In: L. Kálmán & L. Pólos (1990) *Papers from the 2nd Symposium on Logic and Language.* Budapest.

Asher, N. & A. Lascarides (1996) Bridging. Austin/Edinburgh: manuscript.

Barendregt, H.P. (1992) Lambda Calculi with Types. In: Abramsky, S., D. Gabbay and T. Maibaum (eds.), *Handbook of Logic in Computer Science.* Oxford: Oxford University Press.

Beaver, D. (1995) *Presupposition and Assertion in Dynamic Semantics.* Ph.D. thesis, Edinburgh.

Beaver, D. (1997) Presupposition. In: J. van Benthem and A. ter Meulen (eds.), *Handbook of Logic and Language.* Amsterdam: Elsevier.

Beun, R.J. & L. Kievit (1995) *Resolving Definite Descriptions in dialogue.* DENK-report 16, Tilburg.

Bos, J., P. Buitelaar & M. Mineur (1995) Bridging as Coercive Accommodation. In: E. Klein, S. Manandhar, W. Nutt and J. Siekmann (eds.), *Edinburgh Conference on Computational Logic & Natural Language Processing.* Edinburgh:HCRC.

Clark, H. (1975) Bridging, In: R. Schank and B. Nash-Webber (eds.) *Theoretical Issues in Natural Language Processing.* MIT.

Curry, H.B. & R. Feys (1958) *Combinatory Logic.* Vol. 1, Amsterdam: North-Holland.

Hobbs, J.R. (1987) *Implicature and Definite Reference*. Stanford: Report No. CSLI-87-99.

Hobbs, J.R., M. Stickel, D. Appelt & P. Martin (1993) Interpretation as Abduction. In: *Artificial Intelligence*, 63:69–142.

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

Krahmer, E. (1995) *Discourse and presupposition.* PhD. thesis, Tilburg.

Krahmer, E. & K. van Deemter (1997) Presuppositions as Anaphors: Towards a Full Understanding of Partial Matches. In: P. Dekker, J. van der Does and H. de Hoop (eds.) *Proceedings of the Definites Workshop*, University of Utrecht (to appear).

Krahmer, E. & P. Piwek (1997) Exploiting Context for Filling Presuppositional Gaps. In: *Proceedings of Context 97*, Rio de Janeiro.

Krahmer, E. & P. Piwek (in prep.) *A Constructive Approach to Bridging*. Eindhoven: IPO.

Krause, P. (1995) Presupposition and Abduction in Type Theory. In: E. Klein, S. Manandhar, W. Nutt and J. Siekmann *Edinburgh Conference on Computational Logic and Natural Language Processing.* Edinburgh: HCRC.

Martin-Löf, P. (1984) *Intuitionistic Type Theory.* Napels: Bibliopolis.

Milward, D. (1996) Integrating Situations into a Theory of Discourse Anaphora. In: P. Dekker and M. Stokhof (1996) *Proceedings of the Tenth Amsterdam Colloquium*, Amsterdam: ILLC. 538–519.

Piwek, P. (1997) The Construction of Answers. In: Benz, A. & G. Jäger (1997) *Proceedings of MunDial '97.* Department of Computational Linguistics, University of Munich (to appear). `http://www.tue.nl/ipo/people/piwek`.

In: H. Bunt & R. Muskens (eds.), Computing Meaning, Studies in Linguistics Philosophy Series, Kluwer, Dordrecht, 1999.

Ranta, A. (1994) *Type-theoretical grammar.* Oxford: Clarendon Press.

Saurer, W. (1993) A Natural Deduction System for Discourse Representation Theory. *Journal of Philosophical Logic*, 22:249 – 302.

Stalnaker, R. (1974) Pragmatic Presuppositions. In: M.K. Munitz & P.K. Unger (1974) *Semantics and Philosophy.* New York: New York University Press, 197 – 213.

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

Zeevat, H. (1992) Presupposition and Accommodation in Update Semantics. *Journal of Semantics*, 9:379 – 412.