Presupposition and Accommodation in Update Semantics

Henk Zeevat
Computational Linguistics
University of Amsterdam

January 23, 1997

Abstract

A reconstruction is presented of VdSandt's theory of presupposition in the framework of update semantics and extended to belief sentences. The resulting view is confronted with earlier approaches to presupposition (especially Heim's) in update semantics, concentrating on the approach to accommodation. It is shown in some detail that the anaphoric view of presupposition can be maintained for only a subclass of presuppositional triggers and must be given up for another class. The paper shows that the treatment of presuppositional anaphora and presuppositional accommodation is compositional with respect to stacks of information states. The brief development of the approach in section 7 shows however that contrary to what one would expect an approach in terms of stacks of information states is a powerful method in the study of DRT and other dynamic systems.

Acknowledgement.

Thanks go to Noor van Leusen, David Beaver, Friederike Moltmann and Remko Scha for spotting mistakes in and suggesting improvements to earlier drafts of this paper. All remaining errors are entirely my own responsibility.

Introduction

Presuppositions in natural language put a classical puzzle to the theorist known as the projection problem. Certain expressions and constructions in natural language can give rise to inferences of the utterances in which they occur regardless of any operator (such as negation, implication, modal operators, attitude verbs) that has scope over their occurrence. Such expressions and constructions are called presupposition triggers. The implicatures to which they give rise are called presuppositions and the content of the presuppositions can be determined from the trigger and its arguments. The problem is that these inferences do not arise in all circumstances. The projection problem is the problem of describing precisely when this happens and when not. Some prime examples of triggers and their (traditional) presuppositions are given in (1).

(1)

Trigger	**Example**	**Presupposition**
definite decriptions	the N	$\exists x\ N(x)$
names	Bill	$\exists x\ x = bill$
cleft	(it was NP, WH S/NP)	$\exists x\ S/NP(x)$
pseudocleft	(WH S/NP, is NP)	$\exists x\ S/NP(x)$
quantifiers	all N	$\exists x\ N(x)$
factives	x regrets that S	S
subordinate clausal PPs	when S, P	S
iterative	S(e_0) again	$\exists e_1\ (S(e_1) \wedge past(e_1))$
lexical	bachelor(x)	$male(x) \wedge adult(x)$

The first modern treatment of the projection problem is due to Karttunen, taking an essentially semantic approach. Counterexamples to Karttunen gave rise to influential approaches (*Gazdar79, Soames82, VdSandt88*) starting from pragmatic intuitions. Both *Heim81* and *VdSandt89* can (but need not) be described as mixed approaches, combining semantic and pragmatic elements. In *VdSandt89* the projection problem is handled by two concepts, anaphora resolution (a semantic process) and presupposition accommodation (a pragmatic notion). In *Heim81*, the relation of logical consequence holding between the local information state and a presupposition is the contribution of semantics and accommodation is again the pragmatic element. Both approaches seem to relate well to the empirical data and also to restore the insights of Frege and Strawson in the study of presupposition, in particular their view that a presupposition is a precondition on the use of its trigger and that its falsity results in an anomaly of the utterance.

VdSandt's later approach to presupposition is couched in the framework of Discourse Representation Theory. (*Kamp81, Kamp & Reyle90*). This is a fortunate choice since the operations the theory employs and the relations that constrain presupposition resolution and accommodation have a direct visualisation and also because the use of the DRS development algorithm is natural for the treatment of presupposition triggers. At the same time it obscures the relationship to the treatments of Karttunen and Heim which were formulated in terms of update operations and opens the question to what degree the theory depends on syntactic manipulation. This paper tries to give a reformulation in terms of update semantics and in particular to answer the question, what the meaning of accommodation is within update semantics. The reformulation will allow a comparison between VdSandt and the update theories of presupposition.

In the following I assume a version of DRT, where DRSs are defined as in (*df.*1). Here ordered pairs of DRSs formalise implications and variable-labeled DRSs belief reports.

Def. 1 *Discourse Representation Structure, SubDRS*

> A DRS A consists of two sets A_{DR} and A_{CON}, where A_{DR} is a finite set of variables and A_{CON} is a finite set of conditions. A condition is either an atomic formula, the sign $\bot$, a pair $\langle B, C\rangle$ of DRSs, or a DRS B_x labeled with a variable x.
>
> If $\langle B, C\rangle$ is a condition of A, then B and C are subDRSs of A.
> If B_x is a condition of A, B is a subDRS of A.
> If C is a subDRS of B and B is a subDRS of A, C is a subDRS of A.

Within this framework, VdSandt's theory can be briefly sketched as in (2), a complex rule in the DRS-development algorithm. This algorithm starts by putting a syntactic analysis tree in a DRS. There is a set of rules allowing a reduction of the tree under the insertion of new discourse referents, conditions and subDRSs. The process stops when there are no longer any (semi-) syntactic objects in the DRS or its subDRSs. We assume that we have some mechanism that produces the undeveloped version (a schematic analysis tree analogous to the analysis trees provided by the syntactic theory for which the development algorithm is defined) of the presupposition triggered by the trigger we have to deal with.

(2) *Presupposition Trigger Development*

> To develop a trigger T with a presupposition P in a subDRS B in a DRS A, we first test whether P can be found in a DRS C on the accessibility path of B in A. If so, the discourse markers in P occurring in B are replaced by the corresponding markers in C. Else we proceed from A to B down the accessibility path and try to add P. This fails, if adding P to one of the DRSs on the path leads to a conflict with the correctness conditions on the assertion at hand or if formal demands are not satisfied. If failure occurs everywhere on the path, the development as a whole fails.

This sketch draws on the definition of an accessibility path, given in (*df.*2), finding a presupposition (df.3) and adding a presupposition (df.4) .

Def. 2 *Accessibility Path*

1. If A is the topmost DRS, $path(A) = \langle A \rangle$.
2. If $\langle C, D \rangle$ is a condition of B then $path(C) = \langle C.path(B) \rangle$ and $path(D) = \langle D.path(C) \rangle$
3. If B_x is a condition of A then $path(B) = \langle B.path(A) \rangle$

We assume that the trigger give a syntactic form to its presupposition, a form which needs development by the other rules of the system. To this end we open a notepad box in which we place the syntactic form of the presupposition and to which we make accessible all the material that is accessible from the site of the trigger. The notepad is discarded if the presupposition is found or when its contents have been added to some part of the DRS under development.

Def. 3 *Finding a presupposition P in A*

A DRS P is found in a DRS A with respect to Z iff A and its subDRSs have discourse markers $x_1 \dots x_n$ that stand in 1-1 correspondence with the discourse markers of P and its subDRSs $y_1 \dots y_n$ such that P's (simple or complex) conditions are conditions of A under the substitution of $y_1 \dots y_n$ for $x_1 \dots x_n$.

(df.3) imposes a purely syntactic relationship between the presupposition and the DRS in which it is found. The definition can be made more semantical by recursively defining a subsumption relation, so that complex conditions need not have literal counterparts under the substitution, but have counterparts which subsume them. For our purposes however, the current definition is good enough, as the problem disappears completely when we switch from DRSs to information states.

Def. 4 *Adding P to A*

Adding P to a subDRS A of B consists of adding each of P's discourse markers to the markers of A and of adding each of the conditions of P to the conditions of A. Adding is undefined if a condition of P contains a marker that does not have a accessible discourse referent in P or A or in a DRS accessible from A.

The process of looking up a presupposition is analogous to answering a question in Prolog: it instantiates variables. As a simple example, consider (3).

(3) John saw a donkey. The donkey was ill.

A donkey sets up a discourse marker x. In processing *the donkey*, we try to find a DRS consisting of a discourse marker y and the condition $donkey(y)$. This succeeds as we can substitute x for y and find both the condition and the marker in the DRS resulting from processing the first sentence. Substituting the y in *y was ill* completes the development of the trigger.

If we consider a single discourse marker to be a proposition as well, anaphoric binding can be reduced to presupposition. The development of anaphoric pronouns can be defined by inserting a discourse marker in the matrix for the pronoun and by triggering the presupposition that this marker is a discourse marker. (Extra restrictions need to be imposed here though.) This development rule would transform the expression *he sleeps* into the condition *sleep(x)* where the proposition x would have to be resolved. The resolution to a marker y would result in a substitution transforming *sleep(x)* into *sleep(y)*. Notice that the identification of presupposed material is a purely syntactic notion. We will come back to this point later on.

The last notion that is crucial to VdSandt's theory is the notion of a correct assertion. An assertion is correct (VdSandt refers to *VdSandt88*) if it meets the conditions in (*df*.5).

Def. 5 *Correctness*

An assertion is correct iff

1. It does not follow from the DRS it is developed in.

2. It is not in contradiction with the DRS it is developed in.

3. If a DRS A contains a condition $\langle P, Q \rangle$, (1) and (2) also apply to P w.r.t. A and to Q w.r.t. to the DRS obtained by merging A and P.

The correctness conditions derive from Stalnaker's assertion conditions (df.5.1 and .2) and are extended to some subsentential cases in (df.5.3) . The conditions should be compared with Gazdar's definition of clausal implicatures, responsible in his system for the cancellation of presuppositions. (4) spells out some immediate consequences of (df.5) in the form of a list of incorrect texts. The correctness conditions are weaker than clausal implicatures, since they require for saying some assertion A that the common ground does not yet contain A or $\neg A$, not that the speaker does not know these. Full clausal implicatures still require the Gricean maxims.

(4)

It rains. It rains.
It does not rain. It rains.

It rains. It does not rain.
It does not rain. It does not rain.
It rains. If it rains, John is bringing his umbrella.
It does not rain. If it rains, John is bringing his umbrella.
It rains. If it is Monday, it rains.
It rains. If it is Monday, it does not rain.
It does not rain. If it is Monday, it rains.
It does not rain. If it is Monday, it does not rain.

For presupposition projection, the crucial effect of correctness is the absence of projection in case the addition of the presupposition before the assertion would make the whole assertion incorrect. In VdSandt, the other explanation of the absence of projection is the case where the content of the presupposition is found in a proper subDRS accessible from the position of the trigger. Gazdar handles these cases by means of clausal implicatures as well.

1 Basic Update Semantics

If pursued in a principled way, update semantics characterises the meaning of expressions by stating the contribution an expression makes to information states, generally defined as sets of possibilities. As a first example consider propositional logic. Here the set of possibilities can be given as the set of truth value assignments to the propositional variables of a given language L. This forms a set of possibilities I. Information states σ are subsets of I. Two special information states are I itself, the empty information state and $\emptyset$, the inconsistent information state. (1 and 0 are used to refer to the empty and inconsistent information states independently of their definition in a system of information states.)

The standard meaning of the propositional connectives can now be developed by stating what change formulas make to information states. $\sigma[\varphi]$ denotes the information state obtained by adding the information in φ to the information state σ. A recursive definition for all connectives is given in (*df.*6). Below p is used for atomic formulas and φ for arbitrary complex formulas.

Def. 6 *Propositional Updates*

$$\sigma[p] = \{i \in \sigma : i \models p\}$$
$$\sigma[\neg\varphi] = \sigma - \sigma[\varphi]$$
$$\sigma[\varphi \wedge \psi] = \sigma[\varphi][\psi]$$
$$\sigma[\varphi \rightarrow \psi] = \sigma[\neg(\varphi \wedge \neg\psi)]$$

First Order with Discourse Referents

In order to reach first order logic we must generalise slightly. We will consider not first order logic but a formalism similar to DRT, where variables are treated as atomic formulas and formulas (DRSs) are built up using the connectives:

$\neg$

$\wedge$

$\rightarrow$

As an example, (5) gives a formulation of the donkey sentence.

(5) $(x \wedge farmer(x) \wedge y \wedge donkey(y) \wedge own(x,y)) \rightarrow beat(x,y)$

For the interpretation, we start with possibilities $i \in I$ that are functions assigning appropriate values to predicate constants, variables and individual constants. Variables and individual constants are mapped to the elements of some non-empty set U, n-place predicate constants to subsets of U^n . We do not require that every information index is defined for all the variables or for constants. Undefinedness for constants will play no role in the sequel however.

1 is again I and 0 the empty set. Independently of the updating process, the discourse referents are given by a recursive definition (*df.*7).

Def. 7 *Discourse Referents*

1. $DM(\varphi) = \emptyset$ iff φ is atomic (but not a variable) or a negation or an implication.
2. $DM(x) = \{x\}$
3. $DM(\varphi \wedge \psi) = DR(\varphi) \cup DR(\psi)$

Discourse referents are used in the auxiliary notions in (*df.*8). These define information indices i and j to be variants with respect to a given set of variables and the closure of an information state under taking variants with respect to a set of variables. The last notion will play a role in defining negation.

Def. 8 *Variants and Closure*

$i =_{\{x_1 \ldots x_n\}} j$ iff $i(a) = j(a)$ for every $a \notin \{x_1 \ldots x_n\}$

$\sigma^{\{x_1 \ldots x_n\}} = \{j : \exists i \in \sigma \; i =_{\{x_1 \ldots x_n\}} j\}$

The proper updating notion is given in (*df.*9).

Def. 9 *First Order Updates*

$\sigma[Pt_1 \ldots t_n] = \{i \in \sigma : \langle ix_1 \ldots ix_n \rangle \in iP\}$

$\sigma[x] = \{i \in \sigma : ix$ is defined $\}$

$\sigma[\neg\varphi] = \sigma - \sigma[\varphi]^{DM(\varphi)}$

$\sigma[\varphi \wedge \psi] = \sigma[\varphi][\psi]$

$\sigma[\varphi \rightarrow \psi] = \sigma[\neg(\varphi \wedge \neg\psi)]$

Notice that the negation takes care of all quantification.

The final step that we have to cover is the addition of belief operators.

Here we run into a problem. What we want to have is a set I as above, with the extra proviso that $iu \subseteq I$ if $u \in U$. Intuitively iu is the proposition that expresses u's belief state in i. The problem is that in set theory we cannot have the empty information state with these requirements. Here iu would need variants j with respect to u where ju would be any subset of I. Such a set would be too large. Using the canonical model construction we can however guarantee (with respect to a language) that there is an empty information state that has "enough" variants.

A definition due to Kamp and used by Heim in *Heim92* for belief updating is given in (*df.*10).

Def. 10 *Pointwise Belief Update*

$$\sigma[Bx\varphi] = \{i \in \sigma : iix[\varphi] = iix\}$$

Since for presupposition resolution and accommodation belief updates have to be defined over whole information states —which rules out a pointwise definition of belief updates— I have to complicate this definition.

The first step is that we compute the belief-state of the subject from the set of ixs where i is given as a member of σ. The union of these sets of possible worlds can be taken as x's belief according to σ. It is the updated union that forms the criterion for elimination in σ: those worlds whose beliefs for x are not subsets of the updated beliefs are eliminated from σ.

To keep the definition clean, the auxiliary notion (*df.*11) is introduced. This replaces the information that σ has with respect to x's belief with an information state J, by the normal process of eliminating information indices.

Def. 11 *Limitation of Beliefs*

$$\sigma_x^J = \{i \in \sigma : iix \subseteq J\}$$

(*df.*12) defines the information state σ attributes to a subject x.

Def. 12 *Determining Belief States*

$$\sigma^x = \bigcup_{i \in \sigma} iix$$

The update can now be written as (*df.*13), combining (df.11) and (df.12) .

Def. 13 *Global Belief Update*

$$\sigma[Bx\varphi] = \sigma_x^{\sigma^x[\varphi]}$$

2 Presuppositional Anaphora Resolution

There is only one possibility for defining anaphora in update semantics: the notion of local satisfaction. If T is a trigger with a presupposition P which will be added to an information state σ, $\sigma[P] = \sigma^g$ must hold, for some g mapping discourse referents of P to variables[1] This is fairly close to VdSandt, although there are differences.

The way we set up updating makes all the material on the accessibility path of the DRS part of the information state of the trigger, unless we are in a belief context. For good order, we will ignore beliefs for the time being. In this way it follows that if in the corresponding DRS a condition would be on the accessibility path, now the information in the condition will be information in the information state of the trigger, as it has been put there by earlier updates. It holds therefore that if the presupposition P can be found in the DRS, it will be provable from the corresponding information state.

The converse does not hold. One possibility is that the presupposition has been entered divided over a number of DRSs on the accessibility path. Another possibility is that the presupposition is inferable from the information state, but not explicitly coded in the corresponding DRS. It is against the latter possibility that VdSandt directs his counterexample (7) [2].

(7) If John has grandchildren, his children must be adult.

According to VdSandt, this sentence has interpretations where it is inferred that John has children and others where this is not so. (One must assume that it is unknown in the context of utterance whether John has children and one must assume as well that the intonation pattern on *his children* is not such that his children is assigned to the sentence topic. It must also be ruled out that having grandchildren presupposes having children, rather than implying this.) The updating perspective would however predict that from the conceptual fact that in order to have grandchildren one must necessarily have children, it would

[1] To maintain determinism, we will require throughout that such functions are unique. As VdSandt points out, it is more realistic to switch to a non deterministic resolution scheme, where more solutions are allowed. Such a scheme can be easily defined as in (*df.*14), but a the price of losing the clarity of a deterministic update notion.

Def. 14 *Non Deterministic Presupposition Resolution*

$$\exists g(\sigma^g[P] = \sigma^g \Rightarrow \sigma(P) \in \{\sigma^g : \exists g \sigma^g[P] = \sigma^g\}$$

[2] David Beaver (p.c.) rightly objects to this example that *his children* forces the resolver to have made a choice that John would have more than one child, something that does not follow from his having grandchildren. He reports coming up in collaboration with Kamp with (6) where this problem does not seem to arise.

(6) If Pete and John have grandchildren, their children must be adult.

This modification gets rid of the uneasiness that one feels with original example but seems to retain the same two readings.

always be true in the local information state of the trigger *his children* that John has children, so that the projection of John's having children does not occur.

The argument can be turned around by noting that in these circumstances the strict matching required by VdSandt[3] would predict that there is only one reading, namely the one where John's having children is globally accommodated. If the facts are as VdSandt states, we would want a theory where partial matching is properly interpreted: i.e. as a process whereby an antecedent that entails a matching antecedent is (optionally) adapted to include this matching antecedent. Section 8 provides a sketchy development of this alternative.

It depends on the form of the presupposition whether the presupposition is automatically true in the update semantics I sketched. If we enter the fact about having children and having grandchildren as a meaning postulate, it will be a condition on the informational indices: $grandchild(u, v)$ will never be true unless there is an object w such that $child(u, w)$ and $child(w, v)$. So if we make the presupposition equivalent to $\exists x child(x, john)$ (taken as a formula in first order logic) it will be satisfied. But if the presupposition is $child(x, john)$, with x a new variable, or for that matter $x \wedge child(x, john)$, it will not be the case that the presupposition is already entailed. If x is new it will have all kind of values, if we allow variable substitution for entailment there will not be another variable y of which it is already known that y is a child of John.

If we leave these options open, it is easy to envisage other bars on an updating theory for full anaphora. Suppose we follow the proposal of (*Asher*MS) of introducing fact discourse markers whenever we find a full-blown fact expressed in language. (Asher proposes this for anaphora to sentences and texts.) Then simple anaphora to facts that are scattered over different updates would be prevented. It is therefore by no means clear that we would have to admit that update semantics makes the wrong predictions by relying on logical consequence. What the example shows, is that there may be a psychology of presupposition resolution: a notion of proving where the decision whether it is really provable or not may go both ways in certain circumstances. We cannot prove the existence of the children from the information state but the changes needed to the information state are minor. There may be a threshold where the changes become insignificant.

Two final differences involve propositional attitudes. On the updating approach, any old belief of somebody will be a potential antecedent for a presupposition connected to a trigger within a new belief report about that person. These do not lie on the accessibility path of the belief report. As other beliefs are good antecedents, it seems that update semantics is just the better theory in this respect. Compare (8).

[3] In later work (this volume) VdSandt abandons determinism and so obtains a (non-preferred) reading where the children are accommodated at the site of the grandchildren. Strict matching still prevents a proper anaphoric reading where a conceptual link between the children and the children is established

(8) John believes Bill$_i$ is married. He also believes his$_i$ wife is happy.

Secondly (as I noted before), the subDRS for a belief report does not have the property that a fact to which it has access holds within the subDRS: it can be explicitly denied there. In this way, it is possible to have a presupposition triggered within the belief report that would find an antecedent outside but not within the report. (9)can be understood as referring to the butcher we all know, but of whom John does not know he is the butcher.

(9) John believes the butcher sings.

Update semantics predicts that this can never happen, since in such cases the presupposition is not fulfilled in the context of the trigger. In fact, we will see counterexamples against the update semantics claim as well as counterexamples against the claim of VdSandt in this respect.

3 Van der Sandt's Accommodation

Accommodation in VdSandt is defined by a recursion on the accessibility path, defined as a stack of DRSs in (**df. 2**). We try to add the presupposition to the first element, then proceed and check the correctness conditions at the end of the whole updating process. If we fail, we try again on the tail of the path and so on. In update semantics we do not have an accessibility path and so we must find an analogous structure. What we will use instead is the stack of information states under determination for the current update. Recall that a negation $\neg A$ forces us to do an auxiliary update with A and an implication $A \rightarrow B$ auxiliary updates with A and with A and B. A belief $B_x A$ similarly involves the auxiliary update $\sigma_x[A]$. Formally we will switch to an update notion on a stack of information states. I will write these stacks as Σ where $\langle \sigma.\Sigma \rangle$ stands for the result of pushing σ onto Σ, Σ_0 for $pop(\Sigma)$ and Σ_1 for $tail(\Sigma)$.

Def. 15 *Stack Updating*

1. $\Sigma[Pt_1, \ldots, t_n] = \langle \{i \in \Sigma_0 : \langle ix_1, \ldots, ix_n \rangle \in iP\}.\Sigma_1 \rangle$

2. $\Sigma[x] = \langle \{i \in \Sigma_0 : ix \text{ is defined}\}.\Sigma_1 \rangle$

3. $\Sigma[\neg\varphi] = \langle T_{10} - T_0^{DM(\varphi)}.T_{11} \rangle$ where $T = \langle \Sigma_0.\Sigma \rangle[\varphi]$

4. $\Sigma[\varphi \wedge \psi] = \Sigma[\varphi][\psi]$

5. $\Sigma[\varphi \rightarrow \psi] = \Sigma[\neg(\varphi \wedge \neg\psi]$

6. $\Sigma[B_x\varphi] = \langle(\Sigma_0)_x^{T_0}.T_1\rangle$ where $T = \langle\bigcup_{i\in\Sigma_0} iix.\Sigma\rangle[\varphi]$

None of the definitions we employed so far has any relation with presupposition. To enter presupposition we introduce the possibility that certain atomic formulas (and some operations) have a presupposition. This can be done in various ways. I will just assume that an atomic formula *bachelor(x)* puts a limitation on simple updating: updating is undefined in case x is not already an adult male. In stack updating this will change: $\Sigma[bachelor(x)] = \langle\Sigma_0[unmarried(x)].\Sigma_1\rangle$ in case x is an adult male, else the result of first accommodating $adult(x) \wedge male(x)$ over Σ to obtain $\Sigma 1$ and then determining $\Sigma 1[bachelor(x)]$.

We pack the presupposition and accommodation together by having a partial operation *presupposition* (written as "round brackets update") on stacks that in case the presupposition is satisfied, delivers Σ as output and else the accommodation of Σ. In (*df.*16) the contributions of some lexical items are defined, as exceptions to the general proviso in (df.15.1) .

Def. 16

$$\Sigma[bachelor(x)] = \Sigma(adult(x))(male(x))[unmarried(x)]$$
$$\Sigma[know(x,p)] = \Sigma(p)[know(x,p)]$$

Both $\Sigma(.)$ and $\Sigma[.]$ are partial operations: their success depends on the contents of Σ. The undefinedness of accommodation is illustrated by (10),

(10) The king of France is bald.

uttered in a situation where it is known that there is no king of France. This means that the presupposition (11)

(11) There is a king of France

or its DRT-form:

(12) $x \wedge KoF(x)$

cannot be found in the information state. Also, accommodation fails, since adding the presupposition would make the information state inconsistent. So the update is not possible with respect to this state. It is also not possible to update (13) in this information state, since (13) does not give new information.

(13) The king of France is not bald.

Accommodation in the auxiliary information state for the negation leads to an attempted update of (13) which again leads to inconsistency. (*df.*17) gives a definition of accommodation as an operation changing the information stack. The operation depends on the the definition of correctness in the next section. In addition we need the idea of a "unification" update over a stack. If h is a substitution mapping a finite set of variables $\{x_1, \ldots, x_n\}$ to variables $\{y_1, \ldots, y_n\}$, then $\langle \Sigma^h = \Sigma_0[x_1 = y_1 \wedge \ldots x_n = y_n].\Sigma_1\rangle$.

Def. 17 *Presupposition (VdSandt)*

(1) $\Sigma(\varphi) =$	Σ	if there is a substitution h for the discourse markers of φ such that $\Sigma_0[\varphi]^h = \Sigma_0$
(2) else	T	if T= $\langle(\Sigma_0 \cap \Sigma_1(\varphi)_0).\Sigma_1(\varphi)_1\rangle$ is defined and is a correct stack.
(3) else	$\Sigma[\varphi]$	if $\Sigma[\varphi]$ is defined and a correct stack

undefined otherwise

In (df.17.1) we have a version of our earlier notion of finding a presupposition. We exploit here that correctness is not defined for information states (such as Σ_0), so that lack of information will not block the process. In (2) we look further down the stack in case we have not found the presupposition. This process adds information to the first information state on the stack (by unification or accommodation) which is copied into Σ_0 by set intersection. Clause (3) accommodates the material, if it cannot be found here or found or accommodated further down the stack. (df.17) can be illustrated by the examples in (14) interpreted as updates to the empty information stack $\langle 1\rangle$.

(14)
1. There is a king. The king sings.
2. The king does not sing.

If the trigger *the king* is reached in (14.1) , the stack is $\langle 1[x][king(x)]\rangle$ so the information $y \wedge king(y)$ is available in the first information state under the substitution (x,y). So we obtain via $\langle 1[x][king(x)][x = y]\rangle$ the information stack $\langle 1[x][king(x)][x = y][sing(y)]\rangle$. In (14.2) the information stack is $\langle 1, 1\rangle$ by the time we have to process the trigger. Clause (2) now applies since clause (3) makes $\langle 1\rangle(y \wedge king(y)) = \langle 1\rangle[y \wedge king(y)] = \langle 1[y \wedge king(y)]\rangle$. So we obtain via clause (2) the information stack $\langle 1[y \wedge king(y)], 1[y \wedge king(y)]\rangle$ and finally subtract $1[y \wedge king(y)][sing(y)]$ from $1[y \wedge king(y)]$ to obtain the only element of our final information stack.

(df.17) will not be applicable to belief contexts. The addition of new material to a higher clause, enforced by (2) is inherited down the stack by intersecting the current information state by its changed successor. This is necessary since the earlier information states are unchanged by an accommodation further down the stack. The intersection adds the accommodated information to the earlier information states, as long as the stack is correct. For attitudinal information states this will not do as the information in these bears no relation to the information in the information state it derives from. That is why the notion derived from Heim (*df.*18) is more general. In the next section we will see why this is a reasonable reconstruction of Heim81.

Def. 18 *Presupposition (Heim)*

(1) $\Sigma(\varphi) =$	Σ	if there is a substitution h for the discourse markers of φ such that $\Sigma_0[\varphi]^h = \Sigma_0$
(2) else	T	if T= $\langle\Sigma_0[\varphi].\Sigma_1(\varphi)\rangle$ is defined and is a correct stack.
(3) else	$\Sigma[\varphi]$	if $\Sigma[\varphi]$ is defined and is a correct stack

undefined otherwise

The process which brings information to the earlier information states is now a separate update rather than intersection. An important difference is local accommodation must take place on the whole path between the trigger and its antecedent. Global accommodation in the update of $\sigma$with (15) (clause (3) adds a king to σ) can now be handled by clause (2), which adds the king to John's belief state.

(15) John believes the king sings

Under (df.17) we would have to intersect John's belief state with σ which would add the whole information state to John's beliefs and would be impossible if John is known to have a false belief.

We can however use the VdSandt notion as defined in (df.17) if we make a special proviso for belief stacks (stacks whose first information state is an introduced by a belief operator). For those we add the presupposition to the first information state and presuppose it over the rest of the stack.

Def. 19 *Presupposition (VdSandt) for belief*

(2') $\Sigma(\varphi)$ =T if T= $\langle\Sigma_0[\varphi].\Sigma_1(\varphi)_1\rangle$, Σ is a belief stack and T is defined and correct.

Without the proviso, belief sentences are a real problem for VdSandt-style accommodation. The first clause of (df.18) will succeed for (16) since the presupposition that Mary left can be found in the basic context. This fails to predict that in (16) John cannot believe this unless he believes himself that Mary left.

(16) Mary left and John believes that Bill regrets that Mary left.

On the other hand, there are also problems for the Heim style accommodation since it predicts that (17) is impossible (on the assumption that there being a king and a president at the same time is inconsistent).

(17) There is a king and John believes that the king is the president.

Adding the presupposition within John's beliefs would make those inconsistent. We will come back to these problems at a later point.

Unfortunately, what we said so far about presupposition is not the whole story. We have seen that faulty accommodation can result in the failure of later assertions. So accommodation on the trigger *regret*, will prevent the consistent update of the content of the *since*-clause.

(18) John does not regret killing Mary, since he never did kill her.

For (18) , we have to accommodate

(19) John killed Mary.

with respect to the stack $\langle\sigma, \sigma\rangle$, which would give us (20).

(20) $\langle\sigma[\text{John killed Mary}], \sigma[\text{John killed Mary}]\rangle$

The *since*-clause would have to be applied to the stack $\langle\sigma[\text{John killed Mary}][\ldots]\rangle$ which would lead to failure since the update would give us the empty information stack. So, we should perhaps have presupposed (19) with respect to (21).

(21) $\langle\sigma[\text{John did not kill Mary}], \sigma[\text{John did not kill Mary}]\rangle$

But here we cannot accommodate *John killed Mary*, since this would lead to an inconsistent information state under the negation. So it seems that the inconsistency resulting from global accommodation combined with the update in the *since*-clause does not lead to reordering but to a different accommodation, where clause (3) is chosen instead of clause (2) because of a later failure in the update.

A natural model of the situation is backtracking. $\sigma(.)$ would not be an operation but a relation with various ordered solutions. We take the blocking resulting from the attempted update with the content of the *since*-clause, to fail the success of clause (2) in our definition and to lead to the next solution. We moreover assume a default preference for the first solution which we find and which persists throughout the update. This model is not so bad as we can make the accommodation operation depend on the success of the total update.

For this, accommodation must be defined in a simultaneous recursion with updating. The resulting definition runs almost directly in Prolog.

Other solutions using indeterminism or freezing (postponing the evaluation of some relation until all the data for the evaluation are available) run into the probem I just mentioned: for the local accommodation it is essential not to have added the information in the *since*-clause, since consistent local accommodation is otherwise ruled out.

A declarative treatment of this mechanism can however be envisaged taking the lead from the work of Mercer (*Mercer92*). Accommodation is adding the presupposition to each of the relevant information states on the stack as a default statement and not as a fact. When the stack is shortened (the information state is closed off because an auxiliary update is over), we use not the information state we have obtained but the closure of the information state where the default information is turned into factual knowledge whenever there is no conflict with the facts. A similar closure operation must be applied to the single element stack after a successfully completed update[4]. A formalisation of this option will have to wait for another occasion. For the time being (in line with most of the literature) we will pretend that the problem does not arise.

4 Correctness

The stack of contexts as it changes under the influence of successive accommodations can also be the basis for testing correctness. VdSandt defines correctness on the basis of Stalnaker's assertion conditions (*Stalnaker78*) as the requirement that assertions make a consistent and informative contribution to the context. Since we assume that a stack always corresponds to a single assertion, we can catch the requirement as another demand on updating. We will go here for the strong position that every update makes a contribution: it is not allowed that the context becomes inconsistent and it is required that some worlds are eliminated. If any update does not meet these requirements, the update is undefined.

It can be argued that exceptions to this principle are necessary. Take example (22).

(22) The king of France does not exist. So the king of France is not bald.

The second sentence involves the auxiliary update on the context partially produced by the first sentence that the king of France is bald. This is inconsistent with the earlier information that that king does not exist. So updating is blocked. We can allow for this example by being more liberal for contexts introduced by a negation: updates may produce inconsistencies if they happen in

[4]This notion can be defined properly only in a discourse grammar, as cancellation can be caused by the next sentence or even after an interruption or elaboration.

an auxiliary context produced by a negation operator. Maybe we should even change our clause for negation in such a way that the auxiliary context need not be the context in which the negation is introduced but a revision of that context. So much information has to be taken out that it becomes possible to perform the auxiliary update without contradiction arising. A mechanism of this kind is required for counterfactuals anyway.

Another option is, however, to regard the *so*-clause as analogous to a presupposition: a statement that is not meant to introduce new information and marked as such by discourse particles like *so* or *indeed.* Without attempting a treatment of these pseudo-assertions, I will stick to this option for the time being and desist from changing the rules for negation.

We can state correctness as conditions on the stack and as conditions on updates. No update is allowed to produce the empty set as the first member of the stack or to leave the stack unchanged. The stack itself must consist of a sequence of increasingly informative consistent information states,

Def. 20

$$\Sigma = \langle \sigma_1, \sigma_2, \ldots, \sigma_n \rangle \text{ with } \sigma_1 \subset \sigma_2 \subset \ldots \subset \sigma_n$$

allowing only information states generated by belief reports to be unrelated to their successors.

Belief contexts thus supply their own criteria of informativeness and consistency: new beliefs of John must be consistent with what we knew about John's beliefs and must provide new information with respect to what we knew about John's beliefs.

Given the way things are set up, it will be sufficient if we know within the stack when a context is a belief context, information which we already needed independently for characterising accommodations. Apart from belief contexts, correctness is the requirement that the stack keeps consisting of non-empty monotonically decreasing contexts. Whenever we enter a belief, a reinitialisation takes place after which the same requirement will hold for the substack until the reinitialisation.

5 Heim's Accommodation

The position we have arrived at is rather close to (*Heim81*), with accommodation being different. Heim's position on accommodation is not fully explicit, but two differences with the VdSandt position can be given. First, accommodation processes operating on operators like negation come in a *global* and a *local* version. Local accommodation is all that is required from a logical point of view for the presupposition trigger to become felicitous. This leads to the question when global accommodation occurs, and the natural assumption is that it either is a possibility next to local accommodation or that it is the default accommodation. If neither of these is the case, it would follow that global accommodation is a useless operation. Against the first possible interpretation, it can be argued that —in general— the need for accommodation does not seem to lead to a perceived ambiguity, if the context is fixed. So it seems global accommodation should be a default. (Local accommodation as a default is not an option since —again— global accommodation would never be chosen.) Second, global accommodation is different from the process we have studied until now, since it involves adding the presuppositional material to all intermediate information states between the trigger and the global context.

Next to the unclarity with respect to global accommodation, there are two other points where different interpretations seem possible. It is not made explicit what can block global accommodation and with respect to which operator the global operation takes place. As to the blocking we can follow no doubt the position that we take whatever seems appropriate and the correctness conditions derived from *VdSandt88* are a natural choice. As to the second point, we have a problem. Suppose we have a trigger T in a sentence S of the form

(23) $\neg T \rightarrow A$

Suppose that the presupposition of T is not satisfied. The blocking of updating occurs while updating both the negation and the implication: we are doing an auxiliary update within an auxiliary update. We can now globally accommodate with respect to either the implication and the negation. Which one should we choose?

The best choice seems to be the outermost operator, as this leads us towards a natural treatment of the counterexamples against Karttunen, bringing about projection to the global context. Globally accommodating with respect to the outermost operator in general entails the effect of the accommodations performed with respect to the more embedded operators under which the trigger occurs.

So it seems we are back in the situation of the last section. Given the series of operators $O_1, \ldots, O_k$ having scope over the trigger and requiring auxiliary updates, we generally seem to prefer accommodation with respect to the outermost operator O_1, as this entails global accommodations with respect to each of the other operations and as it deals with projection to the global context. If we take the position that with an implication we have three accommodation options, we can describe the accommodation options by referring to our stack of auxiliary updates. This option with respect to implication seems quite reasonable. Recall that $\sigma[\varphi \rightarrow \psi]$ was defined as $\sigma[\neg(\varphi \wedge \neg\psi)] = \sigma - (\sigma[\varphi] - \sigma[\varphi][\psi])$. It seems possible to accommodate a trigger in ψ by adding the presupposition to $\sigma[\varphi][\psi]$ or to both $\sigma[\varphi][\psi]$ and $\sigma[\varphi]$ or to each of $\sigma[\varphi][\psi]$, $\sigma[\varphi]$ and σ. Similarly it is possible to add a presupposition deriving from a trigger in φ in σ or in $\sigma[\varphi]$.

So the accommodation can again be understood as an operation on the stack of contexts. If we moreover stipulate that where possible we use global accommodation on the highest operation for which this works, we are extremely close to VdSandt. The one remaining difference is the question whether we should add the presupposition everywhere between the position of the trigger and the highest position where it can be accommodated or whether we can be satisfied with adding it just once at that position.

There is no difference between the two positions when the auxiliary information states are generated for the benefit of updates of truth-functional operators or quantifiers. In this case it holds that adding the presupposition to a higher state makes it come out as belonging to each of the states further down on the stack. But it makes a difference in case the operator is a propositional attitude or a modal operator. In the next section we will see that both solutions here lead to problems and that we need a combination of the two views.

6 Lexical Presupposition and Anaphora

In the last section we saw that there is a difference between the two notions of accommodation proposed by VdSandt and Heim. Heim demands that the presupposition is satisfied between the trigger and the location of the antecedent on the accessibility path or between the trigger and the accommodation site. VdSandt is already content if the trigger has access to the antecedent or to the accommodation site. On a fragment with just standard logical operators, there is no reason for choosing between the two positions. If a trigger has access, it holds that the presupposition holds in the context of the trigger and in all the intervening contexts, as can be shown by a simple induction. Belief contexts however are different. If the trigger sits in a belief context and the antecedent is outside, it does not follow that the presupposition holds in the belief state.

It is my belief that there are presupposition triggers for which VdSandt is to be preferred and other cases for which Heim is to be preferred and that this has to do with two different types of presupposition.

To avoid confusion let me start by stating that I agree with VdSandt that all presupposition shares important characteristics with anaphora. Like anaphoric pronouns, the presence of the antecedent does not just license the use of the pronoun/trigger: it makes it obligatory to choose the pronoun or the trigger over an alternative. It is wrong[5] to say (24)

(24) John came in. John took of his hat.

instead of (25),

[5] The notion of incorrectness involved here is usually identified with the incorrectness notion of discourse grammar. The observation that this incorrectness is less dramatic than the one arising from sentence grammar has been often made and is usually explained by our superior ability to correct incorrect discourses. Nevertheless, discourse mistakes are easily recognised and form a reliable starting point for the study of discourse.

(25) John came in. He took of his hat.

since the pronoun is fully licensed when John occurs for the second time. It is similarly problematic to say

(26)

It rains. John believes that it rains.

as we could have used *knows* or *accepts* instead.

Like anaphora, presupposition triggers set up relations between different parts of a text. But given this anaphoric character of presupposition triggers there is still a group of triggers that is even more anaphoric in the sense that their primary function is —like anaphora— to collect entities from the environment in order to say new things about them. Prime examples are definite descriptions. But it is reasonable to include factive *when-* and *after*-clauses and perhaps clefts as well. I will refer to these as resolution triggers.

Of the remaining triggers an important group are concepts with the applicability conditions. In these cases, the application of a concept is only an option if certain conditions are already met. The conditions that must be met are the lexical presuppositions of the concept. Their function is therefore very different, even though they may refer to pre-established knowledge and often do so.

A prime example seems to be sortal information associated with verbs and nouns. The meaning of these words can typically be divided into a part that which identifies the type of entity referred to and a part which actually describes the entity. Another prime example is constituted by the preconditions of actions and states. I will refer to the whole class as lexical triggers.

Stated in this way, it seems to follow that resolution triggers follow VdSandt. They are pointers to the referents of their antecedents and deliver their value to the current context to help build a new thought. Consequently, it is not necessary (but often implicated) that the properties they attribute to their referents are part of the thought to which they contribute their referent, as they can play their role without help from the belief subject. (It is sufficient that the adressee of the utterance is able to figure out what the referent is.) The existence of the referent at the position of the trigger within the context is however necessary. It seems immaterial for our purposes whether the existence presupposition is attributed —as part of the characterisation of its meaning— to the trigger or to the matrix[6] in which the trigger occurs. It is necessary however that this presupposition is lexical rather than a resolution presupposition. It is also not possible to associate the customary existence presuppositions with e.g. definite descriptions since those would not be distinguishable from the resolution presupposition. It would be my proposal to use just the discourse

[6]See Seuren88 for a defence of the view that it is the matrix that is responsible for the existence presupposition

referent and to think of the presupposition generated by e.g. *the book* as just x (with $book(x)$ the resolution presupposition) or the one generated by *when Bill left* as just e (with $leave(e, b)$ the resolution presupposition).

In the following, an example is presented.

(27) John believes that the king is bald

The example provides (at least) two presuppositions.

Lexical presupposition	x
Resolution presupposition	$x \wedge king(x)$
Content	$B_j bald(x)$

In the following table[7] the effect of the accommodations is given.

global:	John:	unification	to global:	to John:
$y \wedge king(y)$	$z \wedge king(z)$	$x = z$		
$\neg(y \wedge king(y))$	$z \wedge king(z)$	$x = z$		
ignorance	$z \wedge king(z)$	$x = z$		
$y \wedge king(y)$	$\neg(z \wedge king(z))$	$x = y$		x
$\neg(y \wedge king(y))$	$\neg(z \wedge king(z))$	FAILURE		
ignorance	$\neg(z \wedge king(z))$		$x \wedge king(x)$	x
$y \wedge king(y)$	ignorance	$x = y$		$x \wedge king(x)$
$\neg(y \wedge king(y))$	ignorance			$x \wedge king(x)$
ignorance	ignorance		$x \wedge king(x)$	$x \wedge king(x)$

It is equally necessary that the lexical presuppositions follow Heim. They are conditions on the applicability of the concept that triggers them and their failure would make a judgment to the effect that the concept holds impossible. It follows that the presupposition must hold locally as well as at the place of its antecedent if it has one. So even where an antecedent can be found, it is sometimes necessary to have accommodation.

As an example consider the trigger *regret*. It is usually taken to express the relation of being saddened by some event or state, the one given in the complement of the verb. For this the event or state is presupposed to exist (like the subject) and to be apperceived by the subject. This causes lexical presuppositions to the effect that that the event exists and that the subject believes that the event exists. It is however possible to argue against this approach by examples like (28), where Mary's leaving is not projected even though we are at the top of the stack.

[7] The first line of the table brings out a weakness of VdSandt's treatment. Intuitively, resolution should be possible to both the actual king and the king John assumes, with possibly an identification of both kings.

(28) John believes that Mary left and he regrets that. She never did go.

This suggests that *regret* should be analysed as the combination of attributing an emotive belief to the subject of *regret* (e.g. *sad(e)*) combined with presupposing the existence of the event e (i.e. the truth of the complement) with respect to the subject's belief state. If it is known that the subject believes the truth of the presupposition, projection is blocked, otherwise projection follows by accommodation[8].

(29) John regrets that Mary left.

Lexical presupposition:	$leave(e, mary)$
Content	$B_j sad(e)$

Global:	John	to global:	to John:
$leave(e, m)$	$leave(e, m)$		
$leave(e, m)$	$\neg leave(e, m)$	failure	
$leave(e, m)$	ignorance		$leave(e, m)$
$\neg leave(e, m)$	$leave(e, m)$		
$\neg leave(e, m)$	$\neg leave(e, m)$	failure	
$\neg leave(e, m)$	ignorance		$leave(e, m)$
ignorance	$leave(e, m)$	$leave(e, m)$	
ignorance	$\neg leave(e, m)$	failure	
ignorance	ignorance	$leave(e, m)$	$leave(e, m)$

Does this exhaust the different kinds of presuppositions? It seems that at least there is a third kind associated with triggers such as *too, also, another, again* etc. There is some empirical confirmation that these play a role in the bookkeeping involved in storing information by humans: the bookkeeping that prevents similar objects from being confused with each other, something that may easily happen given our propensity to identify similar things (*Stenning88*). An important difference is their different behaviour under accommodation and anaphora. It is possible for *too* and *another* to identify antecedents in parts of the context that would not normally be accessible and it is less possible to deal with them by means of accommodation. Access to normally inaccessible parts is illustrated in (30). The *modal subordination* effects observed for comparable cases in pronominal anaphora do not arise here. It should be noted however that the phenomenon is quite complex and does not arise equally clearly in all cases. As soon as some semantic effect of the triggers is present their access to inaccessible parts disappears.

[8] The accommodation of *John believes that Mary left* if Mary did not leave seems impossible. It is as if projection is obligatory if John only has an implicit belief that Mary left. This seems a problem for the current account of *regret*.

(30) If John has time, he will visit us tonight. Mary will come too.
If John will come tonight, we must warn Fred. Mary will come too.
John believes that Mary was in Egypt. Sue was there too.

7 Grammar and Stacks

7.1 Equations

The following is a set of equations that define updates in terms of stack operations.

We assume that for (certain) atomic formulas we have information states $[[A]] = \{i \in I : i \models A\}$. So we can define atomic updates by means of (*df.*21).

Def. 21 *Atomic Updates*

$$[A] = \mathbf{ass}\ \langle [[A]].\Sigma \rangle$$

Here **ass** , an operation that reduces the length of the stack by 1, takes in correctness of atomic updates.

Def. 22 *Addition*

$\mathbf{ass}\ \Sigma = \langle \Sigma_0 \cap \Sigma_{10}.\Sigma_{11} \rangle$ if
$\emptyset \subset \Sigma_0 \cap \Sigma_{10} \subset \Sigma_{10}$
else undefined.

For economy of notation we will continue to write atomic updates as $[A]$.

Complex formulas will be handled by the following equations.

Def. 23 *Complex Updates*

$$[\neg\varphi] = \mathbf{neg}\ [\varphi]\mathbf{up}$$
$$[\varphi \wedge \psi] = [\psi][\varphi]$$
$$[\varphi \rightarrow \psi] = \mathbf{neg\ neg}\ [\psi]\mathbf{up}\ [\varphi]\mathbf{up}$$
$$[\varphi \vee \psi] = \mathbf{neg\ neg}\ [\psi]\mathbf{up\ neg}\ [\varphi]\mathbf{up\ up}$$
$$[B_x\varphi] = \mathbf{belout}_x[\varphi]\mathbf{belin}_x$$
$$[regret(x, \varphi(e)] = \mathbf{belout}_x[sad(e)]\mathbf{pres}\ [\varphi(e)]\mathbf{one\ belin}_x$$
$$[bachelor(x)] = \mathbf{neg}\ [married(x)]\mathbf{up\ pres}\ [male(x)][adult(x)][human(x)]\mathbf{one}$$

The operation **neg** is defined as subtraction of the closure of the first element on the stack from the second, followed by stripping off the first element. Closure is defined as an operation **hull** that needs the discourse referents of an information state. On an information state itself, we can lay our hands on the variables that are existent in the information state, as in (*df.*24).

Def. 24 *Existent variables*

$$\textbf{exvar}\ \sigma = \{x \in VAR : \sigma \models x\}$$

On stacks, however, this gives us a definition of discourse markers. They are given as those existent markers that do not already exist in the next information state (if there is one).

Def. 25 *"Discourse markers"*

$$\textbf{dm}\ \Sigma = \textbf{exvar}\ \Sigma_0 \text{ if } \Sigma \text{ has length } 1$$
$$\textbf{dm}\ \Sigma = \textbf{exvar}\ \Sigma_0 - \textbf{exvar}\ \Sigma_{10}$$

The closure operation on the first state of the information stack can now be given in (*df.*26).

Def. 26 *Closure*

$$\textbf{hull}\ \Sigma = \langle \{i \in I : i =_{\textbf{dm}\ \Sigma} j \in \Sigma_0\}.\Sigma_1 \rangle$$

And finally negation.

Def. 27 *Negation*

$$\textbf{neg}\ \Sigma = \langle \Sigma_{10} - (\textbf{hull}\ \Sigma)_0.\Sigma_{11} \rangle$$

The way these three operations give the effect of adding the information from a negative sentence to the (first element) of an information stack is as follows. The operation **up** , defined in (*df.*28) pushes a copy of the first element of the stack to which it applies onto that stack. The scope of the negation updates the new stack. The first element of the result is closed off and subtracted from the original first element.

Def. 28 *Double First Information State*

$$\textbf{up}\ \Sigma = \langle \Sigma_0.\Sigma \rangle$$

Belief sentences are handled by two new operations $\textbf{belin}_x$ and $\textbf{belout}_x$. $\textbf{belin}_x$ determines the information state representing the beliefs of x according to the first information state on the stack and pushes it onto the stack. $\textbf{belout}_x$ considers the contents of the first information state as the beliefs of x and removes those information indices of the second that are in conflict with the assumption that the first information state entails the beliefs of x.

Def. 29 *Belief updating*

$$\mathbf{belin}_x\Sigma = \langle \bigcup_{i \in \Sigma_0} iix.\Sigma \rangle$$
$$\mathbf{belout}_x\Sigma = \langle \{i \in \Sigma_{10} : iix \subseteq \Sigma_0\}.\Sigma_{11} \rangle$$

Presupposition can be defined as a complex stack operation. In order to add the content of a presupposition to an information state by means of an operation **ass** that checks for informativity, it is necessary to make sure that the presupposition is not already entailed by the information state. This is achieved by adding the presupposition to 1, the empty information state which is pushed on the stack. Subsequently, we test/accommodate the resulting information over the rest of the stack and pop to the rest of the possibly changed stack. In (*df.*30) the operation **one** is defined that adds the empty information state 1 to the stack.

Def. 30 *Adding the empty information state*

$$\mathbf{one}\ \Sigma = \langle 1.\Sigma \rangle$$

In (*df.*31) follows the presupposition operation. It is based on Heim but allows skipping of intermediate contexts (when there is a reason for it) by clause c.

Def. 31 *Presupposition*

a. if $\exists! g\ (\mathbf{dom}\ g = \mathbf{dm}\ \Sigma \wedge \mathbf{cod}\ g = \mathbf{exvar}\ \Sigma_0 - \mathbf{dm}\ \Sigma \wedge \Sigma^g_{10} \subseteq \Sigma_0)$ then $\mathbf{pres}\ \Sigma = \langle \Sigma^g_{10}.\Sigma_{11} \rangle$
b. else if $\mathbf{pres}\ \langle \Sigma_0.\Sigma_{11} \rangle$ is defined and

(1) $\Sigma_0 \cap \Sigma_{10} \neq \emptyset$ and
(2) $\Sigma_0 \cap \Sigma_{10} \subset (\mathbf{pres}\ \langle \Sigma_0.\Sigma_{11} \rangle)_0$

then $\mathbf{pres}\ \Sigma = \langle \Sigma_0 \cap \Sigma_{10}.\mathbf{pres}\ \langle \Sigma_0.\Sigma_{11} \rangle \rangle$
c. else if $\mathbf{pres}\ \langle \Sigma_0.\Sigma_{11} \rangle$ is defined then $\mathbf{pres}\ \Sigma = \langle \Sigma_{10}.\mathbf{pres}\ \langle \Sigma_0.\Sigma_{11} \rangle \rangle$
d. else if $\Sigma_0 \cap \Sigma_{10} \neq \emptyset$ then $\mathbf{pres}\ \Sigma = \langle (\Sigma_0 \cap \Sigma_{10}).\Sigma_1 \rangle$
e. else undefined

Clause (a.) appeals to a notion σ^g which relabels discourse referents.

$$\sigma^g = \{i^g : i \in \sigma\}$$

where i^g is defined by putting $i^g(x) = i(g(x))$ for $x \in dom\ g$, and $i^g(x) = i(x)$ otherwise. So i^g is an information index like i itself with the difference —if it used as in clause (a.)— that the discourse markers of the information state of which it is an element are made identical to discourse markers of an information state further down the stack. The intersections in clause (b.) and (d.) push this identification down the stack.

So clause (a.) describes presupposition resolution, clause (b.) deals with complete global accommodation, clause (c.) with skipping and clause (d.) with local accommodation. The operation fails if no accommodation or resolution is possible, but does not cover the case of lexical presuppositions in their local contexts. Failure can here be ensured by demanding that the non-presuppositional meanings of triggers entail the truth of their lexical presuppositions. This would cause information states to become inconsistent if the requirement is not met, and seems natural enough in most cases.

We give some simple examples for the operation of (df.31.) Consider the update of (31)to 1.

(31) There is a king. The king sings.

Updating 1 with the first clause gives $1[x][king(x)]$. Presupposing $y \wedge king(y)$ starts by forming the complex stack $\langle 1[y][king(y)], 1[x][king(x)]\rangle$. Clause (a.) now applies for $g = \{< y, x >\}$ and delivers $\langle 1[x][king(x)][x = y]\rangle$. The final update gives: $\langle 1[x][king(x)][x = y][sing(y)]\rangle$.

Clause (d.) applies when (a.), (b.) and (c.) do not. Updating 1 with the second sentence of (31) is an example. We first get $\langle 1[y][king(y)], 1\rangle$, and from that $\langle 1[y][king(y)]\rangle$ and finally $\langle 1[y][king(y)][sing(y)]\rangle$. (a.) does not apply because 1 does not have the required information and the conditions for (b.) and (c.) are not met on an empty stack.

(e.) applies to (32).

(32) There is no king. The king sings.

Here the condition on clause (d.) is not met.

For (b.) and (c.) we need more complex examples. Consider updating 1 with (33).

(33) There is a king. John believes the king sings.

The trigger update leads to $\langle 1[y][king(y)], 1, 1[x][king(x)]\rangle$. (John's belief state still has no information after the first update.) (a.) does not apply therefore. But (a.) gives the result $\langle 1[x][king(x)][x = y]\rangle$ when applied to $\langle 1[y][king(y)], 1[x][king(x)]\rangle$. This meets the conditions in (b.) so we get the result $\langle 1[y][king(y)], 1[x][king(x)][x = y]\rangle$ and finally (34).

(34) $\mathbf{belout}_j\langle 1[y][king(y)][sing(y)], 1[x][king(x)][x = y]\rangle$

A very similar result is obtained when the addition of the king to the initial context results by clause (d.) (accommodation).

For clause (c.) consider the update of (35) to 1.

(35) John believes there is no king. John believes the king sings.

The initial update for the trigger gives (36).

(36) $\langle 1[y][king(y)], \mathbf{neg}\ \langle 1[x][king(x)], 1, 1\rangle\rangle$

($\mathbf{belin}_j$ composed with $\mathbf{belout}_j$ is an identity). Clause (a.) and (b.) do not apply, as the second element does not have the information that there is a king or can consistently be updated with their information. The third and last clause can however be updated in that way, so that clause (c.) applies, giving (37)by the entailment requirement for lexical presuppositions[9].

(37) $[y]\mathbf{neg}\ \langle 1[x][king(x)], 1, 1[y][king(y)]\rangle$

This then finally gives (38).

(38) $\mathbf{belout}_j[sing(y)][y]\mathbf{neg}\ \langle 1[x][king(x)], 1, 1[y][king(y)]\rangle$

The operation (φ) we had before, can be rendered as $\mathbf{pres}\ [\varphi]\mathbf{one}$.

The relation with DRT must be reasonably clear by now. We render atomic updates by putting things in boxes and negations by prefixing a negation sign to the update corresponding with the scope. The operation **up** corresponds with opening a new box, **one** with opening a notepad box, **belin** with a belief box etc. It is even possible to attempt to give a DRT "semantics" for our operations, as in (*df*.32). This time the operations apply to stacks of DRSs. (Here $\langle\emptyset,\emptyset\rangle$ is the empty DRS and $merge(A, B) = \langle A_0 \cup B_0, A_1, B_1\rangle$.)

Def. 32 *DRS updates*

$\mathbf{ass}\ = \langle merge(\Sigma_0, \Sigma_{10}).\Sigma_{11}\rangle$
$\mathbf{neg}\ = \langle \Sigma_{01} \cup \{NOT\ \Sigma_0\}.\Sigma_{11}\rangle$
$\mathbf{up}\ = \mathbf{one}\ = \mathbf{belin} = \langle\langle\emptyset,\emptyset\rangle.\Sigma\rangle$
$\mathbf{belout}_x = \langle \Sigma_{01} \cup \{BEL_x\Sigma_0\}.\Sigma_{11}\rangle$
$.^g$ = substitute markers according to g
pres see VdSandt.

The fact that DRSs are formal objects obliterates some of the distinctions we were able to make. We distinguish three types of subordinate boxes by initialising them in different ways. Because of the obliteration of these distinctions, correctness cannot be expressed directly anymore. The relation can also be

[9]Though the requirement that y exists in John's belief state is a necessary condition, it seems too minimalistic an account of *de re* belief. What we could define here is a relation *object(x,y)* holding in an information state if y is an object that is existent according to x. This makes it possible to express that an object is both an object of John and an existent object, or that an object is hared between John and Bill. What we do not capture however is the mechanism by which such relationships arise, i.e. the causal effects of objects and representations of objects on belief subjects that causes them to represent these objects themselves and the epistemic effects of such causations.

turned around. Then our efforts can be seen as giving a semantic interpretation of what goes on when we add material to subordinate DRSs as the DRS development algorithm instructs us to do. A semantics for these operations is not available in the bottom-up semantics that have been proposed for DRT, e.g. in *Zeevat89* and *Stokhof & Groenendijk90.*

In the following section, a grammatical formalism will be interpreted directly in terms of the stack operations defined in the current section and atomic updates. This completes the formalisation of VdSandt, as after all a solution to the projection problem is in the end a grammatical and compositional treatment of the syntactic and semantic properties of presupposition triggers.

7.2 Grammar

I will define a small fragment using a mock-prolog without the pretense that any of this will run. We use a prolog notation (capital letters for variables). The notation $A : B : \Sigma : T$ stands for the statement that $A : B : \Sigma : T$ is an expression with form A, category B, that transforms an information stack Σ into an information stack T.

To give some flavour, a treatment of the verb *believe.* The incoming information state Σ is enriched with the new information state that is the belief state of the subject of the verb according to the first information state of Σ, after taking in the information coming from the subject NP, which maps Σ to P. This forms the incoming stack $\mathbf{belin}_x P$ for the belief complement which updates $\mathbf{belin}_x P$ to become T. From T the first element is removed by the $\mathbf{belout}_x$ operator which codes the information in the next element on Σ.

Meeting the goals after the $\Leftarrow$ means that the updates are found for the subject NP and the belief complement so that, if Σ is instantiated to a particular information stack, the clause of which *believes* is the head will denote a concrete update of Σ.

(39)

$$\begin{array}{l} \langle \text{ NP, believes, that, S} \rangle : \text{sentence: } \Sigma : \mathbf{belout}_X \text{T} \Leftarrow \\ \qquad \text{NP:np(X):}\Sigma\text{:P} \\ \qquad \text{S:sentence:}\mathbf{belin}_X \text{ P:T.} \end{array}$$

The following two clauses are presuppositional referential phrases. Names are treated as involving two presuppositions: an existential one and an anaphoric one. To treat names as anaphoric has —under the pressure of problems— become the accepted practice in computational linguistics but can be justified theoretically precisely by the different treatment that one is forced to meet out to names in discourse representation theory: this shows that they are meant to refer to an already accessible discourse referent. Other arguments can be found in the distribution of (short) names in discourse. First, they can (in resumptions) be used in exactly the positions where pronouns and other short definites could occur. Second, they can be used referentially (but only supported by more explicit references that serve as antecedents, i.e. long versions of the name or compounds such as, *my friend doctor Watson*) also in situations where local uniqueness is not satisfied, e.g. discourses where two persons named John are around. In the case of the determiner *the* which still needs a noun, the behaviour is the same. The noun is put on a new empty information state (it consists of an existence statement for the new variable and the statement that the property associated with the noun holds of that new variable) which then is presupposed to obtain the semantic contribution of the whole NP.

Mary:np(X):Σ : $\mathbf{pres}\ [mary = X][X]\mathbf{one}\ \Sigma$.

$\langle$the, N$\rangle$:np(X):Σ : **pres** T $\Leftarrow$
N:noun(X):**one** $\Sigma : T$

The following two examples are two verbs *leave* and *regret*, where I take *regret* to add a property attribution $sad(e)$ to the belief state of the subject of regret and to presuppose the complement clause in the same belief state.

$\langle$NP, leaves$\rangle$: sentence(E): $\Sigma : [leave(E, X)][E]T \Leftarrow$
NP:np(X):$\Sigma : \Sigma 1$

$\langle$NP, regrets, that, S$\rangle$:sentence:$\Sigma : \Sigma 1 \Leftarrow$
NP:np(X):Σ:P,
S:sentence:**one** $\mathbf{belin}_X$ P:T,
$\Sigma 1 = \mathbf{belout}_X[sad(E)]\mathbf{pres}\ T))$

The only one of the next examples that needs a special comment is the tentative treatment of the cleft statement. For this it is necessary to assign an exhaustive interpretation to the WH-variable in the complement. Though exhaustivity can be treated in update semantics (see Zeevat *forthcoming*), it essentially involves use of the technique of pre-order updating pioneered by Veltman (*to appear*). Partly for this reason[10] no treatment of proper quantification is offered.

$\langle$if, S1, S$\rangle$:sentence:Σ : **neg neg** T $\Leftarrow$
S1:sentence:**up** $\Sigma : P$,
S:sentence:**up** $P : T$.

$\langle$it,is,NP, who, S$\rangle$:sentence:$\Sigma : [X = Y]\mathbf{pres}$ T) $\Leftarrow$
NP:np(X):$\Sigma : P$,
S:sentence/Y:**one** $P : T$.

[10] The other reason is type-raising in the current formalism, a discussion of which seems a distraction here.

bachelor:noun(X): $\Sigma : [unmarried(X)]$**pres** $[male(X)][adult(X)][X]$**one** Σ

man:noun(X):$\Sigma : [man(X)][X]\Sigma$

8 A Problem with Accommodation

The process we used here is a default process: accommodation happens as far down the stack as possible and on the intervening states as long as doing so does not lead to conflict with the correctness conditions.

There is a philosophical reason to be unhappy with the notion of accommodation we developed, since it does not seem to follow from the nature of presupposition as such. From what we understand of presupposition and why it occurs, it would rather follow that accommodation should always be as local as possible, as indeed Karttunen predicts.

But there are more empirical problems as well. First of all, the view of accommodation we developed does not lead to the right characterisation of the resolution of definite descriptions. There is a class of definite descriptions that are not meant to be resolved: their content is already sufficient to yield a referent without any contextual dependency. If one wants, these could be subject to accommodation to make their behaviour as much like proper names as possible. But the definite descriptions outside this class do not seem to participate in accommodation at all. They can either be resolved by finding a discourse object that meets the description or one that meets the description well enough or by being functionally related to a high focus discourse object. For the first case compare (40)

(40) A soldier entered the room. The man asked for a beer.

Though the predicate *soldier* does not strictly imply the predicate man, there is certainly a strong expectation here, which makes the resolution unproblematic. But it can be worse, as in (41), where genuinely new information is added.

(41) A man died in a car crash yesterday evening. The Amsterdam family father was found to have been drinking.

For bridging cases, compare (42)

(42) John went into the kitchen. The tap was running.
John got married last April. The priest was bald.

Here the definites are linked to the kitchen and the marriage respectively: they are the tap in the kitchen and the priest who celebrated the wedding respectively. The problem for our accommodation account is that if we do not have antecedents in each of the four cases the interpretation process is blocked and not as accommodation predicts continued in a routine way.

This is not to say that the resolution does not add new information in both cases. We infer that the soldier is a man, that the man who crashed was an Amsterdam family father, that the kitchen had a tap, that the marriage was performed by a priest etc. But this is not accommodation proper, which would also create the antecedents themselves.

It would be an improvement to add for presupposition resolution precisely the possibilities found for definites: the possibility of adding some not implausible material and the possibility of bridging to high focus elements. The conditions under which these resolutions are possible are not very sharply demarcated but nevertheless quite restrictive. It is possible to go from soldier to man, but not as easily from man to soldier[11]. Similarly linking calls on a relation of part and whole, that is hard to formalise, but nevertheless intuitively obvious. Here a marriage normally has a performer, a kitchen normally a tap etc., but not the other way around.

What this comes down to is giving resolution a larger and more realistic role in presupposition, which would decrease the role of accommodation. Perhaps it is then possible to reduce the explanation of projection to just global and strictly local accommodation, a position that is easier to defend than the one we arrived at. Global accommodation —unlike intermediate accommodation— can be seen as the further determination of an object that is not completely explicit from the ongoing discourse.

A sketch of the resulting algorithm would be:

1. Try to resolve allowing also bridging and adding material at the site of antecedent accompanied by accommodation between the antecedent and the trigger.
2. If this fails and the trigger is suitable for it use global accommodation and accommodation on the intervening path.
3. Try local accommodation.
4. Give up

A version of this could deal with examples like McCawley's (43).

(43)

a. LBJ dreamt that he was a homosexual and that everybody knew that his foreign policy was a failure.
b. LBJ dreamt that he was a homosexual and that everybody knew that he waited for boys in the restroom of the YMCA

[11] (41) is an apparent exception but can be brought into line by making a distinction between restrictive and non-restrictive parts in definite descriptions. Here the restrictive material is about the same as that of a male pronoun and the rest must be seen as an adjectival non-restrictive modifier.

In the (b.) example, the recognition of LBJ's behaviour in his dream as implying homosexuality provides a relationship like the one in (42) , so that projection is prevented. In the (a.) example, such a relationship cannot be constructed and projection occurs.

In addition it provides an approach to VdSandt's partial matching, as the resolution processes we now assume have the required soft boundaries. For the grandchildren-children example, it would be possible but difficult to bridge from *his children* to *his grandchildren.* If a bridge is built, John's having children is not projected, otherwise it is.

9 Conclusions and Open Questions

What did we learn from our comparison? In the first place we have established a strong similarity between Karttunen-Heim on the one hand and VdSandt on the other. The similarity is strengthened by our construction of discourse markers as proper names in the information states. This prevents a good many of the problems arising from logical omniscience. This is not to say that the problem of logical omniscience has been solved. The information states can still not distinguish between e.g. two equivalent mathematical statements, if they involve the same discourse referents.

Second, we have provided a reconstruction of Heim's theory of accommodation, in which global accommodation obtains the properties needed for dealing with the problems involved in Karttunen's earlier version of update semantics for presuppositions. Under this view, global accommodation is the default case. Unfortunately, we are in the same position as *Gazdar79, Soames82, VdSandt88, VdSandt89* and *Heim81* in being unable to provide an explanation of the fact that there is this default. Also, we have not succeeded in solving the scheduling problem in a satisfactory way. Future work will have to tell whether the approach to accommodation following *Mercer92* is the way to go.

Third, we have been able to correct a number of details. (a) It is necessary for developing a theory of presupposition under belief in DRT to involve the whole belief state of that person rather than limit oneself to the current belief report. (Unwanted accommodations are the result). In this respect update accounts are crucially better since they do not have the alternative of ignoring a person's other beliefs. (b) Resolution is more complex than we thought since it often involves local accommodations as well. This may seem a point against anaphoric theories of presupposition, but it is not as personal pronouns behave in exactly the same way. In (44)

(44) Mary met a man$_i$ and John believes that Harry thinks he$_i$ stole his watch.

it is necessary to accommodate the existence of the man in Harry's belief state according to John as well as in John's belief state. (c) *De re* readings for definites in belief contexts can be described as resolution and accommodation without (full) local accommodation. These should be allowed, even in Heim's position.

Fourth, we have established a difference in the behaviour with respect to accommodation of two classes of presuppositions, the lexical and the resolution ones. The second class is rather well understood since the recent wave of philosophical attention to anaphora. Or, more prudently, the conceptual problems by resolution problems are the same as addressed in the literature on anaphora. It is different with lexical presuppositions. Though the role of sortal concepts for individuation and identity has been investigated in depth, so that it may now be feasible to explain the presuppositional character of sortal information in terms of the concepts that have been dug out in that discussion, it does not hold that all lexical presupposition can be thought of as sortal information. Though preconditions for action are significant in explaining another class, there are important other cases. *Seuren88* mentions the case of the English *bald*, whose lexical presuppositions rule out that it can be used to translate the Dutch *een kaal landschap* or the German *eine kahle Landschaft* as *a bald landscape* although the kernel meaning of *bald* and *kaal* or *kahl* is the same. So it seems much remains to be done here.

The notion of updating stacks of information states may worry the theoretician. I have no argument to pacify such worries, but hope that the method contributes to clarify the postulated accommodation processes. The formalisation is closely related to DRT and could be used as an alternative model by those who prefer information states and constructa from information states to syntactic objects. Unlike other "semantics" for DRT (*Zeevat89, Stokhof & Groenendijk90, Asher*(MS)), the present one is top-down rather than bottom-up and provides therefore e.g. a more appropriate reconstruction of Kamp's proper name rule than the bottom-up approaches[12]. At the same time, as a semantical approach, it can be useful in ruling out syntactical operations on DRSs that could not be meaningfully interpreted within stack-updating.

This paper will have sequels where a formalisation of accommodation in default logic will be described and one which will document computational work along the lines set out in this paper.

References

Asher, N. Abstract Objects, Semantics and Anaphora. MS Center for Cognitive Science, University of Texas at Austin, 1990.

[12] In Zeevat89, the best result obtainable seemed to be that any text in which a name acts as an antecedent for a pronoun can be reconstructed by a quantification rule ambiguity.This is empirically adequate but does not do justice to the intuition behind the proper name rule in the DRS-development algorithm. On the current account, the proper name rule is a special case of presupposition and so provides an explanation of names for non-existent objects as in *Santa Claus does not exist* or *The Greeks believed that Pegasus was a winged horse*

Gazdar, G. *Pragmatics. Implicature, Presupposition and Logical Form.* Academic Press, New York 1979.

Heim, I. *On the projection problem for presuppositions*, *WCCFL* **2** (1983). p. 114-26.

Heim, I. Presupposition Projection and the Semantics of Attitude Verbs. *Journal of Semantics* 9, 3 (1992), p. 183-222.

Kamp, H. A Theory of Truth and Semantic Representation. In: J.A.G. Groenendijk, T.M.V. Janssen & M.B.J. Stokhof (eds.). *Formal Methods in the Study of Language.* Mathematical Centre, Amsterdam 1981. p. 277-322.

Kamp, H. & U. Reyle. *From Discourse to Logic.* Kluwer, Dordrecht 1990.

Karttunen, L. Presuppositions of Compound Sentences. *Linguistic Inquiry* 1973, p. 167-193.

Karttunen, L. Presupposition and linguistic context. *Theoretical Linguistics* **1**,1974, p.181-94

Karttunen, L. & S. Peters. Conventional Implicature. In: Oh & Dinneen(eds). *Syntax and Semantics 11: Presupposition.* Academic Press, New York 1979. p. 1-56.

Mercer, R. Default Logic: Towards a Common Logical Semantics for presupposition and Entailment. *Journal of Semantics* **9**, 3 (1992), p. 223-250.

Van der Sandt, R. *Context and Presupposition.* Croom Helm, London 1988.

Van der Sandt, R. Presupposition and discourse structure. In: Bartsch, van Benthem, van Emde Boas (eds.)*Semantics and Contextual Expression.* Foris, Dordrecht 1989.

Seuren, P. Lexical Meaning and Presupposition. In: Huellen, W & R. Schulze (eds). *Understanding the Lexicon.* Tuebingen 1988, p. 170-187.

Soames, S. How presuppositions are inherited: a solution to the projection problem. *Linguistic Inquiry* **13**(1982). p. 483-545.

Stalnaker, Robert. Assertion. In: Cole, Peter (ed.): *Syntax and Semantics 9: Pragmatics.* New York 1978, p315 - 332.

Stokhof, M. & J. Groenendijk. Dynamic Predicate Logic. In: *Linguistics and Philosophy* **14**, 1991.

Stenning, K., J. Levy & M. Sheperd. On the Construction of Representations for Individuals from Descriptions in Text. In: *Language and Cognitive Processes* **3** 1988. p. 129-163.

Veltman, F. Defaults in Update Semantics. Accepted by *Journal of Philosophical Logic.*

Zeevat, H. A Compositional Version of Discourse Representation Theory. In

Linguistics and Philosophy **12** 1991, p. 95-131.

Zeevat, H. Questions and Exhaustivity in Update Semantics. MS, University of Amsterdam 1992.

Contents