Discourse multiple dependencies

Claire Gardent
Computational Linguistics,
Universität des Saarlandes
Saarbrücken
E-mail: claire@coli.uni-sb.de

October 13, 1994

Abstract

It is sometimes claimed (cf. [Mann/Thompson 1988, Scha/Polanyi 1988, Webber 1991, Gardent 1991, Prüst 1992]) that discourse has a tree structure which reflects the semantic structure of discourse. In this paper, I argue that this claim is problematic in cases of discourse multiple dependencies i.e. cases where one discourse segment is semantically related to two discourse segments. I develop a discourse framework which is based on [Scha/Polanyi 1988] but integrates ideas from Feature-based Tree Adjoining Grammars (FTAGs). I then show that this new framework adequately captures multiple dependencies whilst retaining the precise linguistic predictions made by the discourse grammar. In particular, I show how it permits a simple modelling of discourse semantics, discourse coherence and discourse based constraints on anaphora resolution.

Contents

1 Introduction

Discourse theories often rest on two main assumptions[1]. First, discourse has a tree structure. Second, this tree structure reflects the semantic structure of discourse: the structural embeddings of the tree correspond to the semantic embedding of (possibly complex) propositions within more complex propositions by means of discourse relations. In other words, discourse theories often postulate both, a tree structure for discourse, and an isomorphism between discourse syntax and discourse semantics.

In this paper, we argue that the data conflicts with these assumptions. It is often the case that one discourse segment is semantically related to two distinct discourse segments. This means, either that discourse should be assigned a graph structure (in this way, one segment may have two parents and thereby be related to two distinct segments), or that the isomorphism between syntax and semantics needs to be relaxed (in this way, the semantic structure need not be a tree). In both cases, the two assumptions described above cannot be simultaneously maintained.

To account for this kind of data (which we shall refer to as "discourse multiple dependencies"), we propose to build on the discourse grammar approach (cf. [Scha/Polanyi 1988; Prüst 1992]), and to modify it using ideas from both incremental parsing (cf. [Webber 1991, Prüst 1992]) and Feature-based Tree Adjoining Grammars (FTAGs, cf. [Vijay-Shanker/Joshi 1988]). As we shall see, the crucial advantage of our approach is that it reconciles the precise predictions of discourse grammar with a correct account of multiple dependencies.

The paper is structured as follows. Section 2 summarises the discourse grammar approach, and shows how it yields precise definitions of such discourse concepts as discourse semantics, discourse coherence and discourse constraints on anaphora resolution. Section 3 introduces examples of discourse multiple dependencies and investigates various possible ways of amending the discourse grammar approach to account for this type of discourses. We argue that the incremental approach yields some interesting insights on how to solve the problem raised by discourse multiple dependencies. Finally, section 4 describes a new discourse grammar framework, namely Tree In-

[1]cf. [Mann/Thompson 1988, Scha/Polanyi 1988, Webber 1991, Gardent 1991, Prüst 1992].

serting Grammar (TIG). Based on ideas from both the incremental and the FTAG approaches, TIG correctly captures discourse multiple dependencies by relaxing the syntax/semantics isomorphism. We show by examples how it can be used to both capture discourse multiple dependencies, and make the appropriate predictions concerning anaphora resolution, discourse semantics and discourse coherence.

2 Discourse grammar

The approach described in [Scha/Polanyi 1988; Prüst 1992] is a paradigm example of the tree based, syntax/semantics isomorphism approach to discourse. The basic idea is that discourse can be described by means of a discourse grammar which in essence consists of a typed unification based sentence grammar augmented with a set of discourse grammar rules. These rules are the tools which predict both the tree structure of discourse and the isomorphism between discourse syntax and discourse semantics. Because they are interpreted under the standard context free rewrite procedure, they predict a tree structure for discourse[2]. Because they are unification based and permit a one-to-one relation between syntax and semantics, they predict an isomorphism between syntax and semantics. An example will illustrate this. Consider the following simplified discourse rule (where the syntactic category of a constituent is taken to be the type of its category and *dcu* is assumed to be a supertype of *contrastive_pair*; For simplicity, irrelevant features have been omitted):

(1)
$$\begin{bmatrix} \textit{contrastive\_pair} & \\ \textsc{sem} & \textit{contrast}(Sem_1,\ Sem_2) \end{bmatrix} \rightarrow \begin{bmatrix} \textit{dcu} & \\ \textsc{sem} & Sem_1 \end{bmatrix}, \begin{bmatrix} \textit{dcu} & \\ \textsc{sem} & Sem_2 \end{bmatrix}$$

At the syntactic level, this rule states that any two discourse constituents may combine together to form a discourse constituent with syntactic category *contrastive_pair*. At the semantic level, it requires that the propositions

[2] For the sake of discussion, I assume here that the discourse grammar is viewed as a context free grammar and that discourse parsing proceeds under a standard context free rewrite procedure. As we shall see later, the assumption made in [Scha/Polanyi 1988] is in fact different – this other case will also be discussed.

denoted by those two constituents be related by the semantic relation of *contrast*. That is, the structural embedding of the syntactic tree is matched by a semantic embedding of the subconstituents semantics within the semantics of the mother constituent.

In essence, the discourse grammar approach uses semantic knowledge (and in particular, knowledge about discourse relations) to build up a structure which can be exploited to explain various discourse phenomena. In particular, it can be used to make precise predictions about discourse based constraints on anaphora resolution, discourse semantics and discourse coherence. In what follows, we examine in more details how discourse grammar helps us capture these notions.

2.1 Discourse semantics

As illustrated by the above discourse grammar rule, the specification of discourse semantics is very simple. As in most unification based sentence grammars, the discourse grammar approach assumes a syntax/semantic interface which adheres to the rule-to-rule principle: for each syntactic rule, there is a corresponding semantic rule. In other words, each discourse rule licences both a complex syntactic constituent and a complex semantic unit. As a result, the discourse grammar approach assumes a compositional semantics where the meaning of a constituent is a function of the meaning of its parts. For instance, given the discourse rule in (1), the discourse *Jon likes bees but Peter hates them* will be assigned the semantics $contrast(like(j,b), hate(p,b))$ where the meaning of the whole discourse is a function of the meaning of its parts.

Thus in the discourse grammar approach, the specification of discourse semantics simply is a side effect of parsing, whereby the semantics of a given discourse is represented by the SEM value of the category labelling the root node of the parse tree. In this way, discourse grammar provides a simple but precise way of defining discourse semantics.

2.2 Discourse based constraints on anaphora

One frequent claim in the discourse literature is that discourse structure constrains anaphora resolution (cf. [Grosz/Sidner 1986; Webber 1988; Nakhimovshky 1988; Webber 1991; Gardent 1991, Prüst 1992]). In the discourse grammar approach, this intuition can be modelled using two attributes IN and OUT which represents the flow of anaphoric information: IN represents the in-going context (where a context is a list of discourse markers in the sense of Discourse Representation Theory, cf. [Kamp 1981]) and OUT represents the out-going context. Discourse based constraints on anaphora resolution result from the fact that only those discourse markers present in the in-going context list are available for resolution. In this way, the discourse grammar approach makes precise predictions as to how discourse constrains anaphora resolution. To see how this works, consider the following discourse:

(2) a. *Hannah likes ants,*
b. *Thomas likes bees*
c. *and Jessy hates them.*

Suppose that the grammar restricts IN and OUT information as follows. First, lexical entries[3] are as illustrated below (where + denotes list concatenation):

$$\begin{bmatrix} \text{PHON} & \textit{Hannah\_likes\_ants} \\ \text{IN} & I_0 \\ \text{OUT} & I_0 + [h, a] \end{bmatrix}$$

That is, the out-going context consists of the in-going context augmented with the list of discourse markers introduced by the current lexical entry. Second, the grammar rules regulates the anaphoric flow as sketched below:

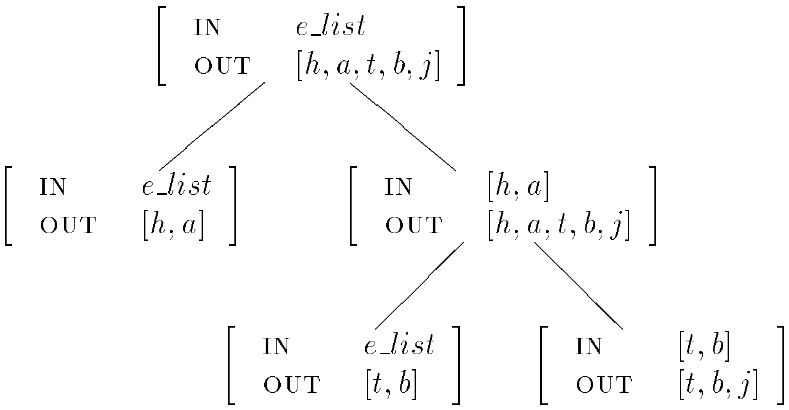
$$\begin{bmatrix} \text{IN} & I_0 \\ \text{OUT} & I_0 + O_2 \end{bmatrix} \rightarrow \begin{bmatrix} \text{IN} & \textit{e\_list} \\ \text{OUT} & O_1 \end{bmatrix}, \begin{bmatrix} \text{IN} & O_1 \\ \text{OUT} & O_2 \end{bmatrix}$$

That is, only the anaphoric information that is local to a left sister (i.e. O_1) is passed on. Global information (i.e. I_0) is only used for upward percolation so that all anaphoric information contained in a tree is available to a right sister constituent.

Given such a grammar, consider the example given above. Let us assume that the discourse grammar allows (b) to combine with (c), and (a) to combine with the complex segment (bc). The resulting tree and the associated

[3] For simplicity, we assume that basic DCU (i.e. clauses) are lexical items. In practice however basic DCUs are complex constituents derived by the sentence grammar.

anaphoric information can then be sketched as follows:



The point is that the in-going context of (c) only contains information about *Tom* and *bees* – in this way, we capture the intuition that *them* in (c) may not resolve to *ants* in (a), because (c) is not related by any discourse relation to (a) and thus does not belong to the same discourse segment.

More generally, discourse grammar allows us to make precise the intuition that anaphors may only resolve to antecedents occuring in the same discourse segment.

2.3 Discourse coherence

Discourse grammar also provides a way of measuring discourse coherence (cf. [Prüst 1992; Grover *et al.* 1994]). This is carried out by introducing an additional attribute – the SCHEMA attribute – whose relational value denotes the semantics common to all the subconstituents of a given constituent. Two relations are used to calculate this value: priority union [4] and generalisation. These are defined as follows.

Definition 2.1 *(Generalisation)*
Given two feature structures A and B, the generalisation of A and B is the

[4] For more information about priority union (also called, default unification) see [Bouma 1990, Calder 1990, Carpenter 1994, Kaplan 1987].

most specific feature structure C such that C subsumes[5] both A and B.

Definition 2.2 *(Priority Union)*
Given two feature structures A and B, the priority union of B with respect to A is the most specific feature structure C such that A subsumes C and C unifies with B.

$$\begin{bmatrix} \textit{contrastive\_pair} & \\ \textsc{rsem} & \textit{contrast}(Sem_1, \boxed{\textit{1}}) \\ \textsc{schema} & \textit{gen}(Schema_1,\ Schema_2) \end{bmatrix} \rightarrow$$

$$\begin{bmatrix} \textit{dcu} & \\ \textsc{sem} & Sem_1 \\ \textsc{rsem} & Sem_1 \\ \textsc{schema} & Schema_1 \end{bmatrix}, \qquad \begin{bmatrix} \textit{dcu} & \\ \textsc{sem} & Sem_2 \\ \textsc{rsem} & \boxed{\textit{1}}\ \textit{pr\_union}(Sem_1,\ Sem_2) \\ \textsc{schema} & Schema_2 \end{bmatrix}$$

$$\textit{Sue\_likes\_bees} \rightarrow \begin{bmatrix} \textit{basic\_dcu} & \\ \textsc{sem} & \begin{bmatrix} \textit{like} & \\ \textsc{arg}1 & \textit{sue} \\ \textsc{arg}2 & \textit{bees} \end{bmatrix} \\ \textsc{rsem} & \boxed{\textit{1}} \\ \textsc{schema} & \boxed{\textit{1}} \end{bmatrix}$$

$$\textit{Thomas\_hates\_them} \rightarrow \begin{bmatrix} \textit{basic\_dcu} & \\ \textsc{sem} & \begin{bmatrix} \textit{hate} & \\ \textsc{arg}1 & \textit{thomas} \\ \textsc{arg}2 & \textit{entity} \end{bmatrix} \\ \textsc{rsem} & \boxed{\textit{1}} \\ \textsc{schema} & \boxed{\textit{1}} \end{bmatrix}$$

Figure 1: A discourse grammar fragment

[5] Intuitively, X subsumes Y if Y is at least as specific/informative than Y. For a formal definition of subsumption, see [Shieber 1986].

To show how these relations can be used to calculate discourse coherence, we now run through a simple example. Suppose that the grammar is as given in Figure 1 and that the discourse to be parsed is:

(3) a. *Sue likes bees*

b. *but Thomas hates them.*

The tree assigned by the grammar to this discourse can be sketched as follows (for readability, the SEM values have been abbreviated; RSEM stands for *resolved semantics*):

$$\begin{bmatrix} \textit{contrastive\_pair} & \\ \text{RSEM} & \textit{contrast(like(s,b),}\ \boxed{1}\textit{)} \\ \text{SCHEMA} & \textit{gen(like(s,b),}\ \boxed{1}\textit{)} \end{bmatrix}$$

$$\begin{bmatrix} \textit{dcu} & \\ \text{SEM} & \textit{like(s,b)} \\ \text{RSEM} & \textit{like(s,b)} \\ \text{SCHEMA} & \textit{like(s,b)} \end{bmatrix} \qquad \begin{bmatrix} \textit{dcu} & \\ \text{SEM} & \textit{hate(t,entity)} \\ \text{RSEM} & \boxed{1}\ \textit{pr\_union(like(s,b),hate(t,entity))} \\ \text{SCHEMA} & \boxed{1} \end{bmatrix}$$

That is, first, priority union is applied to the SEM values of the two daughters – this triggers the resolution of any parallelism dependent anaphor occuring in the right most daughter. Second, generalisation is applied to the SCHEMA values of the two daughters – this computes the semantic information shared by the two daughters. The results of applying these two operations is depicted in Figure 2. Two points are worth noting. First, the underspecified element *them* in the second clause (represented by *entity* in the corresponding feature structure) is resolved to *bees* as a result of applying priority union. Second, generalisation yields *em_att(human,bees)* as the SCHEMA value of the discourse, which is intuitively correct as a measure of the semantic information shared by the subconstituents of discourse (3)

More generally, generalisation yields an intuitive definition of discourse coherence: whatever is common to two discourses is preserved and whatever is incompatible is generalised to the most specific common sort.

$$\begin{array}{ccc} \phi & \psi & pr\_union(\phi,\psi) \\ \begin{bmatrix} like & \\ \textsc{arg1} & sue \\ \textsc{arg2} & bees \end{bmatrix} & \begin{bmatrix} hate & \\ \textsc{arg1} & tom \\ \textsc{arg2} & entity \end{bmatrix} & \begin{bmatrix} hate & \\ \textsc{arg1} & tom \\ \textsc{arg2} & bees \end{bmatrix} \end{array}$$

$$\begin{array}{ccc} \phi & \psi & gen(\phi,\psi) \\ \begin{bmatrix} like & \\ \textsc{arg1} & sue \\ \textsc{arg2} & bees \end{bmatrix} & \begin{bmatrix} hate & \\ \textsc{arg1} & tom \\ \textsc{arg2} & bees \end{bmatrix} & \begin{bmatrix} em\_att & \\ \textsc{arg1} & human \\ \textsc{arg2} & bees \end{bmatrix} \end{array}$$

Figure 2: Applying priority union and generalisation

In summary, discourse grammar allows for precise predictions to be made concerning discourse semantics, discourse coherence and discourse constraints on anaphora resolution. However, as we shall now see, discourses exhibiting multiple dependencies cause problem.

3 Discourse grammar and discourse multiple dependencies

3.1 The data

The following discourse provides an example of discourse multiple dependencies.

(4) a. *Hannah likes ants.*
   b. *Thomas likes bees.*
   c. *But Jessy hates them.*

Intuitively, this discourse can be analysed as follows: there is a relation of parallelism between (a) and (b), and a relation of contrast between (b) and (c). In other words, (b) is semantically related to two distinct propositions. We say that (b) illustrates a case of discourse multiple dependency.

Now because of the notoriously slippery definition of discourse relations and discourse segments, discourse analysis tends to be highly subjective and thus, prone to controversy. As a result, one can quibble over the interpretation of the above example – which relations are assigned between which discourse segments is highly theory dependent and almost always controversial. Nonetheless – irrespective of which discourse relations are postulated – multiple dependencies seems to be a basic phenomenon in discourse. As [Webber 1991] puts it "some discourse segment belongs to two adjacent segments in a sequence essentially serving as a transition between them". Indeed, closer examination of the discourse literature shows that this phenomenon has already been repeatedly remarked upon. For instance, in [Hobbs, 1985], the following example is discussed:

(5)
a. *I had already learned a little bit about hitchhiking.*
b. *I'd split out two or three times and then come back.*
c. *My first trip had been to geneva new york.*
d. *and then twice i'd gone to california.*
e. *and then I'd cut down through the south.*
f. *and I'd sort of covered the united states.*
g. *one very beautiful summer I'll tell you about some other time*
h. *that I spend in idaho.*
i. *that to this day I remember with nothing but happiness,*
j. *it was so beautiful*
k. *I'll never forget it*
l. *right up in the mountains in these tall pine forests*
m. *and it was something that you know is just*
n. *it's indelibly in my memory*
o. *and nothing could ever erase it.*
p. *we'll come back to it one day.*
q. *yeah sometime you ask me about that.*

The relations posited by Hobbs for this discourse are the following:

A. parallel(c,d,e)
B. elaboration(b,A,f)
C. exemplification(B,g)
D. elaboration(a,C)
E. elaboration(l,m)
F. elaboration(k,E,n)
G. elaboration(i,j)
H. parallel(h,G,F)
I. elaboration(g,H)

As Hobbs himself remarks, (g) is an example of multiple dependencies. It is related both to (B) by an exemplification relation – it exemplifies a hitchihiking trip – and to (H) by an elaboration relation – it is described in more detail by the remainder of the discourse.

In short, multiple dependencies are a common phenomena in discourse. In the following section, we discuss how the discourse grammar approach can be amended to deal with them.

3.2 Possible solutions

There are several ways in which the discourse grammar approach can be amended to account for multiple dependencies. In this section, we examine three possible solutions namely, the graph analysis, the reentrancy approach and the incremental account. We argue that the incremental approach, although inappropriate in many respect, yields some interesting insights on how the problem can be solved.

The graph approach

As illustrated in [Reichman 1985], one possibility for dealing with multiple dependencies is to assume that discourse has a graph (rather than tree) structure. In this way the isomorphism between syntax and semantics is maintained, and multiple dependencies can be accounted for. Although intuitively appealing, this approach fails to preserve the simple but precise insights provided by discourse grammar. In particular, it fails to preserve it simple account of discourse semantics, discourse coherence and discourse based constraints on anaphora resolution.

Consider discourse semantics first. Under the tree based approach, the semantics of a discourse can simply be taken to be the semantics associated with the root node of the tree assigned by the grammar to this discourse. However, if a graph structure is assumed, a root node may (and often will) not exist. Hence some additional device must be thought of which would allow discourse grammar to both assign multiple dependency discourses a graph structure and a global semantics.

The specification of discourse coherence is equally affected by the absence of root node. As we saw in section 2, Prüst's discourse grammar assigns to any complex DCU a SCHEMA value which encodes the semantic information shared by its sub-constituents. For instance, the SCHEMA value associated with the discourse in (6) will be the SCHEMA value associated with the root node of the discourse tree namely *like(insect)(human)*.

(6) a. *Hannah likes ants.*
b. *Thomas likes bees.*
c. *Jessy likes flies.*

But again, under a graph based approach (and in cases of multiple dependencies) no root node will be available and hence the simple definition of discourse coherence given by tree based discourse grammar is no longer available.

Finally, consider the analysis of discourse based constraints provided by discourse grammar. As discussed in section 2.2, the general idea underlying this analysis is that only a specific part of the discourse tree is accessible at any time during processing and that it is this part of the tree which determines which antecedents are available for anaphora resolution. But if discourse is not assigned a tree structure, a new proposal must be put forward to explain the interaction of discourse structure and anaphora resolution.

In summary, if a graph based approach is to be adopted to deal with discourse multiple dependencies, and if the insights provided by the discourse grammar approach are to be preserved, new ideas need to be developed to make the two compatible. As we shall see below, we prefer to follow an alternative solution: preserve the tree based approach and modify it to capture multiple dependencies.

Using structure sharing

There is a clear sense in which discourse multiple dependencies are reminiscent of syntactic multiple dependencies. In the sentence given in (7), the NP *Mary* can be viewed as being both the subject of *to be honest* and the object of *believes*.

(7) *Jon believes Mary to be honest*

In unification based frameworks, such multiple dependencies are usually dealt with by using reentrancy: one and the same feature structure contributes to the semantic information associated with two distinct verbs. In this way, a tree structure can be maintained while multiple dependencies may still be accounted for. So one possibility when attempting to deal with discourse multiple dependencies would be to use reentrancy in the grammar rules in such a way that it correctly captures these multiple dependencies. Under such a proposal, the tree based approach to discourse structure is maintained but the isomorphism between syntax and semantics isn't. However this solution is not very satisfactory because at the level of discourse – contrary to the syntactic level – no structural or lexical constraints apply to multiple dependencies. As a result, rules have to be artificially multiplied and parsing will become highly ambiguous.

The incremental approach

In [Prüst 1992], the discourse grammar rules are not treated as context free rewrite rules. Instead they are viewed as construction rules[6], where the basic idea is that the "discourse parse tree" (DPT) is built in a strictly left-to-right, incremental fashion and that construction rules are in effect tree transformations on this DPT. More precisely, the effect of a construction rule on the DPT can be defined as follows:

Definition 3.1 *(Construction rules)*
let τ be a discourse parse tree with right frontier[7] *RF_τ, let R be a construction*

[6]In fact, Prüst divides discourse rules into two distinct types, namely construction rules and extension rules. I ignore extension rules here because they are irrelevant to the discussion.

[7]the right frontier of a tree is the set of all right most nodes occuring in this tree including the root node.

rule of the form:

$$C \rightarrow C_1 C_n$$

where C, C_1 *and* C_n *range over discourse constituent units (DCUs). Then for any node* $n \in RF_\tau$ *where* n *is labelled with a category* $C_{1'}$ *such that* $C_{1'}$ *unifies with* C_1*, execution of* R *on* n *will yield a new tree* τ' *just like* τ *except that* n *is replaced with a tree* τ'' *such that the mother of* τ'' *is labelled with* C*, the daughters of* $\tau_{''}$ *are labelled with the categories* C_1 *and* C_n *respectively and* C_1 *is unified with* $C_{1'}$

That is, application of a construction rule will replace a node in the tree with the local tree licensed by this rule provided that the category labelling this node unifies with the left corner of this rule.

One interesting property of the incremental account is that it allows for multiple dependencies. Intuitively, the reason for this, is that the construction rules allow a node in the discourse tree to be used twice, once in relation with some left adjacent discourse segment and once in relation with some right adjacent discourse segment. To see this, consider the example in (8).

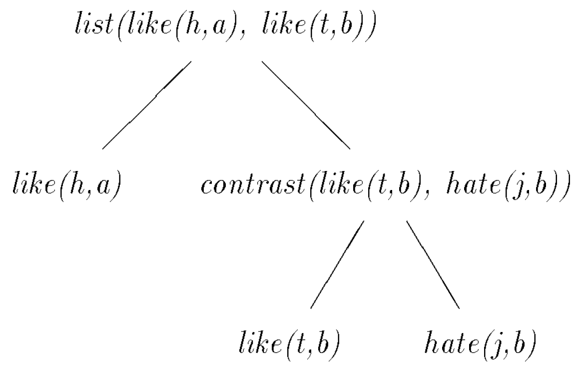
(8)
a. *Hannah likes ants.*
b. *Thomas likes bees.*
c. *But jessy hates them.*

Under the incremental approach described in [Prüst, 1992], segments (a) and (b) will first combine to yield a tree of the form:

list(like(h,a), like(t,b))

like(h,a)　　*like(t,b)*

The (construction) rule for contrast will then modify this tree as follows:



That is, the result of applying the construction rule to the original discourse tree is a tree which encodes both the relation of (b) to (a) and the relation of (b) to (c). Construction rules allow us to have our cake and eat it: discourse is still assigned a tree structure, but multiple dependencies can be accounted for. In other words, as in the reentrancy account, the tree based approach to discourse is maintained, but the isomorphism between syntax and semantics isn't.

Unfortunately, there are a number of problems with the incremental approach, all of which can be traced to the fact that when insertion proceeds, the information introduced by the local tree being inserted fails to be percolated up.

First, there is no obvious way in which the semantics of the discourse can be read off the tree: neither the root nor the right frontier of the discourse tree contains the relevant information. That the root does not contain this information is obvious: if the information associated with inserted trees is not percolated up the tree, then neither will their semantic information and hence the semantics of the root node may fail to represent the meaning of the whole discourse (for instance, in example (7) above, the semantics associated with the root node of the associated tree fail to contain information about (c)). To see that the right frontier of the tree does not contain the relevant semantic information consider the following example (taken from [Prüst 1992]):

(9) a. *We were going to see our son tonight*

b. *but we are not*

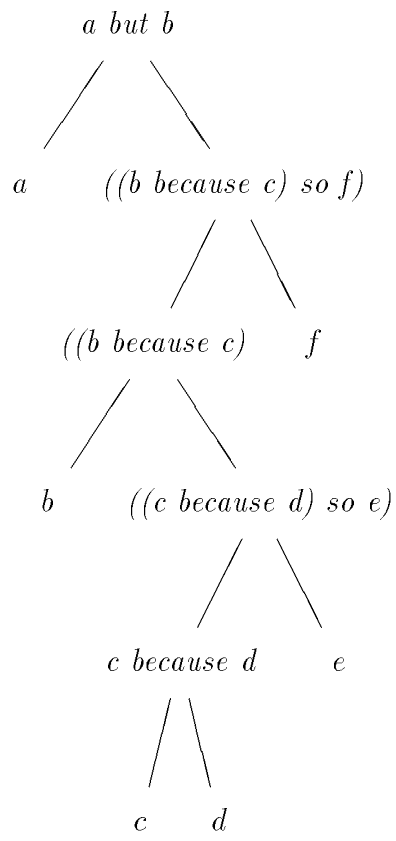
c. *because the younger one is coming home for dinner*

d. *because he is working in the neighbourghood*

e. *so he is coming home for dinner*

f. *so we are not*

Under the incremental approach described in [Prüst 1992], the tree associated with this discourse is:



If the right frontier is taken to determine the semantics of the discourse being described, the discourse semantics will then be:

((a but b) and ((b because c) so f) and f)

That is, the following information is lost:

(((c because d) so e) and e and d and b and a)

A second problem with the incremental approach is that the predictions made by the theory about anaphora resolution are inadequate. Consider first nominal anaphora. As discussed in section 2, the assumption captured in the discourse grammar approach, is that the discourse entities introduced in a discourse segment d_1 delimit the set of potential antecedents available to anaphors occuring in any discourse segment d_2 such that d_1 and d_2 are related by some discourse relation. This intuition is captured in the grammar rules by having all the discourse entities introduced in sub-segments be percolated up to their mother so that they become available for any anaphor occuring in a discourse segment related to their mother. However given that inserted information is not percolated up, this will not be true under the incremental account. To take a concrete example, consider the discourse in (10):

(10) a. *Hannah likes ants.*
b. *Thomas likes bees.*
c. *But Jessy hates them.*
d. *They are all friends though.*

If, as described above, the tree for (a), (b) and (c) is built by insertion, then the root node contains no information about (c) and thus no information about *Jessy*. Hence *they* in (c) cannot refer to the set of *Hannah, Thomas* and *Jessy*.

A similar problem occurs with propositional anaphora (or discourse deixis, cf. [Webber, 1991]). This is because, as discussed above, the nodes in the tree do not have a compositional semantics. That is, the semantics associated with a node is not necessarily a function of the semantics of the set of nodes it dominates. In those cases where it does not, this might yield the wrong resolution for propositional anaphors. As a case in point, consider the discourse in (11).

(11) a. *Hannah likes ants.*
b. *Thomas likes bees.*
c. *But Jessy hates them.*
d. *That's all I know about them.*

Here, the propositional anaphor *that* refers to the complex proposition *(list(a,b) and contrast(b,c))* – however this antecedent will not be available at the root of the tree dominating (a), (b) and (c); instead the semantic information associated with the root will be *list(a, b)*. As a result, there is no obvious way in which the propositional anaphor can be appropriately resolved.

A third problem with the incremental approach is that the predictions made by discourse grammar about discourse coherence information become inadequate. To see this consider the SCHEMA information associated with the discourse in (8). As the list rule applies to relate (a) with (b), the value of the schema attribute for the mother node will be $like(human, insect)$, which is fine. However as the construction rule for rhetorical subordination applies to relate (b) to (c) the schema value of the root remains unchanged. This is intuitively wrong. On hearing discourse (8), the common information between (a), (b) and (c) is taken to be *em_att(human,insect)* and not *like(human,insect)*.

To conclude, the incremental parsing approach provides some interesting insights on how to capture discourse multiple dependencies. However, like the graph approach, it fails to preserve the precise predictions made by discourse grammar about discourse based constraints on anaphora resolution, coherence information and discourse semantics specification. In what follows, we develop a framework for discourse which reconciles the "constructional insight" of the incremental approach with the precise predictions permitted by discourse grammar.

4 Tree Inserting Grammars

In the preceding section, we observed that incremental parsing permits the modelling of multiple dependencies. However, we also noted that if fails to preserve the precise predictions of discourse grammar, mainly because it does not allow for inserted information to be percolated up the tree. How can we reconcile the two approaches?

Interestingly, Feature-based Tree Adjoining Grammars integrate a similar idea as the incremental approach, the idea that one node can be used twice. In FTAGs, the adjunction operation allows for one tree to be inserted within another tree *and* for inserted information to be percolated up. The framework we propose (TIGs) simply is a modified version of FTAGs, one that (like the incremental approach) restricts insertion to nodes on the right frontier of the discourse parse tree. In this way, we model the fact that once a discourse segment is closed off (i.e. embedded within another constituent), this discourse segment can no longer be combined with.

4.1 The grammar formalism: TIGs

A Tree Inserting Grammar (TIG) consists of a finite set of elementary binary trees and a composition operation (called *insertion*) which combines trees into bigger (derived) trees. (The intuition is that elementary trees describe discourse rules and basic DCUs, while derived trees describe discourse structure). As in FTAGs, tree nodes are decorated with two feature structures called TOP and $BOTTOM$. As we shall see, this allows us to capture the fact that insertion might stretch a node into two. Intuitively, TOP information describes how a node relates to its upper tree, whereas BOTTOM information describes the relation of a node to its subtree.

The main difference between TIGs and FTAGs is that the combining operation of TIGs (insertion) combines three trees into one, whereas in FTAGs, adjunction combines two trees into one. The idea is that two trees correspond to the discourse segments being combined[8] whereas the third one corresponds to the discourse rule being applied. The insertion operation can be represented informally by the following pictures where Figure 3 describes the trees to be combined and Figure 4 illustrates the result of combining these trees by insertion (in Figure 3 the lines linking T'$_1$ to T, B'$_1$ to B$_1$, T$_2$ to T'$_2$ and B'$_2$ to B$_2$ indicate required unifications; In Figure 4 the $\wedge$ symbol is used to represent unifications).

More formally, insertion is defined as follows.

Definition 4.1 *(Insertion)*
The insertion operation combines an elementary tree α with two derived trees β_1 and β_2 as follows. Let β_1 be a tree whose right frontier contains a node

[8] Note that while adopting the construction rule idea of the incremental approach, we do not adhere to strict incrementality. That is, in contrast with a strictly incremental approach where construction always combines the discourse tree parsed so far with a *basic* discourse unit, insertion may combine the discourse tree with a non basic discourse segment i.e. a complex one.

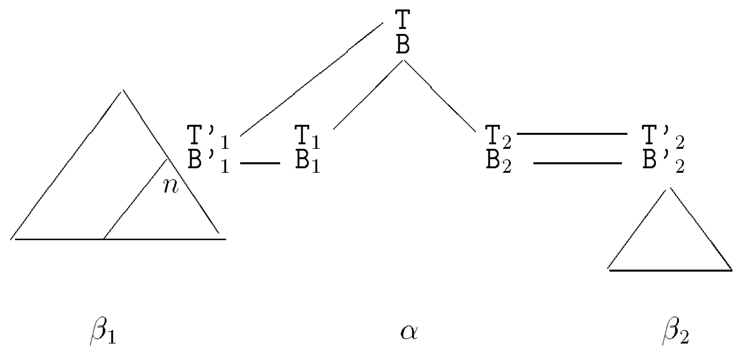


Figure 3: The insertion operation

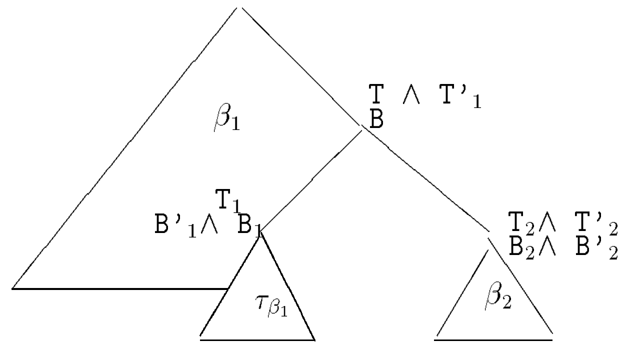


Figure 4: The result (β") of inserting α and β_2 at node n in β_1.

n with top and bottom information T_1' and B_1' respectively. Let β_2 be a tree whose mother node is labelled with categories T_2' and B_2'. Let α be an elementary tree whose leftmost and rightmost daughters are labelled with the categories T_1, B_1 and T_2, B_2 respectively, and whose mother node is labelled with T and B. Then inserting α and β_2 in β_1 at node n will yield the tree β" such that:

1. *The sub-tree τ_n of β_1 dominated by n is excised and the category T_1' labelling n is dissociated from n,*
2. *α is attached at n,*
3. *The sub-tree τ_n is attached at the left corner of α and for each of its nodes, bottom and top information are unified,*
4. *β_2 is attached at the right corner of α*
5. *T_1' is unified with T, T_2 with T_2', B_2 with B_2' and B_1' is unified with B_1.*

In essence, the adjoining operation inserts a tree at a node occuring on the right frontier of the discourse tree. Furthermore, the TOP information of the node being inserted at is unified with the TOP information of the rule mother whereas its BOTTOM information is unified with the BOTTOM information of the rule left corner. Finally, the tree being excised is "closed off" in that for all nodes in this tree, BOTTOM and TOP information is unified. This captures the fact that nodes that are not on the right frontier (cf. section 3) can no longer be combined with. Since no insertion will ever apply to these nodes we can safely assume that their TOP and BOTTOM information is in effect identical. By closing off the sub-tree being excised we further ensure that the root node of this sub-tree contains all the information contained in this tree. Or in other words, we ensure that the information contained in the nodes of this tree is percolated up to the mother node (see below).

4.2 Some example derivations

We now show by means of examples how TIGs can be used to solve the problems described in section 3.2.

Specifying discourse semantics.

Consider first the specification of discourse semantics. The idea is that contrary to the incremental approach, TIGs allow us to percolate information up the tree. This means in particular that semantic information will be percolated up from the daughter nodes to the root. Suppose for instance, that the discourse rules are encoded as in Figure 5 (R, A, B, TA and TB are variables over feature structures).

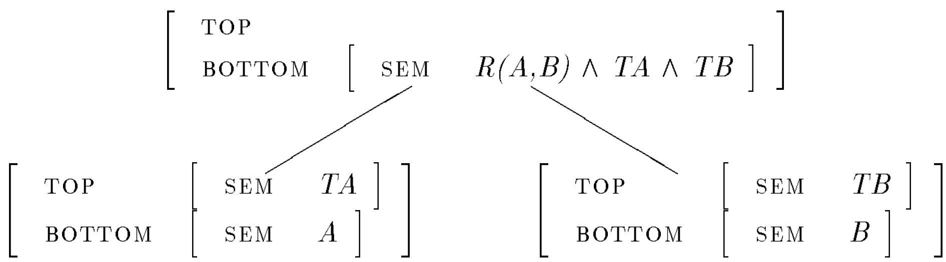


Figure 5: Percolation of semantic information within TIGs elementary trees

In essence, such a rule states that the semantics of the mother node consists of the semantics of the immediate daughters (A and B) conjoined with the semantics (TA and TB) of whatever has been inserted at the nodes labelled with the daughter categories licensed by this specific rule[9]

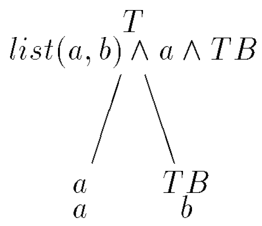
Suppose further that in lexical entries, semantic information is a BOTTOM feature. Now consider the discourse in (8) repeated as (12) for convenience.

(12) a. *Hannah likes ants.*
b. *Thomas likes bees.*
c. *But Jessy hates them.*

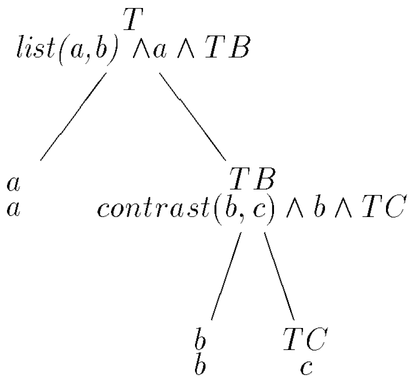
The derivation for this discourse can be summarised as follows. First, the leaf trees for (a) and (b) are combined. In the process (a) is closed off. The result can be as sketched below (irrelevant information has been omitted, and

[9]For simplicity (and contrary to section 2.1), we assume here that the semantics of a discourse is a direct function of the semantics of its parts. That is, we do not use priority union to resolve parallel dependent anaphors. This does not affect the argument.

tree nodes are labelled with an abbreviated semantic representation where a, b and c stand for the propositional content; for instance, a abbreviates $like(hannah, bees)$):



In the second step of the derivation, the rule for contrast applies to the (b) node in (1), thus yielding the following tree.



When parsing stops for lack of input, the TOP and BOTTOM information of all nodes is unified. Thus TC unifies with c and TB with $(contrast(b, c) \wedge b \wedge TC)$. The semantics of the root node can be represented as follows:

$$list(a, b) \wedge a \wedge\ contrast(b, c) \wedge b \wedge c$$

That is, a, b and c are asserted, a *list* relation holds between a and b and a relation of *contrast* holds between b and c. Just as required. Note in particular, that the multiple semantic dependency of (b) has been adequately captured.

Another interesting observation is as follows. First, observe that in this case, the semantics of the two local trees involved are conjoined: first (a) combines with (b) to yield the semantics ($list(a, b) \wedge a \wedge TB$), second (b)combines with (c) to yield the semantics ($contrast(b, c) \wedge b \wedge c$) –the resulting semantics associated with the root node of the parse tree is ($list(a, b) \wedge contrast(b, c) \wedge a \wedge b \wedge c$ which is indeed the conjunction (TB is unified to b) of the semantics associated with each local tree produced during the derivation. We call this a *conjoining derivation*. Now, consider another possible derivation of the discourse in (12). Suppose that (b) is first combined with (c) under the list rule thus yielding a tree of the following form:

$$\begin{array}{ccc} & \begin{array}{c} T1 \\ list(b,c) \wedge b \wedge TC \end{array} & \\ / & & \backslash \\ \begin{array}{c} b \\ b \end{array} & & \begin{array}{c} TC \\ c \end{array} \end{array}$$

Suppose further that the second step in the derivation combines this first derived tree with the basic tree associated with (a). Then by insertion the resulting tree is:

$$\begin{array}{ccc} & \begin{array}{c} T0 \\ list(a, (contrast(b,c) \wedge b \wedge TC)) \wedge a \wedge T1 \end{array} & \\ / & & \backslash \\ \begin{array}{c} a \\ a \end{array} & & \begin{array}{ccc} & \begin{array}{c} T1 \\ contrast(b,c) \wedge b \wedge TC \end{array} & \\ / & & \backslash \\ \begin{array}{c} b \\ b \end{array} & & \begin{array}{c} TC \\ c \end{array} \end{array} \end{array}$$

Closing off the tree yields the following semantics for the root node:

$$list(a, (contrast(b, c) \wedge b \wedge c)) \wedge contrast(b, c) \wedge a \wedge b \wedge c$$

Under this second (embedding) derivation, the semantics of the discourse segment relating (b) to (c) is embedded in the semantics of the segment relating (a) to the complex discourse segment encompassing (b) and (c).

At first sight, this might seem inappropriate. In the particular example being discussed, the embedding derivation yields the wrong interpretation: a is not in a list relation to $contrast(b, c)$but to b. There are two answers to this objection. First, under the grammar described in [Prüst 1992], this second derivation will actually not be produced since the list rule requires that the generalisation of the semantics being related be non-trivial. This is not the case when a is related to $contrast(b, c)$ – in this case, the most specific common denominator of the two propositions being compared will simply be the value *true* – hence the list rule cannot be applied to combine (a) with the complex segment formed by (b but c). Second, there are cases for which the two derivations reflect an existing semantic ambiguity and for those cases, the two derivations indeed yield the desired readings. To see this consider the following discourse (example from Hub Prüst, personal communication):

(13) a. *Dick does not come to work*
b. *because the trains aren't running.*
c. *(and) buses aren't either.*

Depending on the intonational properties of this discourse, two readings are possible which can schematically be represented as follows:

Reading 1: *cause(a, list(b,c))*
Reading 2: *cause(a,b)* $\wedge$ *list(b,c)*

As should be obvious from the preceding discussion, these two readings are precisely those produced by the alternative, embedding versus conjoining derivations.

Resolving anaphora

In section 3.2, we noted that the incremental approach failed to provide the appropriate notion of antecedent availability. This was shown to hold both for nominal anaphora and for discourse deixis (or propositional anaphora).

In effect, a solution to the criticism concerning propositional anaphora has already been presented in the preceding section. The point about incremental parsing and the resolution of propositional anaphora was this: if semantic information is not percolated up the tree, then propositional anaphors which refer to the (compositional) semantics of an open node in the discourse tree cannot be resolved appropriately (cf. example (8), section 3.2). But since (as shown in the previous section), our proposal does ensure that semantic information is percolated up the tree, it also ensures that propositional anaphors be appropriately resolved. Hence this particular problem is solved.

Consider now the problems raised by nominal anaphora. Here the problem is that anaphoric information (i.e. information about potential antecedents) also needs to be percolated up the tree if nominal anaphors are to be resolved appropriately (cf. example (8), section 2.2). We now illustrate by an example how the TOP/BOTTOM distinction allows for the correct percolation of anaphoric information. Recall (cf. section 2.2) that we represent anaphoric information using two features IN and OUT and that grammar rules regulate the flow of anaphoric information. Using TIGs TOP and BOTTOM features, we can reformulate the rule sketched in section 2.2 as illustrated in Figure 6. Given the discourse in (7), the derivation will pro-

$$\left[\ \text{BOTTOM} \left[\begin{array}{ll} \text{IN} & I_0 \\ \text{OUT} & I_0 + O_2 \end{array}\right]\right]$$

$$\left[\ \text{TOP} \left[\begin{array}{ll} \text{IN} & e\_list \\ \text{OUT} & O_1 \end{array}\right]\right] \qquad \left[\ \text{TOP} \left[\begin{array}{ll} \text{IN} & O_1 \\ \text{OUT} & O_2 \end{array}\right]\right]$$

Figure 6: Percolation of anaphoric information within TIGs elementary trees

ceed as described in the preceding section except that the last clause (d) will be attached as the sister of the root node of the tree spanning (a), (b) and (c). The flow of anaphoric information within the derivation tree is illustrated in Figure 7 (for each pair X,Y labelling a tree node, X denotes the IN value and Y the OUT value). On closing off the tree, the following unifications occur:

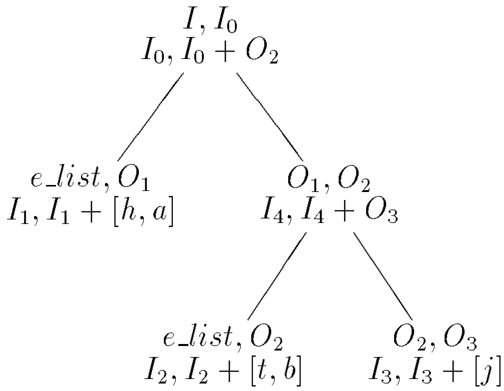


Figure 7: Distribution of anaphoric information for example (7)

1.	I_1	$= e\_list$
2.	O_1	$= [h, a]$
3.	I_2	$= e\_list$
4.	O_2	$= [t, b]$
5.	I_3	$= O_2$
6.	O_3	$= I_3 + [j]$
7.	O_3	$= [t, b, j]$
8.	O_1	$= I_4$
9.	I_4	$= [h, a]$
10.	O_2	$= I_4 + O_3$
11.	O_2	$= [h, a, t, b, j]$
12.	O	$= O_2$
13.	O	$= [h, a, t, b, j]$

Thus the out-going list of discourse referents associated with the root node does contain information about *Jessy* (namely the discourse referent j). Providing that a treatment of plural anaphora be provided, this means that the plural anaphor *them* in (7) can be appropriately resolved. More generally, the TOP/BOTTOM distinction allows us to percolate any information that needs to be. Hence if the assumption is that anaphoric information need to be percolated, TIG provides the appropriate framework for doing so.

Schema calculation

By using TOP and BOTTOM information adequately, we can provide a solution to the problem described in section 2.3. Let us assume that the schema information in rules is distributed as shown in figure 8.

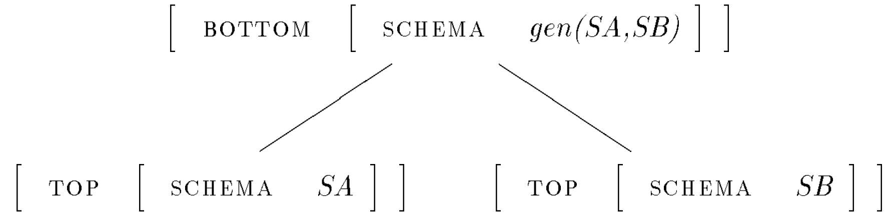
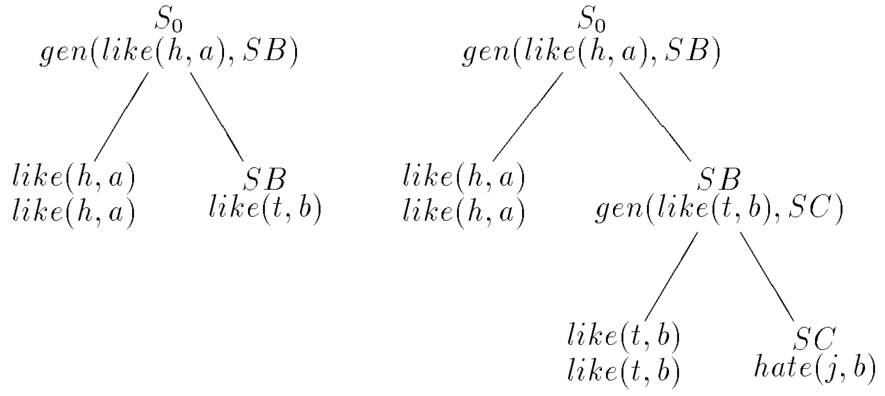


Figure 8: Percolation of schema information within TIGs elementary trees

Furthermore, we assume that at the lexical level, schema information is a BOTTOM feature. Given these assumptions the main derivation steps for the discourse in (8) can be represented as follows.



$$gen(like(h,a), gen(like(t,b), hate(j,b))$$
$$gen(like(h,a), gen(like(t,b), hate(j,b)))$$

$$like(h,a) \qquad gen(like(t,b), hate(j,b)$$
$$like(h,a) \qquad gen(like(t,b), hate(j,b))$$

$$like(t,b) \qquad hate(j,b)$$
$$like(t,b) \qquad hate(j,b)$$

So that the schema value of the whole discourse is $emot\_att(human, insect)$ just as required.

5 Conclusion

In this paper, we have questioned some basic assumptions of discourse theory concerning discourse structure and its relationship to discourse semantics. We have shown that the data did not warrant the combined hypothesis that discourse has a tree structure and that this structure is isomorphic to semantic structure. Furthermore, we argued that an FTAG like approach allows one to relax the isomorphism between syntax and semantics in such a way that precisely this data that conflicts with the above hypothesis, can be handled. In conclusion, we would like to point to some further, related issues.

A first point bears on the linguistic adequacy of the proposed approach. In the present paper, we have concentrated on specific cases of multiple dependencies, namely cases where the related discourse constituents are adjacent to each other. This raises the question whether non adjacent multiple dependencies occur in discourse and whether TIGs could handle such data. More generally, two questions arise. First, what are the semantic patterns of discourse? And more specifically, can two discourse constituents be semantically related by several discourse relations? Can non adjacent discourse segments be semantically related? Can one discourse constituent be related to more than two distinct discourse constituents? Although most theories[10]

[10] A notable exception to this generalisation is [Asher 1993] where semantic structure is

assume that none of those three possibilities is realised in discourse, the present paper has shown that at least some of the data conflicts with the basic assumptions of discourse theory. Hence, it would be interesting to pursue the issue further and proceed to a more careful examination of the data. A second question concerns the formal framework to be used for discourse processing. If the data is as complex as suggested above, tree based approaches have to be either modified or abandonned. Which of these two possibilities is most appropriate, we leave here as an open research question.

A second point which would deserve further investigation concerns the issue of processing with TIGs. As argued in [Gardent/Bloembergen 1994], the introduction of such operations as priority union and generalisation make the grammar non declarative and non monotonic. Although this is rather unfortunate, it can also be shown that this does not threaten the reversibility of the grammar: given a discourse grammar D, it is indeed possible to find both a generation and an analysis program that enumerate all possible solutions to a given query and stops. What remains to be shown is whether this generalises to TIGs.

Acknowledgements

This paper owes much to comments from, and discussion with, Laszlo Kalman, Hub Prüst, Remko Scha and Henk Zeevat. The idea of using the top and bottom features of FTAGs is due to Henk Zeevat. I would also like to acknowledge the financial support of the LRE project 61-062 *Towards a declarative theory of discourse* and of the Deutsche Forschungsgesellschaft project SAMOS *Semantic Analysis in a Modular System.*

References

[Asher, 1993] Asher, N. (1993). *Reference to abstract objects in discourse.* Dordrecht: Kluwer.

[Bouma, 1990] Bouma, G. (1990). Defaults in Unification grammar. In *Proceedings of the 28th Annual Meeting of the Association for Computational Linguistics*, Pittsburgh, Pennsylvannia, pp. 165–173.

a graph, and tree structure is derived from the embeddings of Discourse Representation Structures.

[Calder, 1990] Calder, J. (1990). *An interpretation of paradigmatic morphology.* PhD thesis, University of Edinburgh.

[Carpenter 1994] Carpenter, B. (1994). Skeptical and credulous default unification with applications to templates and inheritance. In T. Briscoe *et al,* (eds). *Inheritance, defaults and the lexicon,* pp. 13–37. Cambridge, Cambridge University Press.

[Gardent 1991] Gardent, C. (1991). Gapping and VP Ellipsis in a unification based grammar. PhD Thesis, University of Edinburgh.

[Gardent and Bloembergen 1994] Gardent, C. and S. Bloembergen (1994). Generating discourse. LRE Project 61–062, Deliverable C.2, University of Amsterdam.

[Grover *et al.*, 1994] Grover, C., Brew, C., Manandhar, S. aand M. Moens. Priority union and generalization in discourse grammars. To appear in *Proceedings of the Association for Computational Linguistics.*

[Grosz and Sidner, 1986] Grosz, B. and C. Sidner (1986). Attention, Intention and the structure of discourse. *Computational Linguistics,* 12, p.175–204.

[Hobbs 1985] Hobbs, J. On the coherence and structure of discourse. Ms. SRI international, Menlo Park, California.

[Kamp 1981] Kamp, H. (1981) A theory of truth and semantic representation. In *Formal methods in the studz of language,* Groenendijk, J., Janssen, T. and M. Stokhof (eds.), Amsterdam, Mathematisch Centrum.

[Kaplan 1987] Kaplan, R. M. (1987). Three seductions of computational psycholinguistics. In P.J. Whitelock *et al,* (eds.), *Linguistics theory and computer applications,* pp. 149–188. London: Academic Press.

[Mann and Thompson 1988] Mann, W and S. Thompson (1988). Rhetorical structure theory: Toward a functional theory of text organization. In *Text 3.*

[Nakhimovsky, 1988] Nakhimovsky, A. (1988). Aspect, aspectual class and the temporal structure of narrative. *Computational Linguistics,* 14, p.29–43.

[Prüst, 1992] Prüst, H. (1992) On discourse structuring, VP anaphora and Gapping. PhD Thesis, Universiteit van Amsterdam.

[Reichman, 1985] Reichman, R. The context space approach to discourse structure. Ms. University of California, La Jolla.

[Scha and Polanyi, 1988] Scha, R. and L. Polanyi (1988) An augmented context free grammar for discourse. In *Proceedings of the 12th International Conference on Computational Linguistics*, Budapest, pp. 573–577.

[Shieber, 1986] Shieber, S (1986). An introduction to Unification-based Approaches to Grammar, CSLI Lecture Notes 4, Stanford.

[Vijay-Shanker and Joshi, 1988] Vijay-Shanker, L. and A. Joshi (1988). Feature based TAGs. In *Proceedings of the 12th International Conference on Computational Linguistics*, Budapest, pp. 573–577.

[Webber, 1988] Webber, B.L. (1988) Discourse deixis: reference to discourse segments. In *Proceedings of the Association for Computational Linguistics*, pp.113–122.

[Webber, 1988] Webber, B.L. (1988) Tense as discourse anaphor. In *Computational Linguistics,* Volume 14, Number 2, pp.61–73.

[Webber, 1991] Webber, B.L. (1991) Structure and Ostension in the interpretation of discourse deixis. In *Language and Cognitive Processes,* 6 (2), pages 107–135.