D-STAG: A discourse formalism using synchronous TAG

Laurence Danlos
ALPAGE - LATTICE
Université Paris 7
Institut Universitaire de France
Laurence.Danlos@linguist.jussieu.fr

Abstract We propose a new formalism for discourse, called D-STAG, which is inspired by SDRT as a discourse theory and which is akin to D-LTAG as a formalism which extends a sentential TAG syntax/semantic interface to the discourse level.

Keywords: Discourse, SDRT, (Synchronous) Tree Adjoining Grammars, Syntax/Semantic

1 Introduction

We propose a new formalism for discourse, called D-STAG, which is inspired by SDRT as a discourse theory and which is akin to D-LTAG as a formalism which extends a sentential TAG syntax/semantic interface to the discourse level.

SDRT – Segmented Discourse Representation Theory (Asher, 1993; Asher & Lascarides, 2003) – came after RST – Rhetorical Structure Theory (Mann & Thompson, 1988; Taboada & Mann, 2006). These two popular discourse theories rely upon *discourse relations*. They share the idea that some parts of a discourse, called *Satellites*, play a “subordinate" (“less important") role relative to other parts, called *Nuclei*. This asymmetry is akin to the syntactic distinction between modifiers (satellites) and heads (nuclei). It leads to posit the existence of two types of discourse relations: a *coordinating (multinuclear)* relation links two Nuclei, while a *subordinating (nucleus-satellite)* relation links a Nucleus and a Satellite. It allows the construction of hierarchical discourse structures richly annotated with coordinating and subordinating relations.

In D-STAG, discourse analyses are also hierarchical structures richly annotated with coordinating and subordinating discourse relations. They can deterministically be converted into SDRT discourse structures. As a consequence, D-STAG can take advantage of the results brought by this *discourse theory*. For example, D-STAG can use SDRT's discourse update to infer discourse relations and it can call upon SDRT's Right Frontier Constraint so as to greatly simplify the computation of discourse structures.

D-STAG is like D-LTAG — Discourse Lexicalized TAG in the version presented in (Forbes-Riley *et al.*, 2006) — in that both formalisms extend a sentential syntax/semantic interface to the discourse level. The idea behind D-LTAG and D-STAG is to build a complete integrated text understanding system which incorporates the same mechanisms for the sentence and discourse levels. Using the same mechanisms for sentences and discourses is theoretically justified since a multi-sentensential discourse and a single sentence – without any discourse connective (subordinating or coordinating conjunction) – can exhibit the same discourse relations. This is the case in the pair in (1).

(1)a. John held out a bone to the dog. She caught it quickly.
b. John held out a bone to the dog who caught it quickly.

The phenomenon illustrated in (1) is not yet handled in SDRT[1]. An obvious explanation for that is that SDRT does not say a word about the sentence level, neither for syntax nor for semantics. A book such as (Asher & Lascarides, 2003) focuses on discourse interpretation – a very tough topic – so it only presents (constructed) multi-sentential discourse examples made up of quite short sentences (fewer than 10 words). Therefore, the syntactic analysis (parsing) of sentences is not touched on and their semantic analysis (as DRSs) is assumed to be straightforward. However, this does not reflect the reality observed in corpora examples. In corpora, most sentences include around twenty words and their syntactic and/or semantic analysis is still a hot research topic. In short, if one aims at building a complete integrated text understanding system, both the sentence and discourse levels must be taken into account and share the same interpretation models to handle pairs of discourses such as (1).

TAG – Tree Adjoining Grammar (Joshi, 1985) – has been successfully used as a syntactic formalism implemented in parsing systems for numerous languages and it has been extended to cover semantic analyses of sentences, mainly in two approaches: either the syntax/semantic interface uses ideas from Minimal Recursion Semantics (Copestake *et al.*, 1999) and Hole Semantics (Bos, 1995)– see the work of (Joshi *et al.*, 2007; Kallmeyer, 2002) – , or it uses STAG – Synchronous TAG (Shieber, 1994; Shieber & Schabes, 1990)– see (Nesson & Shieber, 2006; Nesson & Shieber, 2007). TAG has also been extended to the discourse level, first in the NLG perspective (Danlos, 1998; Danlos, 2001), next in the NLU perspective with D-LTAG using the first approach for the syntax/semantic interface and D-STAG that we present now using the second approach.

We leave aside here discussions on the pros and cons of the two approaches for the TAG syntax/semantic interfaces. We prefer the S-TAG approach, but we do not justify this preference here. On the other hand, we want to focus on a crucial difference between D-LTAG and D-STAG for the discourse level: D-LTAG does not rely upon discourse relations and ignores the distinction between coordinating and subordinating discourse relations, contrarily to D-STAG. D-LTAG has thus nothing in common with SDRT (nor with RST) and cannot benefit from the results brought by this discourse theory, especially at the rhetorical and pragmatic levels.

In a nutshell, D-STAG is designed so as to (i) use S-TAG for processing the syntactic and semantic sentence levels, (ii) extend this syntax/semantic interface to the discourse level while being based on a discourse theory, namely SDRT, (iii) interleave the sentence and discourse levels not only to get a homogeneous process from a discourse to its interpretation (efficiency reasons) but also to handle pairs such as (1) (theoretical reasons).

The paper is organized as follows. Section 2 briefly introduces SDRT, Section 3 STAG. Section 4 presents D-STAG data structures, Section 5 D-STAG processing. Section 6 compares D-LTAG and D-STAG.

The same reference discourse is used throughout the paper, namely (2) of the form S_1 *because*

[1]Nevertheless, Nicholas Asher, Manfred Stede and myself submitted a proposal to study this kind of phenomenon in English, German and French.

S_2. *Next* S_3., in which *because* conveys the subordinating relation Explanation and *next* the coordinating relation Narration.

(2) John went to the supermarket because his fridge was empty. Next, he went to the movies.

2 Brief introduction to SDRT

The SDRT graph for (2) is represented in Figure 1. The nodes are either "sentence nodes" (noted π_i) or "scope nodes" (noted π', π''). Sentence nodes are labels for the DRSs giving the logical forms of the sentences. Two sentence nodes are linked by an arrow labeled by a discourse relation R. The arrow is horizontal if R is coordinating, while it is vertical (oblique) if R is subordinating (Asher & Vieu, 2005). Taking into account the Nucleus/Satellite distinction, this means that an horizontal arrow links two Nuclei, while a vertical arrow goes from a Nucleus down to a Satellite. Scope nodes are linked by *lines* (and not arrows) to sentence nodes.

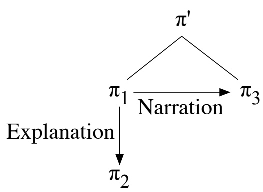

Figure 1: SDRT graph for discourse (2)

In the process of building discourse structures, SDRT calls upon the notion of *right frontier*, originally proposed in (Polanyi, 1988). Informally, in an SDRT graph for a discourse with n sentences (clauses), the right frontier contains the node π_n representing the last sentence, and the sentence nodes which are on the rightmost frontier of the graph[2]. As an illustration, the right frontier in the graph of Figure 1 contains only node π_3. In the dynamic construction of a discourse structure, the discourse constituents on the right frontier are the only available nodes for attachment of new information. This is known as the "Right Frontier Constraint"[3]. This constraint greatly simplifies the building of discourse structures, hence the computation of semantic representations for discourses, which are derived from discourse structures.

For the reader familiar with RST, we present the RST tree for (2) – in the graphical representation proposed in (Marcu, 2000) – in Figure 2. The symbol C_i stands for sentence S_i, its syntactic analysis or semantic representation (according to the application in which RST is used). The labels N and S on the edges stand respectively for Nucleus and Satellite.

(Marcu, 2000) has laid down the "Nuclearity Principle": "When a discourse relation is postulated to hold between two spans of a discourse, then it should also hold between the Nuclei of these two spans". The Nuclearity Principle gives the predicate-argument dependencies that must be derived from an RST tree. For example, it indicates that C_1 is the left argument of Narration in the RST tree for (2) given in Figure 2.

[2]In fact, the right frontier also contains the *topic nodes* which are on the rightmost frontier of the graph. The notion of *topic node* is important in SDRT, but left aside in this paper.

[3]This constraint also states that the antecedent of an anaphoric expression must be ("DRS-accessible") on the right frontier, but anaphoric expressions are not discussed here.

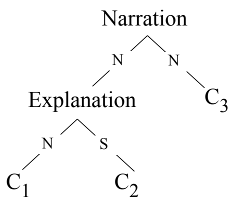

Figure 2: RST tree for discourse (2)

As explained in (Danlos, 2007b; Danlos, 2006a), the unlimited applicability of the Nuclearity Principle has the following consequence: some felicitous discourses are *de facto* excluded by RST. This is the case for the discourse in (3), in which the antecedent of *this* is the interpretation of its left context, namely the causal relation linking the interpretations of the two first sentences.

(3) Fred is upset because his wife is abroad for a week. This proves that he does love her.

The RST discourse structure for (3) should be the tree in the left hand side of Figure 3, in which the Nucleus argument of Comment is the sub-tree rooted at Explanation. This predicate-argument relation is not possible in RST because of the Nuclearity Principle which states that the Nucleus argument of Comment is C_1. On the other hand, (3) is not a problem in SDRT: its discourse structure is the graph in the right hand side of Figure 3. This graph includes brackets around π_1, π_2 and the arrow labeled Explanation, which means that π_1 and π_2, linked by Explanation, form a *complex constituent*.

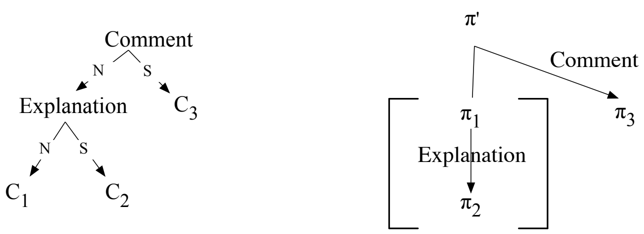

Figure 3: RST tree and SDRT graph for (3)

When comparing the discourse structures of (2) and (3), we see that *because* in (2) introduces a modifier while in (3) it is used to form a complex constituent. In both cases, it conveys the discourse relation Explanation, however in order to explicitly state the difference between these two uses, we say in D-STAG that *because* conveys Explanation in (2) and [Explanation] in (3) (Section 4).

SDRT focuses mainly on how to infer discourse relations which are *not* made explicit through a discourse connective[4]. D-STAG takes the SDRT mechanisms which have been set up for these cases (Section 5), but we posit the existence of an empty connective noted ε (the empty connective is also used in D-LTAG). As an illustration, when two sentences are simply juxtaposed, e.g. *John fell. Max pushed him.*, we lay down that the discourse is of the form S_1. ε S_2. and by misuse of language, we say that the empty connective "conveyed" Explanation, for example.

[4]A discourse connective is a subordinating or coordinating conjunction, or an adverbial such as *next* or *therefore*.

3 Introduction to Synchronous TAG

This section is reproduced except where noted from (Nesson & Shieber, 2006) with permission of the authors. It begins with a brief introduction to the use of TAG in syntax.

"A tree-adjoining grammar (TAG) consists of a set of elementary tree structures and two operations, substitution and adjunction, used to combine these structures. The elementary trees can be of arbitrary depth. Each internal node is labeled with a nonterminal symbol. Frontier nodes may be labeled with either terminal symbols or nonterminal symbols and one of the diacritics $\downarrow$ or $*$. Use of the diacritic $\downarrow$ on a frontier node indicates that it is a *substitution node*. The *substitution* operation occurs when an elementary tree rooted in the nonterminal symbol A is substituted for a substitution node labeled with the nonterminal symbol A. Auxiliary trees are elementary trees in which the root and a frontier node, called the *foot node* and distinguished by the diacritic $*$, are labeled with the same nonterminal. The *adjunction* operation involves splicing an auxiliary tree with root and designated foot node labeled with a nonterminal A at a node in an elementary tree also labeled with nonterminal A. Examples of the substitution and adjunction operations on sample elementary trees are shown in Figure 4."

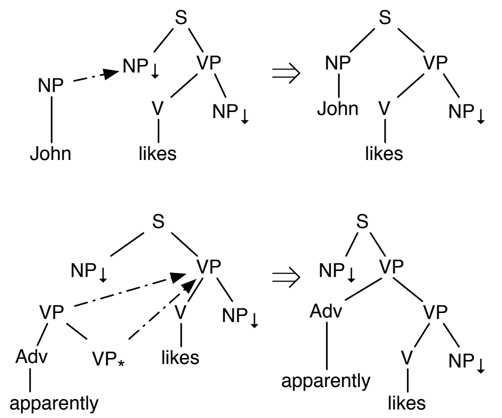

Figure 4: Example TAG substitution and adjunction operations (Reproduced from (Nesson and Shieber, 2006))

"Synchronous TAG (STAG) extends TAG by taking the elementary structures to be pairs of TAG trees with links between particular nodes in those trees. An STAG is a set of triples, $\langle t_L, t_R, \frown \rangle$ where t_L and t_R are elementary TAG trees and $\frown$ is a linking relation between nodes in t_L and nodes in t_R (Shieber, 1994; Shieber & Schabes, 1990). Derivation proceeds as in TAG except that all operations must be paired. That is, a tree can only be substituted or adjoined at a node if its pair is simultaneously substituted or adjoined at a linked node. We notate the links by using circled indices (e.g. ①) marking linked nodes."

STAG has been successfully used in an English sentential syntax/semantics interface (Nesson & Shieber, 2006; Nesson & Shieber, 2007). This interface is illustrated in Figure 3 with the parsing of the sentence "John apparently likes Mary": (a) contains the sample English syntax/semantic grammar fragment, (b) shows the derived tree pair, (c) the derivation tree. In derivation trees, "substitutions are notated with a solid line and adjunctions are notated with a dashed line. Note that each link in the derivation tree specifies a link number in the elementary tree pair. The links provide the location of the operations in the syntax tree and in the semantics tree. These operations must occur at linked nodes in the target elementary tree pair. (...) The resulting

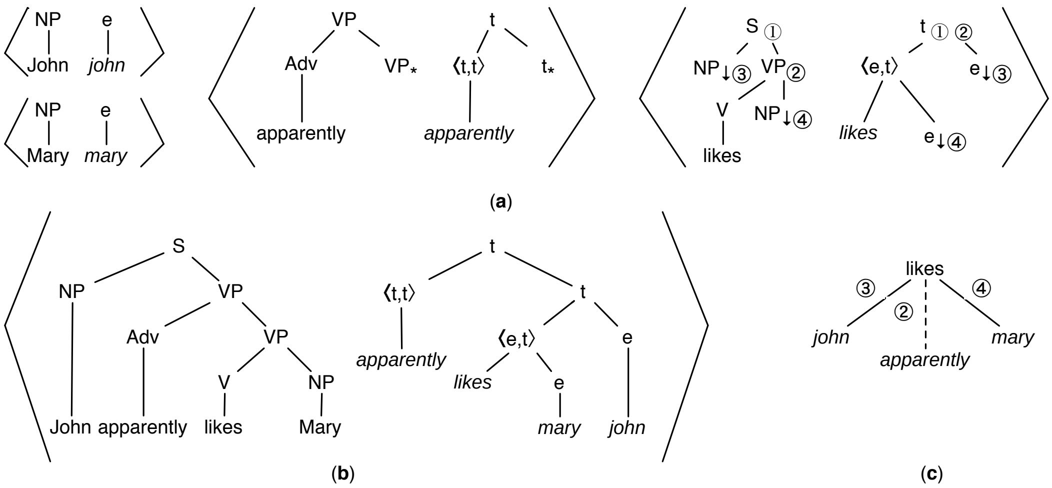

Figure 5: An English syntax/semantics STAG fragment (a), derived tree pair (b), and derivation tree (c), for *John apparently likes Mary.* (Reproduced from (Nesson and Shieber, 2006))

semantic representation can be read off the derived tree by treating the leftmost child of a node as a functor and its siblings as its arguments. Our sample sentences thus results in the semantic representation $apparently(likes(john, mary))$."

In the rest of the paper, we assume that the syntactic and semantic analyses of "simple" sentences (i.e. clauses, without discourse connective) are simultaneously generated by an STAG grammar, and we use the following symbols: T_i represents the syntactic analysis of sentence S_i (a tree rooted S), F_i its semantic analysis (a tree rooted t), τ_i its derivation tree.

4 D-STAG data structures

In this section, we present D-STAG data structures, namely the D-STAG grammar made of pairs of TAG discourse trees, the syntactic and semantic derived trees illustrated by the cases of discourses (2) and (3), and the derivation trees, which are compatible with SDRT discourse structures. In the next section, we discuss D-STAG processing, namely how to use the D-STAG grammar so as to obtain derivation trees, and thereby syntactic and semantic derived trees. As a consequence, this section about data structures does not touch on discourse ambiguities, e.g. the fact that a discourse connective can convey several discourse relations (which is the case of ε, for example). For discourse (2) of the form S_1 *because* S_2. *Next* S_3. we assume that *because* and *next* convey respectively Explanation and Narration. For discourse (3) of the form S_1 *because* S_2. ε S_3. we assume that *because* and ε convey respectively [Explanation] and Comment (see Section 2 for the difference between Explanation and [Explanation]).

In D-STAG, coordinating and subordinating relations are introduced respectively through initial and auxiliary trees. In TAG, initial trees introduce the predicate-argument dependencies by substitution, while auxiliary trees introduce recursion and allow elementary trees to be modified by adjuncts through adjunction. D-STAG follows these principles in that coordinating relations anchor initial trees which introduce their arguments (Nuclei) by substitution, while subordinating relations anchor auxiliary trees in which the foot node (head) corresponds to the Nucleus while the Satellite (modifier) is introduced by substitution.

A pair of D-STAG elementary trees consists of an elementary tree anchored by a discourse connective[5] linked to an elementary tree anchored by the discourse relation conveyed by the connective (assuming – as we do in this section – that the discourse relation conveyed by the connective is determined). A pair of trees links two initial trees if the discourse relation is coordinating, while it links two auxiliary trees if the discourse relation is subordinating. Any elementary tree includes two frontier nodes, which correspond to the arguments of the anchor.

Figure 6 contains a D-STAG tree pair named α*next*-Narration[6]. It pairs the initial tree for *next*

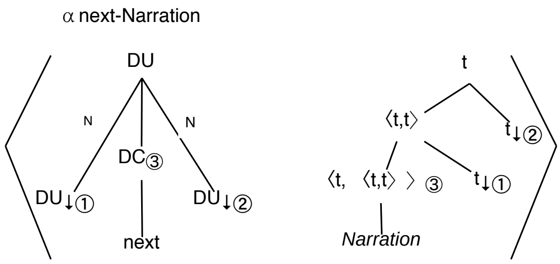

Figure 6: α*next*-Narration

with the initial tree for the coordination relation Narration conveyed by *next*. In the syntactic discourse tree, the nonterminal symbols DC and DU are used respectively for discourse connectives and discourse units[7]. A discourse unit can be simple or complex. When simple, it is the syntactic analysis of a "simple" sentence (clause, without discourse connective), i.e. a tree rooted S which is introduced at the discourse level by the left tree of α*S-to-D* presented below in Figure 11. When complex, it is recursively the syntactic discourse analysis of a complex sentence including discourse connective(s) or of a multisentential text. In order to help the reader who is familiar with RST, we added the labels N (Nucleus) and S (Satellite) on the edges pointing to the nodes labeled DU. In the semantic tree, the anchor *Narration* is of type $\langle t, \langle t, t\rangle\rangle$. It is a functor which takes its sibling as argument leading to a new functor which takes its sibling as argument (as explained in Section 3 about STAG). Its semantic representation is $\lambda pq.\phi_{Narration(^\wedge p,^\wedge q)}$ in which p and q range over type t, and $\phi_{Narration(^\wedge p,^\wedge q)}$ are "the special semantic constraints pertinent to the particular discourse relation $Narration(^\wedge p,^\wedge q)$" (Asher & Lascarides, 2003, p.156). For example, $\phi_{Narration(^\wedge p,^\wedge q)}$ would allow us to state – in a simplified way, omitting prestates and poststates of events – that the main event of p occurs before the main event of q, while $\phi_{Explanation(^\wedge p,^\wedge q))}$ would allow us to state that the main event of q occurs before and causes the main event of p. In fact, as Narration is veridical (Asher & Lascarides, 2003, p.156), the semantic representation of *Narration* should be $\lambda pq.\phi_{Narration(^\wedge p,^\wedge q)} \wedge p \wedge q$. However, we use a simplified expression for the sake of clarification.

Figure 7 contains $\beta\varepsilon$-Comment which pairs auxiliary trees. The semantic representation of *Comment* is $\lambda pq.\phi_{Comment(^\wedge q,^\wedge p)}$ in which p and q range over type t.

Figure 8 contains β*because*-Explanation which pairs the auxiliary tree anchored by *because* with either an auxiliary tree anchored by *Explanation* for the cases where *because* introduces a

[5]The discourse tree of a connective can be quite different from its syntactic tree. For example, the discourse tree for an adverbial such as *next* has two arguments while its syntactic tree has a single argument. Therefore, a D-STAG parser should include a *Tree Extractor* and a *Tree Mapping* as in a parser for D-LTAG (Webber, 2004).

[6]The prefix α or β indicates if the paired trees are initial (α) or auxiliary (β).

[7]The use of indice ③ will be illustrated with discourses (5) presented in Section 6 below.

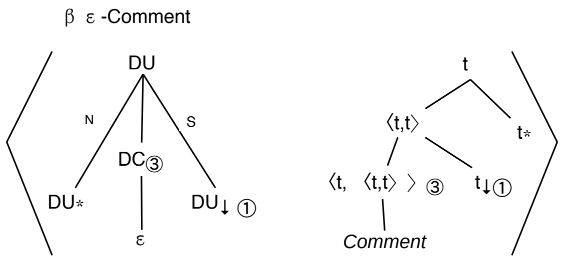

Figure 7: $\beta\varepsilon$-Comment

modifier, or an auxiliary tree anchored by *[Explanation]* for the cases where *because* is used to form a complex constituent (see Section 2.).

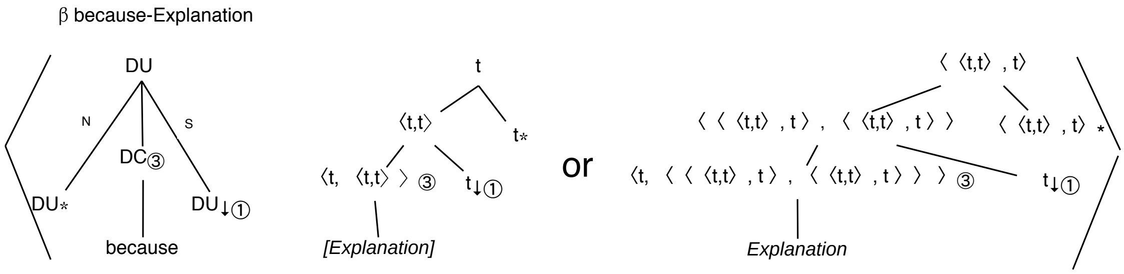

Figure 8: β*because*-Explanation

The semantic auxiliary tree anchored by *[Explanation]* follows the pattern of the semantic tree anchored by *Comment*; the semantic representation of *[Explanation]* is $\lambda pq.\phi_{Explanation(^\wedge q,^\wedge p)}$ with $p, q : t$. Figure 9 shows the derived tree pair (a) and (b) for (3). The semantic representation read off (b) is $\phi_{Comment(\phi_{Explanation(^\wedge F_1,^\wedge F_2)},^\wedge F_3)}$, which is the expected result.

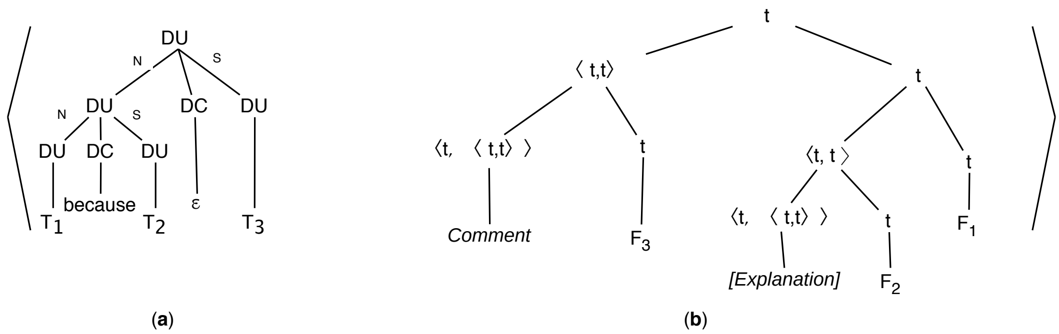

Figure 9: D-STAG derived tree pair (a)-(b) for discourse (3)

On the other hand, the tree anchored by *Explanation* is more complex:

- first, it involves a type raising for the foot node (hence also for the root node): it is not of type t but of type $\langle\langle t, t\rangle, t\rangle$.
- second, the semantic representation of *Explanation* is:

$$\lambda pAB.A(\lambda q.\phi_{Explanation(^\wedge q,^\wedge p)} \wedge B(q)) \text{ with } A, B : \langle\langle t,t\rangle, t\rangle, \text{and } p, q : t$$

Why do we need such a complex semantic tree? Let us come back to the discourse examples (2) and (3). We want to compute the following semantic representations:

- for (2), $\phi_{Explanation(^\wedge F_1,^\wedge F_2)} \wedge \phi_{Narration(^\wedge F_1,^\wedge F_3)}$
- for (3), $\phi_{Comment(\phi_{Explanation(^\wedge F_1,^\wedge F_2)},^\wedge F_3)}$

In words, we want to compute that the causal relation expressed by *because* is under the scope of the Comment discourse relation in (3), but is not under the scope of the Narration discourse relation in (2): only the Nucleus of this causal relation is under the scope of Narration. In technical terms, the semantic tree anchored by *Explanation* is designed so as to give the accurate predicate-argument dependencies for examples such as (2), as does the Nuclearity Principle for such an example (Section 2). Figure 10 shows the derived tree pair (c) and (d) for (2). In (d), the

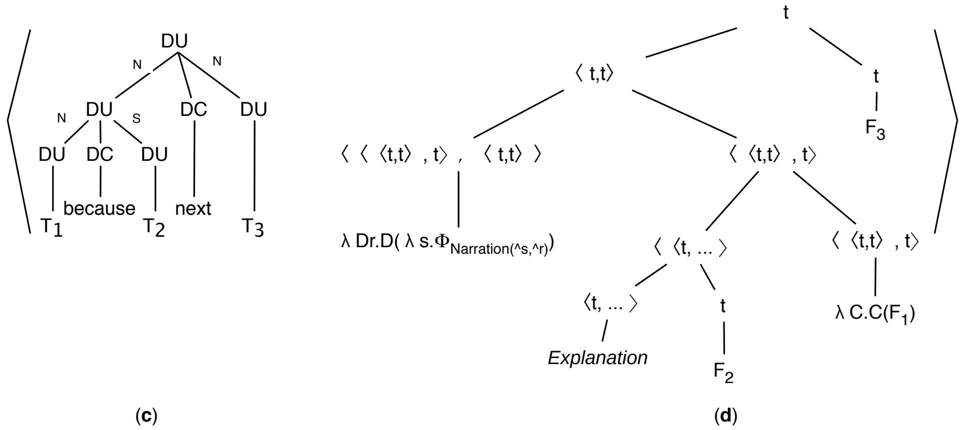

Figure 10: D-STAG derived tree pair (c)-(d) for discourse (2)

functor *Explanation* , with F_2 as value for the substitution node, and $\lambda C.C(F_1)$ with $C :< t, t >$ (type raising) as value for the foot node, leads to the term Υ of type $\langle\langle t,t\rangle, t\rangle$.

$$\Upsilon = \lambda B.\phi_{Explanation(^\wedge F_1,^\wedge F_2)} \wedge B(F_1)$$

This term is the sibling of the functor *Narration* which requires an argument of type t (see Figure 6). Therefore, for (2), the functor *Narration* needs to be applied a rule of argument raising so as to take into account that its argument is not of type t but of type $\langle\langle t,t\rangle, t\rangle$: its semantic representation becomes $\lambda Dr.D(\lambda s.\phi_{Narration(^\wedge s,^\wedge r)})$ with $D : \langle\langle t,t\rangle, t\rangle$ and $r, s : t$. This functor taking Υ as argument leads to: $\lambda r.\phi_{Explanation(^\wedge F_1,^\wedge F_2)} \wedge \phi_{Narration(^\wedge F_1,^\wedge r)}$. This functor taking F_3 as argument leads to $\phi_{Explanation(^\wedge F_1,^\wedge F_2)} \wedge \phi_{Narration(^\wedge F_1,^\wedge F_3)}$, which is the expected result.

Figure 11 shows a special tree pair, α*S-to-D*, which is designed to plug the STAG syntactic and semantic analyses of a sentence generated by an STAG grammar into a D-STAG grammar.

Figure 12 contains the derivation tree (e) for (2). (e) can be simplified as (e') by using the symbol τ_i' which stands for α*S-to-D* in which τ_i is substituted at link ①. Figure 12 also contains the derivation tree (f') for (3).

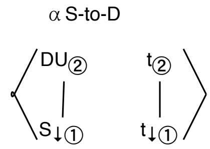

Figure 11: *αS-to-D*

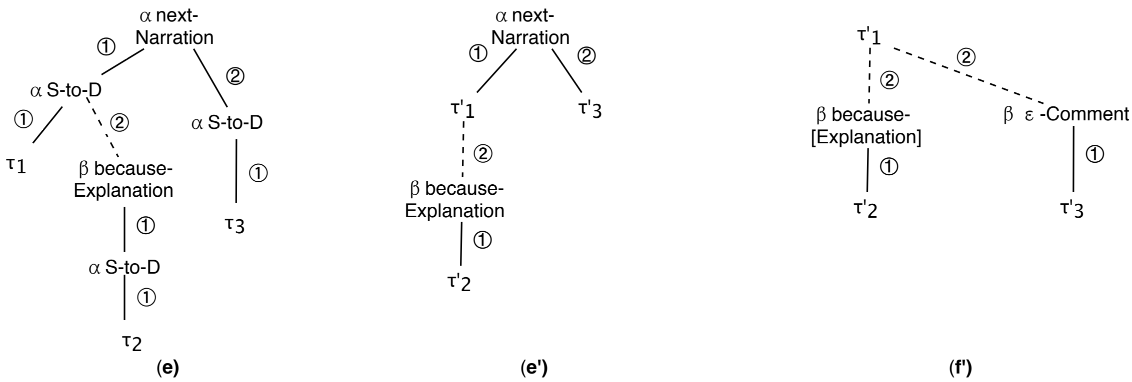

Figure 12: D-STAG derivations tree (e) or (e') for (2) and (f') for (3)

D-STAG derivation trees deterministically convert into SDRT graphs by recursively applying the patterns given in Figure 13. The main difference between D-STAG derivation trees and SDRT graphs is that SDRT uses labels tagging propositions ("sentence nodes") and "scope nodes" (Section 2). The use of labels tagging propositions is theoretically justified in (Asher & Lascarides, 2003, p.136) by the fact that a given sequence, e.g. *John fell. Max pushed it.* can occur twice in the same discourse, once with the interpretation that the falling happened after the pushing, the other time with the interpretation that the falling happened before the pushing. This kind of phenomenon should be rare in corpora, so we have left labels tagging propositions aside in D-STAG for the time being. Scope nodes do not seem necessary either. The dashed arrows in the patterns of Figure 13 link a node in a derivation tree and a scope node in the corresponding SDRT graph. They indicate to which node in the derivation tree a node attached to a scope node in the SDRT graph must be attached. With these patterns, (e') converts into the SDRT graph for (2) given in Figure 1 and (f') converts into the SDRT graph for (3) given in Figure 3.

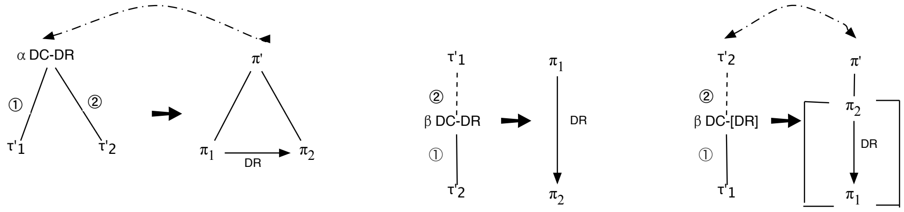

Figure 13: Conversion patterns from a D-STAG derivation tree to an SDRT graph

The fact that D-STAG derivation trees are compatible with SDRT discourse structures allows us to use the SDRT mechanisms set up for discourse ambiguities (and under-specifications), as described in the next section.

5 D-STAG processing

Since D-STAG derivation trees are compatible with SDRT discourse structures, D-STAG can take advantage of the results brought by this **discourse** theory. In particular, it can take advantage of the solutions proposed to deal with discourse ambiguities (and discourse under-specifications). For example, a parser for a D-STAG grammar has to take into account that a given discourse connective can be semantically ambiguous, i.e. can convey several discourse relations. This is the case for the empty connective ε, which "conveys" Explanation, Elaboration, Comment and Narration, among others. In D-STAG, such an ambiguous discourse connective anchors as many syntactic elementary trees as it has interpretations, bringing a D-STAG tree pair for each interpretation, e.g. $\beta\varepsilon$-Explanation, $\beta\varepsilon$-Elaboration, $\beta\varepsilon$-Comment, $\alpha\varepsilon$-Narration. The choice of the right interpretation for an ambiguous connective depends on (extra)-linguistic and rhetorical considerations. For example, the discourse in (4) of the form S_1 *because* S_2. ε S_3. should receive the derivation tree shown in Figure 14 in which ε "conveys" the subordinating relation Elaboration, S_3 elaborating S_1. Figure 14 also shows the SDRT graph for (4), which can be obtained from the derivation tree by applying the pattern in the middle of Figure 13.

(4) John went to the supermarket because his fridge was empty. He bought a lot of cheese.

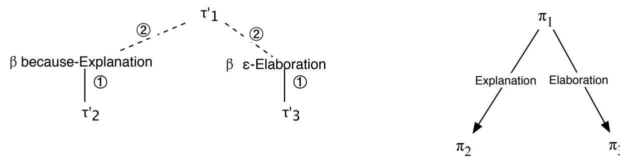

Figure 14: D-STAG derivation tree and SDRT graph for discourse (4)

The interpretation of (4) comes, among other things, from the knowledge that one goes to a supermarket to buy food. Taking into account (extra)-linguistic considerations requires a machinery such as that proposed in SDRT which relies on an incremental process based on a glue logic (Asher & Lascarides, 2003). Since D-STAG derivation trees are compatible with SDRT discourse structures, we can rely upon the SDRT incremental process e.g. to determine where and how new information should be attached to the discourse structure (derivation tree or SDRT graph) representing its left context (making use of the Right Frontier Constraint which strongly guides the attachment of new information as explained in Section 2)[8].

6 Comparison between D-STAG and D-LTAG

D-STAG is like D-LTAG - in the version presented in (Forbes-Riley *et al.*, 2006) - in that the two formalisms extend a sentential TAG syntax/semantic interface to the discourse level. Therefore, each formalism presents a model of discourse interpretation that exploits the same mechanisms used at the sentence level.

[8]Another solution for dealing with discourse ambiguities consists in calling upon probabilistic methods, as proposed in the framework of D-LTAG thanks to the Penn Discourse TreeBank (Webber, 2004). The two solutions are complementary.

The crucial difference between D-STAG and D-LTAG is that D-LTAG ignores discourse relations and their coordinating or subordinating type; it computes logical forms for discourses without building discourse structures annotated with discourse relations. As a consequence, D-LTAG has nothing in common with SDRT (nor with RST) and does not use rhetoric or pragmatic knowledge.

There exists another difference between D-STAG and D-LTAG: the syntactic discourse analyses are different because, in D-STAG, discourse connectives anchor elementary trees with two arguments, while, in D-LTAG, they can anchor elementary trees with just one argument which is *structurally* retrieved, the other one being provided *anaphorically* (Webber *et al.*, 2003; Webber, 2004). As an illustration, the D-LTAG syntactic analysis for (2) is given in Figure 15; it includes **three** DC nodes (for *because*, ε, and *next*). It is different from the D-STAG syntactic analysis for (2) given in (c) in Figure 10, which includes only **two** DC nodes.

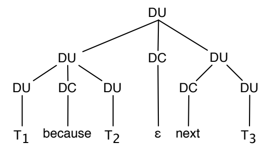

Figure 15: D-LTAG syntactic tree for discourse (2)

It is the topic of another paper to explain why we disagree with the difference between arguments structurally retrieved versus provided anaphorically. However, let us just say here that the discrepancy between discourses (2) and (3), in which *because* either introduces a modifier or is the pivot of a complex constituent, is not taken into account in D-LTAG.

Moreover, D-STAG can benefit of the adjunction operation for modification of discourse relations. This phenomenon is illustrated by the discourse in (5a) taken from (Webber *et al.*, 2003) and of the form S_1 *because, for example,* S_2. As these authors explain, the interpretation of (5a) is that John never returning what he borrows is one example of the reasons for not trusting him. We give (5a) the following semantic representation which is inspired by (Webber *et al.*, 2003) and adapted to D-STAG:

$$Exemplify(^\wedge F_2, \lambda p.\phi_{Explanation(^\wedge F_1, ^\wedge p)}) \text{ with } p : t$$

(5)a. You should not trust John because, for example, he never returns what he borrows.
b. You should not trust John only because he never returns what he borrows.

We postulate that *for example* in (5a) modifies *because*, in the same way as *only* modifies *because* in (5b)[9]. Therefore, in D-STAG, this adverbial anchors a syntactic auxiliary tree whose foot node is labeled DC and which is paired with a semantic auxiliary tree whose foot node is a discourse relation of type $\langle t, \langle t, t \rangle \rangle$. The semantic representation of the functor *for example* is:

$$\lambda Rpq.Exemplify(\wedge p, \lambda r.R(^\wedge q, ^\wedge r)) \text{ with } R : \langle t, \langle t, t \rangle \rangle, \text{and } p, q, r : t$$

[9]On the other hand, *for example* in a discourse such as *John loves cheese. For example, he loves camembert.* could be considered as a discourse connective.

The resulting tree pair, named $\beta$for-ex, is shown in Figure 16. When parsing (5a), $\beta$for-ex adjoins at link ③ in βbecause-[Explanation] given in Figure 8[10]. The derivation tree for (5a) is shown in Figure 16. The reader can check that the semantic representation read off the semantic derived tree derived from the derivation tree is $Exemplify(^\wedge F_2, \lambda p.\phi_{Explanation(^\wedge F_1, ^\wedge p)})$.

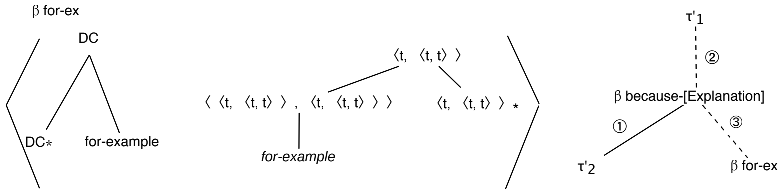

Figure 16: Derivation tree for discourse (5a)

On the other hand, in D-LTAG, *for example* in (5) is considered as a discourse connective, and the computations made to obtain the semantic representation of this discourse are heavy, see (Forbes-Riley *et al.*, 2006). More generally, modification of discourse connectives/relations is a phenomenon which is not contemplated in D-LTAG, although the adjunction operation allows us to handle this (common) phenomenon easily.

7 Conclusion

D-STAG is designed to build a complete integrated text understanding system, which is based on SDRT for dealing with discourse ambiguities (and underspecifications).

On theoretical grounds, future research will concern e.g. sentences with a relative clause such as (1b) for which we want to compute a semantic representation equivalent to that of (1a). This is made possible by the fact that D-STAG exploits the same mechanisms at the sentence and discourse levels. It should require interleaving the S-TAG sentential grammar and the D-STAG discourse grammar, while they have been used in a pipeline architecture in this paper. Future research will also concern discourses in which arguments of discourse relations come from discontinuous text spans, which happens with the relation Attribution when one of its arguments is embedded in the other one (which is therefore discontinuous). We think that the adjunction operation will be of great help for these cases. Finally, future research will concern "discourse verbs" (Danlos, 2006b; Danlos, 2007a). A discourse verb is a verb such as *prove* in discourse (3). In this paper, (3) has been analyzed with the discourse relation Comment linking the last sentence to its left context. This is not really satisfactory since, firstly, Comment is a "poor" discourse relation, secondly, this analysis does not reflect that Fred being upset because his wife is abroad is evidence of his love for her. The notion of discourse verb will allow us to propose a more accurate analysis for (3).

On practical grounds, the issue of implementing D-STAG is currently being discussed within the ALPAGE team (http://alpage.inria.fr), in which there already exists a number of parsers for TAG grammars (for French).

[10]In (5a), *because* conveyed [Explanation] and not Explanation because, in informal terms, the Nuclearity Principle is not at stake.

References

ASHER N. (1993). *Reference to Abstract Objects in Discourse*. Dordrecht: Kluwer.

ASHER N. & LASCARIDES A. (2003). *Logics of Conversation*. Cambridge: Cambridge University Press.

ASHER N. & VIEU L. (2005). Subordinating and coordinating discourse relations. *Lingua*, **115**(4), 591–610.

BOS J. (1995). Predicate logic unplugged. In P. DEKKER & M. STOKHOF, Eds., *Proceedings of the 10th Amsterdam Colloquium*, p. 133–142.

COPESTAKE A., FLICKINGER D., SAG I. A. & POLLARD C. (1999). Minimal Recursion Semantics. An Introduction. Manuscript, Stanford University.

DANLOS L. (1998). G-TAG : un formalisme lexicalisé pour la génération de textes inspiré de TAG. *Revue TAL*, **39**(2).

DANLOS L. (2001). A lexicalized formalism for text generation inspired from TAG. In A. ABEILLÉ & O. RAMBOW, Eds., *TAG Grammar*. CSLI.

DANLOS L. (2006a). Capacité générative forte de RST, SDRT et des DAG de dépendances pour le discours. *Revue TAL*, **47**(2).

DANLOS L. (2006b). Discourse verbs and discourse periphrastic links. In *Proceedings of Constraints in Discourse'06*, Maynooth, Ireland.

DANLOS L. (2007a). Integrating discourse relations into lexical semantics. In *Proceedings of GL'07*, Paris, France.

DANLOS L. (2007b). Strong generative capacity of RST, SDRT and discourse dependency DAGs. In A. BENZ & P. KÜHNLEIN, Eds., *Constraints in Discourse*. Benjamins.

FORBES-RILEY K., WEBBER B. & JOSHI A. (2006). Computing discourse semantics: The predicate-argument semantics of discourse connectives in D-LTAG. *Journal of Semantics*, **23**(1).

JOSHI A. (1985). Tree-adjoining grammars. In D. DOWTY, L. KARTTUNEN & A. ZWICKY, Eds., *Natural language parsing*, p. 206–250. Cambridge University Press.

JOSHI A., KALLMEYER L. & ROMERO M. (2007). Flexible composition in LTAG: Quantifier scope and inverse linking. In R. MUSKENS & H. BUNT, Eds., *Computing Meaning Volume 3*. Kluwer.

KALLMEYER L. (2002). Using an enriched tag derivation structure as basis for semantics. In *Proceedings of the TAG+6 Workshop*, p. 101–110, Venice.

MANN W. & THOMPSON S. (1988). Rhetorical structure theory : Toward a functional theory of text organization. *Text*, **8**(3), 243–281.

MARCU D. (2000). *The Theory and Practice of Discourse Parsing and Summarization*. The MIT Press.

NESSON R. & SHIEBER S. (2006). Simpler TAG semantics through synchronization. In *Proceedings of the 11th Conference on Formal Grammars*, Malaga.

NESSON R. & SHIEBER S. (2007). Extraction phenomena in synchronous TAG syntax and semantics. In *Proceedings of the Workshop on Syntax and Structure in Statistical Translation*, Rochester.

POLANYI L. (1988). A formal model of the structure of discourse. *Journal of Pragmatics*, **12**, 601–638.

SHIEBER S. (1994). Restricting the weak-generative capacity of synchronous tree-adjoining grammars. *Computational Intelligence*, **10**(4), 371–385.

SHIEBER S. & SCHABES Y. (1990). Synchronous tree-adjoining grammars. In *Proceedings of the 13th International Conference on Computational Linguistics*, volume 3, p. 253–258, Helsinki.

TABOADA M. & MANN W. (2006). Rhetorical structure theory: Looking back and moving ahead. *Discourse Studies*, **8**(3), 423–459.

WEBBER B. (2004). DTAG: extending lexicalized TAG to discourse. *Cognitive Science*, **28**(5), 751–779.

WEBBER B., JOSHI A., STONE M. & KNOTT A. (2003). Anaphora and discourse structure. *Computational Linguistics*, **29**(4), 545–587.