Interfacing Sentential and Discourse TAG-based Grammars
Laurence Danlos, Aleksandre Maskharashvili, Sylvain Pogodalla

To cite this version:
Laurence Danlos, Aleksandre Maskharashvili, Sylvain Pogodalla. Interfacing Sentential and Discourse TAG-based Grammars. Proceedings of the 12th International Workshop on Tree Adjoining Grammars and Related Formalisms (TAG+12), Jun 2016, Düsseldorf, Germany. hal-01328697v3

HAL Id: hal-01328697
https://inria.hal.science/hal-01328697v3
Submitted on 10 Jun 2016

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L’archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.
Interfacing Sentential and Discourse TAG-based Grammars

Laurence Danlos
Université Paris Diderot
ALPAGE
INRIA Paris–Rocquencourt
Institut Universitaire de France
Paris, F-75005, France
laurence.danlos@inria.fr

Aleksandre Maskharashvili
Sylvain Pogodalla
INRIA, Villers-lès-Nancy, F-54600, France
Université de Lorraine, CNRS
LORIA, UMR 7503
Vandœuvre-lès-Nancy, F-54500, France
aleksandre.maskharashvili@inria.fr
sylvain.pogodalla@inria.fr

Abstract
Tree-Adjoining Grammars (TAG) have been used both for syntactic parsing, with sentential grammars, and for discourse parsing, with discourse grammars. But the modeling of discourse connectives (coordinate conjunctions, subordinate conjunctions, adverbs, etc.) in TAG-based formalisms for discourse differ from their modeling in sentential grammars. Because of this mismatch, an intermediate, not TAG-related, processing step is required between the sentential and the discourse processes, both in parsing and in generation. We present a method to smoothly interface sentential and discourse TAG grammars, without using such an intermediate processing step. This method is based on Abstract Categorial Grammars (ACG) and relies on the modularity of the latter. It also provides the possibility, as in D-STAG, to build discourse structures that are direct acyclic graphs (DAG) and not only trees. All the examples may be run and tested with the appropriate software.

1 Introduction
It is usually assumed that the internal structure of a text, typically characterized by discourse or rhetorical relations, plays an important role in its overall interpretation. Building this structure may resort to different techniques such as segmenting the discourse into elementary discourse units and then relating them with appropriate relations (Marcu, 2000; Soricut and Marcu, 2003). Other techniques use discourse grammars, and a particular trend relies on tree grammars (Polanyi and van den Berg, 1996; Gardent, 1997; Schilder, 1997). This trend has been further developed by integrating the modeling of both clausal syntax and semantics, and discourse syntax and semantics within the framework of Tree-Adjoining Grammar (TAG, Joshi et al. (1975); Joshi and Schabes (1997)). This gave rise to the TAG for Discourse (D-LTAG) formalism (Webber and Joshi, 1998; Forbes et al., 2003; Webber, 2004; Forbes-Riley et al., 2006), and to the Discourse Synchronous TAG (D-STAG) formalism (Danlos, 2009; Danlos, 2011). The latter derives semantic interpretation using Synchronous Tree-Adjoining Grammars (STAG, Shieber and Schabes (1990); Nesson and Shieber (2006); Shieber (2006)).

While one may think that using similar frameworks for both levels should help to interface them, it is not as smooth as one can expect. Indeed, a shared feature of D-LTAG and D-STAG is that grammatical parsing and discourse parsing are performed at two different stages. Moreover, the result of the first stage requires additional, not TAG-related, processing before being able to enter the second stage. This intermediary step consists in discourse relation extraction in D-LTAG and in discourse normalization in D-STAG.

The reason for this intermediary step relates to the mismatch between the syntactic properties and the discourse properties of discourse markers. For instance, at the syntactical level, sentences as in (1) are well-formed.

(1) a. Then, John went to Paris.
   b. John then went to Paris.

The discourse marker then, an adverb, is considered as a modifier, either of the whole clause in (1a) or of the verb phrase (1b). In TAG, they are represented as auxiliary trees with S or VP root nodes (XTAG Research Group, 2001; Abeillé, 2002). Using the elementary trees of Figure 1, Figure 2 (Figure 4, resp.) shows the TAG analysis of (1a) (of (1b), resp.).
At the discourse level, it is difficult to interpret these sentences without referring to preceding sentences. The discourse relation (e.g., Narration) has two arguments: the discourse unit consisting of the clause in which the discourse cue appears (the host clause), and some other discourse unit (it can be a complex one). D-LTAG and D-STAG propose different models of such adverbials, in particular in the way the first argument is provided. But in both accounts, adverbials are fronted (see Figure 6(c) and Figure 9(a)). Hence sentences with medial adverbials such as (1b) are excluded without the intermediary step of discourse relation extraction.

A similar mismatch occurs with subordinate conjunctions. In a typical TAG analysis, they are modeled with auxiliary trees because they modify the matrix clause and are not part of its predicate-argument structure.\(^1\) In D-LTAG, however, they are modeled with initial trees with two substitution sites (see Figure 6(a)) for the two discourse units they are predicating over.

So the question of relating the syntactic modeling and the discourse modeling arises. In particular, we wish to avoid this relation to rely on some intermediary step. Indeed, the latter has several drawbacks. First, it complicates the modeling of connectives that are ambiguous in their syntactic and discourse use, and prevents us from using standard grammar inference and disambiguation techniques. Second, while most of the syntax-semantics interfaces, in particular in TAG, aim at satisfying a compositional assumption (Gardent and Kallmeyer, 2003; Pogodalla, 2004; Kallmeyer and Romero, 2008; Nesson and Shieber, 2006), the syntax-discourse interface seems to escape it. Third, a better integration of the sentential and of the discourse components also seems an interest-

\(^1\)It is not always the case, though. Bernard and Danlos (2016) propose different elementary trees, depending on the syntactical, semantic, and discourse properties of the conjunction.
ing feature if we want to better describe the interaction between discourse connectives and propositional attitude predicates (Danlos, 2013; Bernard and Danlos, 2016).

Finally, when generation instead of parsing is at stake, this architecture also prevents the reversibility of the grammars and requires ad-hoc post-processing. G-TAG, a TAG-based formalism dedicated to generation that includes elements of a discourse grammar, had this requirement (Danlos, 1998; Meunier, 1997; Danlos, 2000).

In this article, we describe how to interface a sentential and a discourse TAG-based grammar. We show how to link such two grammars and their proposed modelings of discourse connectives, overcoming the above mentioned issue. We use an encoding of TAG into Abstract Categorial Grammar (ACG, de Groote (2001)), a grammatical framework based on the simply typed λ-calculus. As we aim at reusing previous works such as existing TAG sentential grammars as well as discourse analysis, our approach relies on two key features of ACG: the ACG account of the TAG operations and the ACG-based syntax-semantics interface for TAG (Pogodalla, 2004; Pogodalla, 2009) on the one hand; and the modular ACG composition, in order to smoothly integrate the syntactical and discourse behavior of adverbial connectives without using a two-step analysis on the other hand. Note, however, that the operations we use in the ACG composition are not available as TAG operations. While the encoding of TAG into ACG is standard (de Groote, 2002; Pogodalla, 2009), our contribution is to use the interpreting device of ACG to relate the ACG encoding of a TAG sentential grammar and the ACG encoding of a TAG discourse grammar. The example grammars we use may be run and tested2 on the ACG development software.

2 TAG Based Discourse Grammars

As TAG grammars, D-LTAG and D-STAG do not differ from any other TAG grammar: they define elementary trees that can be combined using the operations of substitution and adjunction. However, if some elementary trees are anchored by lexical items (the discourse markers) as in sentential grammars, the others are anchored by clauses resulting from the syntactic analysis. Contrary to sentential grammars that contain a lot of different elementary tree families, discourse grammars have a small set of such families. In this section, we focus on these elementary trees, anchored by discourse markers. We show how the structure of these trees influences the interaction between the sentential and the discourse grammars, and why this interaction calls for an intermediary processing step. For an in-depth presentation of these formalisms, we refer the reader to (Webber and Joshi, 1998; Forbes et al., 2003; Webber, 2004; Forbes-Riley et al., 2006) for D-LTAG and to (Danlos, 2009; Danlos, 2011) for D-STAG.

Figure 6: D-LTAG elementary tree schemes

D-LTAG D-LTAG proposes three main families of elementary trees that capture different insights on discourse structures. Trees for subordinate conjunctions are modeled using initial trees with two substitution nodes for each of the arguments as Figure 6(a) shows. This reflects the predicate-argument structure of these connectives at the discourse level. But this contrasts with the syntactic account of these connectives: because they are outside the domain of locality of the verbs to which they can adjoin (at S or VP nodes), they typically are modeled using auxiliary trees (see Figure 7).

The second family of connectives is used to extend or to elaborate on clauses with auxiliary trees anchored by coordinate conjunctions (or by the empty connective). The first argument of the connective corresponds to the discourse unit the tree is anchored by. These trees are modeled using auxiliary trees as Figure 6(b) shows. The third family of connectives is used to elaborate on clauses with auxiliary trees anchored by coordinate conjunctions (or by the empty connective). The first argument of the connective corresponds to the discourse unit the tree is anchored by.
adjoined to, and the second, the extending clause, corresponds to the clause that is substituted at the substitution node, as Figure 6(b) shows.

The third family also consists of auxiliary trees. But the latter are associated with a single clause as Figure 6(c) shows. The second argument comes from the anaphoric interpretation of the connectives anchoring such trees.

The two-stage process for parsing discourse proceeds as follows: first, each sentence gets a TAG analysis (derived and derivation trees) by a standard TAG. Then, each derivation tree is processed in order to identify the possible discourse connectives and their arguments from a syntactic point of view. The latter (one or two, depending on the connective) are added as initial trees with root Du to the discourse grammar, as well as the (discourse) elementary tree anchored by the connective. For instance, from the clausal derivation tree of Figure 8, the two arguments $\alpha^d_{s_1}$ and $\alpha^d_{s_2}$, and the connective $\beta^d_{s_3}$ are extracted. A similar extraction step takes care of the extraction of clausemedial adverbial connectives.

**D-STAG** Contrary to D-LTAG, D-STAG models all discourse connectives with auxiliary trees that are adjoined to the discourse unit they extend. The clause content that serves as second argument of the connective is substituted within this tree. Figure 9 shows some of the schemes for the elementary (auxiliary) trees of a D-STAG. The three internal Du nodes are available for adjunctions, achieving different effects on the semantic trees (following the principles of synchronous TAG, each discourse elementary tree is paired with a semantic tree). Together with a higher-order type for the semantic trees, this allows D-STAG to structurally generate DAG discourse structures.\(^4\) But as the focus of this article is on the articulation between the sentential and the discourse grammar, we do not enter the details of the discourse-semantics interface here (see Danlos et al. (2015)).

D-STAG shares with D-LTAG the requirement of a transformation from the sequence of sentence analysis with a sentential grammar into a sequence of clauses and discourse connectives in a “dis- course normal form”. The reasons are basically the same as for D-LTAG: discourse connectives need to be identified in order to anchor their associated discourse elementary trees, as they differ from their syntactic elementary trees, and clausemedial extractions need to be managed at this level as well.

**Medial Adverbial Extraction** Looking at the elementary trees of Figure 9(a) (the problem is similar for D-LTAG elementary trees), we observe that the host clause of the adverbial are substituted into the elementary tree, at the Du $\downarrow$ node. But at the sentential level, it is auxiliary trees anchored by the adverbials that adjoin into the host clause. When the adjunction occurs at the top S node, we get the same surface form in both cases. However, whenever the adjunction occurs at the VP node in the sentential grammar, this is not the case anymore: the adverbial is not fronted, and the discourse grammar cannot account for this position. An intermediary form, such as the discourse normal form in D-STAG, or the tree extraction in D-LTAG is then required. In order to get rid of this intermediary step, we should be

---

\(^4\)Such structures are not easily available with D-LTAG, and this was a motivation to introduce D-STAG.
able to describe an operation that simultaneously substitutes a clause within the elementary tree of the discourse connective, and adjoins the auxiliary tree on the VP node. Figure 10(a) describes such an operation. The dotted lines would represent a dominance constraint that the tree to be substituted should satisfy. It is also natural then to use the same approach for fronted discourse adverbials, as in Figure 10(b).

Because the adverb has to adjoin within the tree that is being substituted, describing such an operation seems not to be possible in TAG nor in multicomponent TAG (at least in a single step). It would be possible with D-Tree Substitution Grammars (Rambow et al., 2001), but then the derivation trees would be different, the synchronous syntax-semantics interface would have to be redefined, and the reversibility properties (for generation) would have to be stated. We instead use an encoding with ACG, where these properties naturally follow the standard encoding of TAG into ACG.

Figure 9: D-STAG elementary trees

Figure 10: Auxiliary trees for discourse connectives

3 Abstract Categorial Grammars

ACG derives from type-theoretic grammars. Rather than a grammatical formalism on its own, it provides a framework in which several grammatical formalisms may be encoded (de Groote and Pogodalla, 2004), in particular TAG (de Groote, 2002). The definition of an ACG is based on type-theory, λ-calculus, and linear logic. In particular, ACG generates languages of linear λ-terms, which generalize both string and tree languages.

As key feature, ACG provides the user with a direct control over the parse structures of the grammar, the abstract language. Such structures are later on interpreted by a morphism, the lexicon, to get the concrete object language. We use the standard notations of the typed λ-calculus.

Definition (Types). Let A be a set of atomic types. The set $\mathcal{T}(A)$ of implicative types built upon $A$ is defined with the following grammar:

$$
\mathcal{T}(A) ::= A | \mathcal{T}(A) \rightarrow \mathcal{T}(A)
$$

Definition (Higher-Order Signatures). A higher-order signature $\Sigma$ is a triple $\Sigma = \langle A, C, \tau \rangle$ where:

\footnote{We use the linear arrow $\rightarrow$ of linear logic (Girard, 1987) for the implication.}
A is a finite set of atomic types;

• C is a finite set of constants;

• \( \tau : C \rightarrow \mathcal{T}(A) \) is a function assigning types to constants.

We note \( \Lambda(\Sigma) \) the set of typed terms build on \( \Sigma \). For \( t \in \Lambda(\Sigma) \) and \( \alpha \in \mathcal{T}(A) \), we denote that \( t \) has type \( \alpha \) by \( t :\Sigma_{\alpha} \alpha \) (possibly omitting the subscript).

**Definition** (Lexicon). Let \( \Sigma_{1} = (A_{1}, C_{1}, \tau_{1}) \) and \( \Sigma_{2} = (A_{2}, C_{2}, \tau_{2}) \) be two higher-order signatures. A lexicon \( \mathcal{L} = (F, G) \) from \( \Sigma_{1} \) to \( \Sigma_{2} \) is such that:

- \( F : A_{1} \rightarrow \mathcal{T}(A_{2}) \). We also note \( F : \mathcal{T}(A_{1}) \rightarrow \mathcal{T}(A_{2}) \) its homomorphic extension;\(^6\)

- \( G : C_{1} \rightarrow \Lambda(\Sigma_{2}) \). We also note \( G : \Lambda(\Sigma_{1}) \rightarrow \Lambda(\Sigma_{2}) \) its homomorphic extension;\(^7\)

- \( F \) and \( G \) are such that for all \( c \in C_{1}, G(c) \) is of type \( F(\tau_{1}(c)) \) (i.e., \( G(c) :\Sigma_{2} F(\tau_{1}(c)) \)).

We also use \( \mathcal{L} \) instead of \( F \) or \( G \).

**Definition** (Abstract Categorial Grammar and vocabulary). An **abstract categorial grammar** is a quadruple \( \mathcal{G} = (\Sigma_{1}, \Sigma_{2}, \mathcal{L}, \mathcal{S}) \) where:

- \( \Sigma_{1} = (A_{1}, C_{1}, \tau_{1}) \) and \( \Sigma_{2} = (A_{2}, C_{2}, \tau_{2}) \) are two higher-order signatures. \( \Sigma_{1} \) (resp. \( \Sigma_{2} \)) is called the **abstract vocabulary** (resp. the **object vocabulary**) and \( \Lambda(\Sigma_{1}) \) (resp. \( \Lambda(\Sigma_{2}) \)) is the set of **abstract terms** (resp. the set of **object terms**).

- \( \mathcal{L} : \Sigma_{1} \rightarrow \Sigma_{2} \) is a lexicon.

- \( \mathcal{S} \in \mathcal{T}(A_{1}) \) is the **distinguished type** of the grammar.

Given an ACG \( \mathcal{G}_{\text{name}} = (\Sigma_{1}, \Sigma_{2}, \mathcal{L}_{\text{name}}, \mathcal{S}) \), we use the following notational variants for the interpretation \( \beta \) (resp. \( \eta \)) of the type \( \alpha \) (resp. of the term \( t \)):

\( \mathcal{G}_{\text{name}}(\alpha) = \beta \) and \( \alpha := \text{name } \beta \) (resp. \( \mathcal{G}_{\text{name}}(t) = \eta \) and \( t := \text{name } \eta \)). The subscript may be omitted if clear from the context.

**Definition** (Abstract and Object Languages). Given an ACG \( \mathcal{G} \), the **abstract language** is defined by:

\[ \mathcal{A}(\mathcal{G}) = \{ t \in \Lambda(\Sigma_{1}) \mid t :\Sigma_{1} \mathcal{S} \} \]

The **object language** is defined by:

\[ \mathcal{O}(\mathcal{G}) = \{ u \in \Lambda(\Sigma_{2}) \mid \exists t \in \mathcal{A}(\mathcal{G}) \text{ s.t. } u = \mathcal{T}(t) \} \]

The process of recovering an abstract structure from an object term \( o \) is called **ACG parsing** and consists in finding the inverse image of \( \{ o \} \) under the lexicon (lexicon inversion). In this perspective, derivation trees of TAG are represented as terms of an abstract language, while derived trees (and yields) are represented by terms of some object languages. It is an object language of trees in the derived tree case and an object language of strings in the yield case. The class of **second-order ACG** is polynomially parsable with the usual complexity bounds \( O(n^{3}) \) for ACG encoding CFG, \( O(n^{6}) \) for ACG encoding TAG, Kanazawa (2008)).

The lexicon, i.e., the way structures are interpreted, plays a crucial role in our proposal in two different ways. First, two interpretations may share the same abstract vocabulary, hence mapping a single structure into two different ones, typically a surface form and a semantic form. This composition, illustrated for instance by \( \mathcal{G}_{\text{derived}} \) and \( \mathcal{G}_{\text{TAG sem.}} \), sharing the \( \Sigma_{\text{TAG sem.}} \) vocabulary in Figure 11, allows for the **semantic interpretation of derivation trees**. Second, the result of a first interpretation can itself be interpreted by a second lexicon when the object vocabulary of the first interpretation is the abstract vocabulary of the second one. This composition, illustrated for instance by the \( \mathcal{G}_{\text{yield}} \circ \mathcal{G}_{\text{derived}} \) composition in Figure 11, allows for modularity and **partial specification of derivations**. This is how we relate the discourse derivation trees to the clausal derivation trees in \( \mathcal{G}_{\text{disc.}} \).
4 Examples

4.1 TAG as ACG

We present the TAG and D-STAG encoding using examples. This encoding follows (de Groote, 2001; de Groote, 2002; Pogodalla, 2009).

In order to encode a TAG into an ACG, we use a higher-order signature $\Sigma_{TAG}$ whose atomic types include $S$, $VP$, $NP$, $S_A$, $VP_A$, . . . where the $X$ types stand for the categories $X$ of the nodes where a substitution can occur while the $X_A$ types stand for the categories $X$ of the nodes where an adjunction can occur. For each elementary tree $\tau$, there is a constant $C_{\text{ux, entry}}$ whose type is based on the adjunction and substitution sites as Table 1 shows. It additionally contains constants $I_X : X_A$ that are meant to provide a fake auxiliary tree on adjunction sites where no adjunction actually takes place in a TAG derivation. Terms built on this signature are interpreted by $\mathcal{I}_{\text{derived}}$ in the higher-order signature whose unique atomic type is $\tau$ the type of trees. In this signature, for any $X$ of arity $n$ belonging to the ranked alphabet describing the elementary trees of the TAG, we have a constant $X_n : \tau \to \cdots \to \tau \to \tau$. Then $\mathcal{I}_{\text{yield}}$ interprets $\tau$ into $\sigma$, the type for strings, and $X_n$ as $\lambda x_1 \cdots x_n . x_1 + \cdots + x_n$. For instance, the lexicon of Table 1 allows one to interpret two terms of the type of trees. In this signature, for any $X$ of arity $n$ belonging to the ranked alphabet describing the elementary trees of the TAG, we have a constant $X_n : \tau \to \cdots \to \tau \to \tau$. Then $\mathcal{I}_{\text{yield}}$ interprets $\tau$ into $\sigma$, the type for strings, and $X_n$ as $\lambda x_1 \cdots x_n . x_1 + \cdots + x_n$. For instance, the lexicon of Table 1 allows one to interpret two terms of the type of trees. In this signature, for any $X$ of arity $n$ belonging to the ranked alphabet describing the elementary trees of the TAG, we have

\[
(2a) \quad \mathcal{I}_{\text{old}} \circ \mathcal{I}_{\text{derived}}(C_{\text{went to}} C_{\text{then}} C_{\text{then}} C_{\text{Fred}} C_{\text{Paris}}) = \text{then + Fred + went + to + Paris}
\]

\[
(2b) \quad \mathcal{I}_{\text{old}} \circ \mathcal{I}_{\text{derived}}(C_{\text{went to}} I_5 C_{\text{then}} C_{\text{Fred}} C_{\text{Paris}}) = \text{Fred + then + went + to + Paris}
\]

4.2 D-STAG as ACG

The ACG encoding of D-STAG follows the above mentioned principles to encode the derived and the derivation trees resulting from the D-STAG elementary trees of Figure 9. As a consequence, we get the same derivation trees. The main differences with (Danlos, 2009; Danlos, 2011) lie in the interpretations:

- $\mathcal{I}_{\text{disc}}$ implements the interface between the discourse grammar and the sentential grammar, avoiding the intermediate step of building a discourse normal form (or the extraction step in D-LTAG). It is central to our proposal.

- $\mathcal{I}_{\text{TAG, sem.}}^8$ implements the interpretation of the discourse structures. It slightly differs from (Danlos, 2011) in order to allow for a more unified view on the semantic types and to deal with the relative scope of quantifiers and discourse relations.

Sentence-Discourse Interface The higher-order vocabulary $\Sigma_{D-STAG}$ includes the usual atomic types to describe the sentence level ($NP$, $VP$, $VP_A$, etc.) and new atomic types to describe the discourse level: $Du$, which is the type for discourse units, and the corresponding $Du_A$ type representing adjunction sites. A typical constant introducing a discourse marker such as $d_{\text{then}}^S$ has type $DC \Rightarrow Du_A \leftarrow Du_A \leftarrow Du \leftarrow Du \leftarrow Du_A$ that reflects the auxiliary trees of D-STAG (Figure 9). For comparison, see the encoding of the $C_{\text{then}}^{VP}$ encoding an auxiliary tree adjoining at a VP node. We also use a type $T$ for full texts.

The key point to smoothly interface the sentential and the discourse grammar is to have the constant that describes a discourse marker $d_{\text{then}}$ of type $DC$ at the discourse level interpreted using the corresponding auxiliary tree $C_{\text{then}}$ at the right place, i.e., as adjoining into the host clause. So, crucially, the interpretation specifies an adjunction of the auxiliary tree into the tree that is being substituted (i.e., the argument of $Du$ type that is parameter of $d_{\text{then}}$ or, in D-STAG terms, the one plugged into the $Du \downarrow$ node of the auxiliary trees of Figure 9). This operation mimics the insertion of the auxiliary tree in Figure 10.

In order to enable this adjunction, we interpret discourse units (with $Du$ type) as missing a subordinate conjunction, a fronted adverbial, or a clause-medial adverbial. This corresponds to interpreting the atomic type $Du$ as a second-order type such as $S_A \rightarrow VP_A \rightarrow S$. We actually rather interpret $Du$ as $S_A \rightarrow (VP_A \rightarrow VP_A) \rightarrow S$ in order to account for clause-medial adverbials occurring between to other adverbs such as in John suddenly then passionately kissed her. Accord-
...ing that juxtaposing TAG derivation trees of elementary dis-
...

Table 1: Sample ACG lexicon encoding the TAG grammar of Figure 1

<table>
<thead>
<tr>
<th>Constants of $\Sigma_{TAG}$</th>
<th>Their interpretations by $\mathcal{G}_{\text{derived}}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$C_{Fred}$ : NP</td>
<td>$\gamma_{Fred} : \tau$</td>
</tr>
<tr>
<td>$C_{went}$ : $S_A \rightarrow VP_A \rightarrow$</td>
<td>$\gamma_{went} : (\tau \rightarrow \tau) \rightarrow (\tau \rightarrow \tau)$</td>
</tr>
<tr>
<td>$C^\text{then}$ : $S_A$</td>
<td>$\gamma_{\text{then}} : \tau \rightarrow \tau$</td>
</tr>
<tr>
<td>$C^\text{VP}$ : $VP_A \rightarrow VP_A$</td>
<td>$\gamma_{VP} : \lambda x. (S_2 (\text{Adv}_i \text{then}) x)$</td>
</tr>
</tbody>
</table>

Table 2: $\mathcal{G}_{\text{disc}}$ interpretation for the sentence-discourse interface

We exemplify our approach on the examples (3). In D-STAG, the associated discourse replication is as in Figure 13, and the discourse derivation trees is the one of Figure 12 where the $C_i$s correspond to the derivation trees of the bracketed discourse units of the examples. In D-STAG, the discourse derivation tree of course results from the discourse normal form $F_0$ because $F_7$ then $F_8$ that are the same for (3a) and (3b).

(3)  a. [Fred went to the supermarket]$_6$ because [his fridge is empty]$_7$. Then, [he went to the movies]$_8$.

b. [Fred went to the supermarket]$_6$ because [his fridge is empty]$_7$. [He]$_8$ then [went to the movies]$_8$.

Figure 12: Discourse derivation trees

Figure 13: Discourse structure for (3a)
Produced "cue threading" where "connectives can be thought of as percolating from where they take scope semantically down to the clause in which they appear" (Nakatsu and White, 2010, p. 19). Here, the connective at the discourse level takes scope over its argument, but it is interpreted at the sentential level as an auxiliary tree adjoining within the clause.

6 Conclusion

This article shows how to interface TAG-based sentential and discourse grammars without resorting to a two step process. It relies on the interpretation of abstract terms encoding discourse derivation trees into terms encoding sentential derivation trees using ACG. The approach also allows us to build DAG discourse structures. ACG grammars have been implemented to compute (and parse) the surface forms and associate them with the relevant semantic forms. In this article, we only applied the approach to D-STAG, but it should be clear that it applies to D-LTAG as well. The approach is also suitable to model connective modifications (...probably because it rains). Our future work will concern multiple connectives (...because then he discovered he was broke), some of them we already account for. It will also concern the integration of discourse structure constraints such as the right frontier principle and the interaction with pronominal anaphora resolution.

Finally, discourse grammars are highly ambiguous. Hence the ACG we derive from such grammars also are ambiguous. We want to take advantage of our integrated approach to apply the disambiguation methods used in syntactic parsing. Moreover, as the analysis can now be dealt with at the level of the text, even with polynomial algorithms, the size of the input will be an issue. This calls for further analysis of discourse structuring, both in parsing and generation.
References


