Interfacing Sentential and Discourse TAG-based Grammars

Laurence Danlos, Aleksandre Maskharashvili, Sylvain Pogodalla

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 *dicourse 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.).

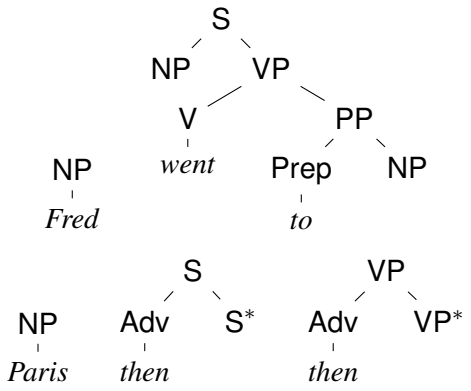


Figure 1: Elementary trees of a toy TAG grammar

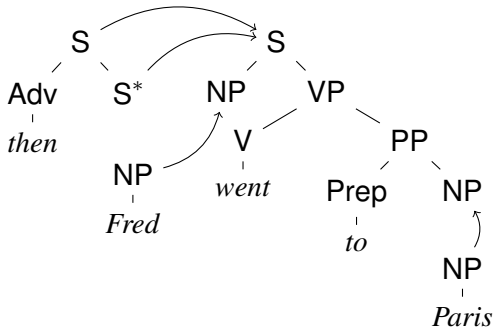


Figure 2: TAG analysis of (1a)

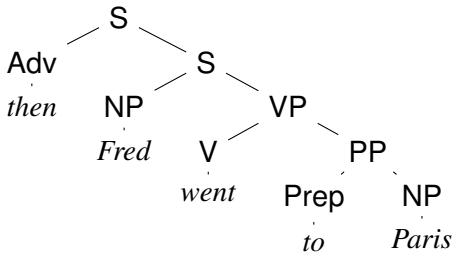


Figure 3: Derived tree for (1a)

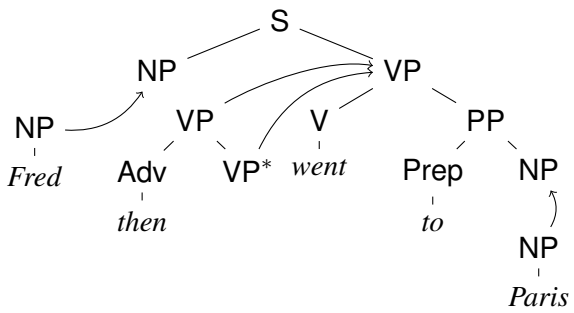


Figure 4: TAG analysis of (1b)

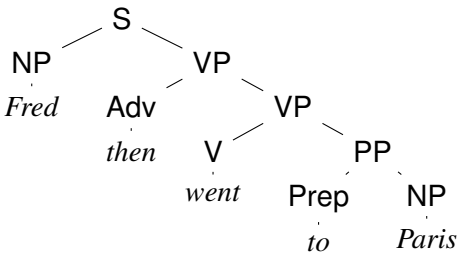


Figure 5: Derived tree for (1b)

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.

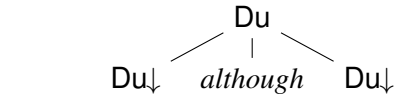
ing feature if we want to better describe the interaction between discourse connectives and propositional attitude predicates (Danlos, 2013; Bernard and Danlos, 2016).

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).

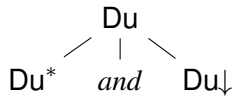
In this article, we describe how to interface a sentential and a discourse TAG-based grammar. We show how to link such two grammars and their proposed modelings of discourse connectives, overcoming the above mentioned issue. We use an encoding of TAG into Abstract Categorial Grammar (ACG, de Groote (2001)), 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 tested[2] on the ACG development software.[3]

2 TAG Based Discourse Grammars

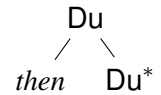
As TAG grammars, D-LTAG and D-STAG do not differ from any other TAG grammar: they define elementary trees that can be combined using the operations of substitution and adjunction. However, if some elementary trees are anchored by lexical items (the discourse markers) as in sentential grammars, the others 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.



(a) D-LTAG initial trees for subordinate conjunctions



(b) D-LTAG auxiliary trees for coordinating conjunctions (*and* and ϵ)



(c) D-LTAG auxiliary trees for connective adverbials

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

[2]The ACG example files can be downloaded from http://hal.inria.fr/hal-01328697v3/file/acg-examples.zip. They also include the semantic interpretation that generates the expected DAG discourse structures. But because of lack of space, we cannot present here the semantic part that builds on the one proposed for D-STAG (Danlos, 2009; Danlos, 2011) and extends it.

[3]http://www.loria.fr/equipes/calligramme/acg/#Software.

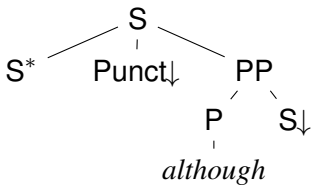


Figure 7: Syntactic modeling of subordinate conjunctions

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 clause-medial 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

[4]Such structures are not easily available with D-LTAG, and this was a motivation to introduce D-STAG.

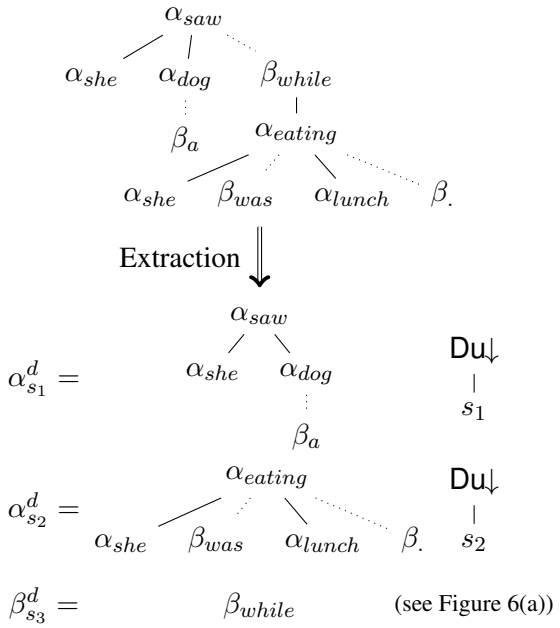
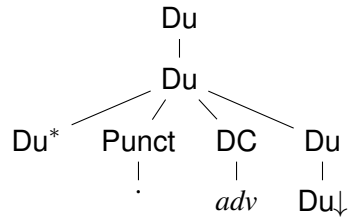


Figure 8: Discourse elementary tree extraction

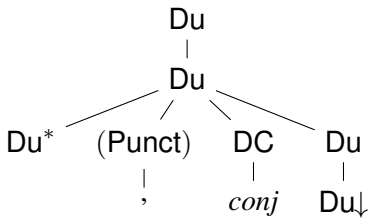
not enter the details of the discourse-semantics interface here (see Danlos et al. (2015)).

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 "discourse 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 clause-medial extractions need to be managed at this level as well.

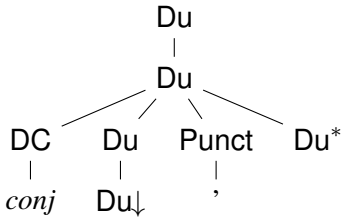
Medial Adverbial Extraction Looking at the elementary trees of Figure 9(a) (the problem is similar for D-LTAG elementary trees), we observe that the host clause of the adverbial are *substituted* into the elementary tree, at the Du↓ node. But at the sentential level, it is auxiliary trees anchored by 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



(a) Adverbial connectives



(b) Postposed conjunctions

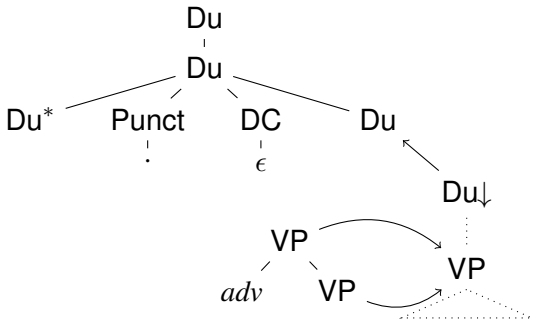


(c) Preposed conjunction

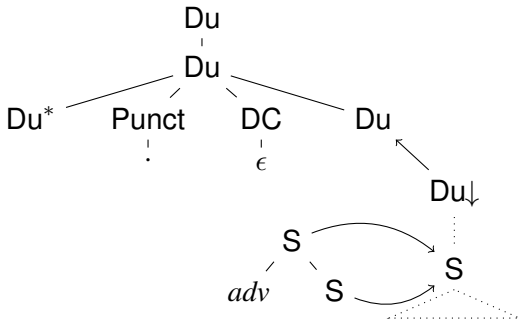
Figure 9: D-STAG elementary trees

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 at Du$\downarrow$ 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.



(a)



(b)

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 $\mathscr{T}(A)$ of *implicative types* built upon A is defined with the following grammar:[5]

$$\mathscr{T}(A) ::= A | \mathscr{T}(A) \multimap \mathscr{T}(A)$$

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

[5]We use the linear arrow $\multimap$ of linear logic (Girard, 1987) for the implication.

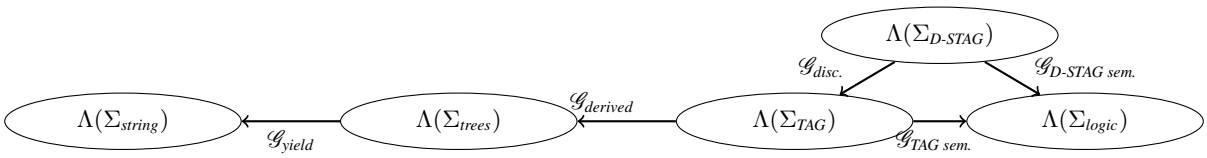


Figure 11: ACG architecture for a discourse and clause grammar interface

- A is a finite set of atomic types;
- C is a finite set of constants;
- $\tau : C \to \mathscr{T}(A)$ is a function assigning types to constants.

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

Definition (Lexicon)**.** Let $\Sigma_1 = \langle A_1, C_1, \tau_1 \rangle$ and $\Sigma_2 = \langle A_2, C_2, \tau_2 \rangle$ be two higher-order signatures. A lexicon $\mathscr{L} = \langle F, G \rangle$ from Σ_1 to Σ_2 is such that:

- $F : A_1 \to \mathscr{T}(A_2)$. We also note $F : \mathscr{T}(A_1) \to \mathscr{T}(A_2)$ its homomorphic extension;[6]
- $G : C_1 \to \Lambda(\Sigma_2)$. We also note $G : \Lambda(\Sigma_1) \to \Lambda(\Sigma_2)$ 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 $\mathscr{L}$ instead of F or G.

Definition (Abstract Categorial Grammar and vocabulary)**.** An *abstract categorial grammar* is a quadruple $\mathscr{G} = \langle \Sigma_1, \Sigma_2, \mathscr{L}, \mathsf{S} \rangle$ where:

- $\Sigma_1 = \langle A_1, C_1, \tau_1 \rangle$ and $\Sigma_2 = \langle A_2, C_2, \tau_2 \rangle$ are two higher-order signatures. Σ_1 (resp. Σ_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*).
- $\mathscr{L} : \Sigma_1 \to \Sigma_2$ is a lexicon.
- $\mathsf{S} \in \mathscr{T}(A_1)$ is the *distinguished type* of the grammar.

Given an ACG $\mathscr{G}_{name} = \langle \Sigma_1, \Sigma_2, \mathscr{L}_{name}, \mathsf{S} \rangle$, we use the following notational variants for the interpretation β (resp. u) of the type α (resp. of the term t): $\mathscr{G}_{name}(\alpha) = \beta$ and $\alpha :=_{name} \beta$ (resp. $\mathscr{G}_{name}(t) = u$ and $t :=_{name} u$). The subscript may be omitted if clear from the context.

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

$$\mathcal{A}(\mathscr{G}) = \{t \in \Lambda(\Sigma_1) \,|\, t :_{\Sigma_1} \mathsf{S}\}$$

The *object language* is defined by

$$\mathcal{O}(\mathscr{G}) = \{u \in \Lambda(\Sigma_2) \,|\, \exists t \in \mathcal{A}(\mathscr{G}) \text{ s.t. } u = \mathscr{G}(t)\}$$

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 $\mathscr{G}_{derived}$ and $\mathscr{G}_{TAG\ sem.}$ sharing the Σ_{TAG} 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 $\mathscr{G}_{yield} \circ \mathscr{G}_{derived}$ 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 $\mathscr{G}_{disc.}$.

[6] Such that $F(\alpha \multimap \beta) = F(\alpha) \multimap F(\beta)$.

[7] Such that $G(\lambda x.t) = \lambda x.G(t)$ and $G(t\ u) = G(t)\ G(u)$.

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 Σ_{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 $\gamma_{\textit{lex. entry}}$, there is a constant $C_{\textit{lex. entry}}$ 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 $\mathscr{G}_{derived}$ in the higher-order signature whose unique atomic type is τ 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 $X_n : \overbrace{\tau \multimap \cdots \multimap \tau}^{n \text{ times}} \multimap \tau$. Then $\mathscr{G}_{yield}$ interprets τ into σ, 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 $\Lambda(\Sigma_{TAG})$ representing a derivation with an adjunction at the S node (resp. at the VP node) of the given sentences as the equation (2a) (resp. (2b)) shows.

(2a) $\mathscr{G}_{yield} \circ \mathscr{G}_{derived}(C_{\textit{went to}}\ C^{\mathsf{S}}_{\textit{then}}\ I_{\mathsf{VP}}\ C_{\textit{Fred}}\ C_{\textit{Paris}}) = \textit{then} + \textit{Fred} + \textit{went} + \textit{to} + \textit{Paris}$

(2b) $\mathscr{G}_{yield} \circ \mathscr{G}_{derived}(C_{\textit{went to}}\ I_{\mathsf{S}}\ C^{\mathsf{VP}}_{\textit{then}}\ C_{\textit{Fred}}\ C_{\textit{Paris}}) = \textit{Fred} + \textit{then} + \textit{went} + \textit{to} + \textit{Paris}$

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:

- $\mathscr{G}_{disc.}$ 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.
- $\mathscr{G}_{TAG\ sem.}$[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 $\Sigma_{D\text{-}STAG}$ 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 $d^{\mathsf{S}}_{\textit{then}}$ has type $\mathsf{DC} \triangleq \mathsf{Du}_A \multimap \mathsf{Du}_A \multimap \mathsf{Du}_A \multimap \mathsf{Du} \multimap \mathsf{Du}_A$ that reflects the auxiliary trees of D-STAG (Figure 9). For comparison, see the encoding of the $C^{\mathsf{VP}}_{\textit{then}}$ 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_{dm} of type DC at the discourse level interpreted using *the corresponding auxiliary tree* C_{dm} 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_{dm} or, in D-STAG terms, the one plugged into the $\mathsf{Du}\downarrow$ 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 $\mathsf{S}_A \multimap \mathsf{VP}_A \multimap \mathsf{S}$.[9] We actually rather interpret Du as $\mathsf{S}_A \multimap (\mathsf{VP}_A \multimap \mathsf{VP}_A) \multimap \mathsf{S}$ in order to account for clause-medial adverbials occurring between to other adverbs such as in *John suddenly then passionately kissed her*.[10] Accord-

[8] Not discussed here but implemented in the example files.

[9] Another solution would be to have DC requires a $(\mathsf{S}_A \multimap \mathsf{VP}_A \multimap \mathsf{Du})$ type as fourth parameter. But the ACG would not be second-order anymore.

[10] It should be clear that from a technical point of view, both fronted and clause-medial missing adverbials could be dealt with the same way (i.e. with a $\mathsf{S}_A \multimap \mathsf{VP}_A \multimap \mathsf{S}$ or a

Constants of Σ_{TAG}		Their interpretations by $\mathscr{G}_{derived}$	
C_{Fred}	: NP	γ_{Fred}	$: \tau$
		γ_{Fred}	$= \mathsf{NP}_1\ \mathit{Fred}$
$C_{went\ to}$	: $\mathsf{S}_A \multimap \mathsf{VP}_A \multimap \mathsf{NP} \multimap \mathsf{NP} \multimap \mathsf{S}$	$\gamma_{went\ to}$	$: (\tau \multimap \tau) \multimap (\tau \multimap \tau) \multimap \tau \multimap \tau \multimap \tau$
		$\gamma_{went\ to}$	$= \lambda SAsc.S(\mathsf{S}_2\ s\ (A\ (\mathsf{VP}_2\ (\mathsf{V}_1\ \mathit{went})\ (\mathsf{PP}_2\ (\mathsf{Prep}\ \mathit{to})\ c))))$
C^{S}_{then}	: S_A	$\gamma^{\mathsf{S}}_{then}$	$: \tau \multimap \tau$
		$\gamma^{\mathsf{S}}_{then}$	$= \lambda x.(\mathsf{S}_2\ (\mathsf{Adv}_1\ \mathit{then})\ x)$
C^{VP}_{then}	: $\mathsf{VP}_A \multimap \mathsf{VP}_A$	$\gamma^{\mathsf{VP}}_{then}$	$: (\tau \multimap \tau) \multimap \tau \multimap \tau$
		$\gamma^{\mathsf{VP}}_{then}$	$= \lambda A\ x.A\ (\mathsf{VP}_2\ (\mathsf{Adv}_1\ \mathit{then})\ x)$

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

ingly, at the discourse level, the type of an intransitive verb will be $\mathsf{S}_A \multimap \mathsf{VP}_A \multimap \mathsf{VP}_A \multimap \mathsf{NP} \multimap \mathsf{S}$ instead of $\mathsf{S}_A \multimap \mathsf{VP}_A \multimap \mathsf{NP} \multimap \mathsf{S}$, allowing to specify the two VP_A auxiliary trees that can adjoin *before* and *after* the possible discourse marker. This leads us to the interpretation of Table 2. Note that even though the same name can occur on both sides of the $:=$ symbol, the atomic types and the constants on the left hand side belong to $\Sigma_{D\text{-}STAG}$ while the (possibly complex) types and the terms on the right hand side belong to $\Lambda(\Sigma_{TAG})$.

NP_A	$:= \mathsf{NP}_A$	N_A	$:= \mathsf{N}_A$
VP	$:=$ VP	Du_A	$:= \mathsf{S}_A$
VP_A	$:= \mathsf{VP}_A \multimap \mathsf{VP}_A$	T	$:=$ S
Du	$:= \mathsf{S}_A \multimap (\mathsf{VP}_A \multimap \mathsf{VP}_A) \multimap \mathsf{S}$	NP	$:=$ NP
S	$:= \mathsf{S}_A \multimap (\mathsf{VP}_A \multimap \mathsf{VP}_A) \multimap \mathsf{S}$	N	$:=$ N
S_A	$:= \mathsf{S}_A \multimap \mathsf{S}_A$		

I_X	$: X_A \quad := \lambda P.P$
d_{Fred}	$:$ NP$:=\quad C_{Fred}$
$d_{went\ to}$	$: \mathsf{S}_A \multimap \mathsf{VP}_A \multimap \mathsf{VP}_A \multimap \mathsf{S} \multimap \mathsf{S}$
	$:= \lambda S\ a_1\ a_2\ s\ o\ c\ m.$
	$\quad C_{went\ to}\ (S\ c)(a_2(m(a_1 I_{\mathsf{VP}})))\ s\ o$
$d_{in.\ anc.}$	$: \mathsf{S} \multimap \mathsf{Du}_A \multimap \mathsf{Du}$
	$:= \lambda s\ m\ d_s\ d_v.\mathsf{mod}\ (s\ d_s\ d_v)\ m$
d_{anchor}	$: \mathsf{S} \multimap \mathsf{Du}_A \multimap \mathsf{Du}$
	$:= \lambda s\ m\ d_s\ d_v.\mathsf{mod}\ (s\ d_s\ d_v)\ m$
d^{S}_{then}	$: \mathsf{Du}_A \multimap \mathsf{Du}_A \multimap \mathsf{Du}_A \multimap \mathsf{Du} \multimap \mathsf{Du}_A$
	$:= \lambda d_1\ d_2\ d_3\ s.\mathsf{cons}\ d_1\ d_2\ d_3\ (s\ C^{\mathsf{S}}_{then}\ (\lambda x.x))$
d^{VP}_{then}	$: \mathsf{Du}_A \multimap \mathsf{Du}_A \multimap \mathsf{Du}_A \multimap \mathsf{Du} \multimap \mathsf{Du}_A$
	$:= \lambda d_1\ d_2\ d_3\ s.\mathsf{cons}\ d_1\ d_2\ d_3\ (s\ I_{\mathsf{S}}\ C^{\mathsf{VP}}_{then})$

Table 2: $\mathscr{G}_{disc.}$ interpretation for the sentence-discourse interface[12]

We exemplify our approach on the examples (3). In D-STAG, the associated discourse representation is as in Figure 13, and the discourse derivation trees is the one of Figure 12 where the C_is 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_6 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]$_8$.

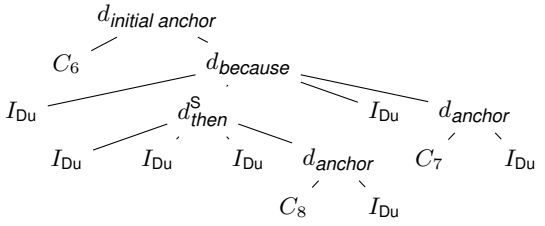


Figure 12: Discourse derivation trees

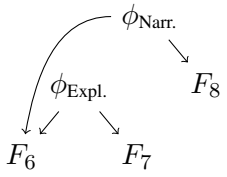
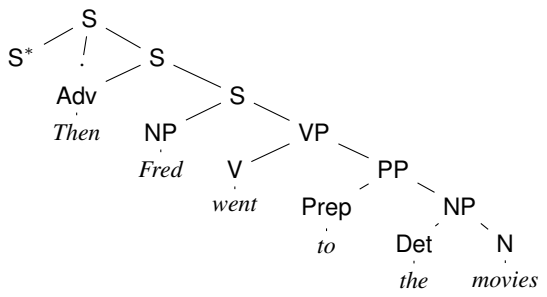


Figure 13: Discourse structure for (3a)

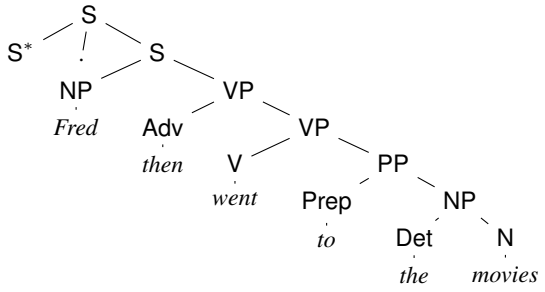
If we define the terms d_8 and d'_8 as in (4) and (5), we can compute their interpretations (6)-(11) using the lexicons of Tables 1 and 2. They show that both positions for the adverbs are now available directly from the abstract terms representing discourse derivations. Consequently, the two terms defined in (12) and (13) account for both sentences of (3). Note that they differ only in the constant they use for the adverb.

$(\mathsf{S}_A \multimap \mathsf{S}_A) \multimap (\mathsf{VP}_A \multimap \mathsf{VP}_A) \multimap \mathsf{S}$ type). We leave it for further work to check the adequacy of the same phenomena occurring for fronted adverbials and how it compares with discourse connective modification or multiple connectives.

[12] mod and cons are two operators that have no other meaning that juxtaposing TAG derivation trees of elementary discourse units. They are interpreted as: $\mathsf{mod} := \lambda s\ m.m\ s$ (it performs the actual adjunction on the derived tree) and $\mathsf{cons} := \lambda s_1\ s_2\ s_3\ s\ x.s_1(s_2(\mathsf{S}_3\ x\ .\ (s_3\ s)))$ (it builds a derived tree, inserting a period between the derived trees corresponding to the elementary discourse units).



(a) Derived tree for d_8



(b) Derived tree for d'_8

Figure 14: Interpretations as derived trees

(4) $d_8 = d^{\mathsf{S}}_{\mathit{then}}\ I_{\mathsf{Du}}\ I_{\mathsf{Du}}\ I_{\mathsf{Du}}\ (d_{\mathit{anchor}}\ C_8\ I_{\mathsf{Du}}) : \mathsf{Du}_A$

(5) $d'_8 = d^{\mathsf{VP}}_{\mathit{then}}\ I_{\mathsf{Du}}\ I_{\mathsf{Du}}\ I_{\mathsf{Du}}\ (d_{\mathit{anchor}}\ C_8\ I_{\mathsf{Du}}) : \mathsf{Du}_A$

5 Related Works

This problem of avoiding an intermediate step between a sentential and a discourse analysis has also been addressed within the framework of Combinatory Categorial Grammar (CCG, Steedman (2001); Steedman and Baldridge (2011)) by Nakatsu and White (2010). They propose a single grammar to treat both sentential and discourse phenomena using Discourse Combinatory Categorial Grammar (DCCG). This approach introduced "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.

(6) $\mathscr{G}_{\mathit{disc.}}(d_8) = \mathsf{cons}\ \ I_{\mathsf{S}}\ I_{\mathsf{S}}\ I_{\mathsf{S}}\ (\mathsf{mod}\ \ (C_{\mathit{went\ to}}\ C^{\mathsf{S}}_{\mathit{then}}\ I_{\mathsf{VP}}\ C_{\mathit{Fred}}(C_{\mathit{the}}\ (C_{\mathit{movies}}\ I_{\mathsf{N}})))\ I_{\mathsf{S}}) : \mathsf{S}_A$

(7) $\mathscr{G}_{\mathit{derived}} \circ \mathscr{G}_{\mathit{disc.}}(d_8) =$ [see the tree representation in Figure 14(a)]

(8) $\mathscr{G}_{\mathit{yield}} \circ \mathscr{G}_{\mathit{derived}} \circ \mathscr{G}_{\mathit{disc.}}(d_8) = \lambda x.x + . + \mathit{Then} + \mathit{Fred} + \mathit{went} + \mathit{to} + \mathit{the} + \mathit{movies}) : \sigma \multimap \sigma$

(9) $\mathscr{G}_{\mathit{disc.}}(d'_8) = \mathsf{cons}\ \ I_{\mathsf{S}}\ I_{\mathsf{S}}\ I_{\mathsf{S}}\ (\mathsf{mod}\ \ (C_{\mathit{went\ to}}\ I_{\mathsf{S}}\ (C^{\mathsf{VP}}_{\mathit{then}}\ I_{\mathsf{VP}})\ C_{\mathit{Fred}}(C_{\mathit{the}}\ (C_{\mathit{movies}}\ I_{\mathsf{N}})))\ I_{\mathsf{S}}) : \mathsf{S}_A$

(10) $\mathscr{G}_{\mathit{derived}} \circ \mathscr{G}_{\mathit{disc.}}(d'_8) =$ [see the tree representation in Figure 14(b)]

(11) $\mathscr{G}_{\mathit{yield}} \circ \mathscr{G}_{\mathit{derived}} \circ \mathscr{G}_{\mathit{disc.}}(d'_8) = \lambda x.x + . + \mathit{Fred} + \mathit{then} + \mathit{went} + \mathit{to} + \mathit{the} + \mathit{movies}) : \sigma \multimap \sigma$

(12) $d_3 = d_{\mathit{in.\ anc.}}\ C_6\ (d_{\mathit{because}}\ I_{\mathsf{Du}}\ (d^{\mathsf{S}}_{\mathit{then}}\ I_{\mathsf{Du}}\ I_{\mathsf{Du}}\ I_{\mathsf{Du}}\ (d_{\mathit{anc.}}\ C_8\ I_{\mathsf{Du}}))\ I_{\mathsf{Du}}\ (d_{\mathit{anc.}}\ C_7\ I_{\mathsf{Du}}))$

(13) $d'_3 = d_{\mathit{in.\ anc.}}\ C_6\ (d_{\mathit{because}}\ I_{\mathsf{Du}}\ (d^{\mathsf{VP}}_{\mathit{then}}\ I_{\mathsf{Du}}\ I_{\mathsf{Du}}\ I_{\mathsf{Du}}\ (d_{\mathit{anc.}}\ C_8\ I_{\mathsf{Du}}))\ I_{\mathsf{Du}}\ (d_{\mathit{anc.}}\ C_7\ I_{\mathsf{Du}}))$

References

Anne Abeillé. 2002. *Une grammaire électronique du français*. Sciences du langage. CNRS Éditions.

Timothée Bernard and Laurence Danlos. 2016. Modelling discourse in stag: Subordinate conjunctions and attributing phrases. In David Chiang and Alexander Koller, editors, *Proceedings of the Twelfth International Workshop on Tree Adjoining Grammars and Related Framework (TAG+12)*. HAL open archive: hal-01329539.

Laurence Danlos, Aleksandre Maskharashvili, and Sylvain Pogodalla. 2015. Grammaires phrastiques et discursives fondées sur les TAG : une approche de D-STAG avec les ACG. In *TALN 2015 - 22e conférence sur le Traitement Automatique des Langues Naturelles*, Actes de TALN 2015, pages 158–169, Caen, France. Association pour le Traitement Automatique des Langues. HAL open archive: hal-01145994.

Laurence Danlos. 1998. G-TAG : Un formalisme lexicalisé pour la génération de textes inspiré de TAG. *Traitement Automatique des Langues*, 39(2). HAL open archive: inria-00098489.

Laurence Danlos. 2000. G-TAG: A lexicalized formalism for text generation inspired by Tree Adjoining Grammar. In Anne Abeillé and Owen Rambow, editors, *Tree Adjoining Grammars: Formalisms, Linguistic Analysis, and Processing*, volume 107 of *CSLI Lecture Notes*, pages 343–370. CSLI Publications. http://www.linguist.jussieu.fr/~danlos/Dossier%20publis/G-TAG-eng'01.pdf.

Laurence Danlos. 2009. D-STAG : un formalisme d'analyse automatique de discours basé sur les TAG synchrones. *T.A.L.*, 50(1):111–143. HAL open archive: inria-00524743.

Laurence Danlos. 2011. D-STAG: a formalism for discourse analysis based on SDRT and using Synchronous TAG. In Philippe de Groote, Markus Egg, and Laura Kallmeyer, editors, *14th conference on Formal Grammar - FG 2009*, volume 5591 of *LNCS/LNAI*, pages 64–84. Springer. DOI: 10.1007/978-3-642-20169-1_5.

Laurence Danlos. 2013. Connecteurs de discours adverbiaux : Problèmes à l'interface syntaxe-sémantique. *LinguisticæInvestigationes*, 36(2):261–275. HAL open archive: hal-00932184. DOI: 10.1075/li.36.2.05dan.

Philippe de Groote and Sylvain Pogodalla. 2004. On the expressive power of Abstract Categorial Grammars: Representing context-free formalisms. *Journal of Logic, Language and Information*, 13(4):421–438. HAL open archive: inria-00112956. DOI: 10.1007/s10849-004-2114-x.

Philippe de Groote. 2001. Towards Abstract Categorial Grammars. In *Association for Computational Linguistics, 39th Annual Meeting and 10th Conference of the European Chapter, Proceedings of the Conference*, pages 148–155. ACL anthology: P01-1033.

Philippe de Groote. 2002. Tree-Adjoining Grammars as Abstract Categorial Grammars. In *Proceedings of the Sixth International Workshop on Tree Adjoining Grammars and Related Frameworks (TAG+6)*, pages 145–150. Università di Venezia. URL: http://www.loria.fr/equipes/calligramme/acg/publications/2002-tag+6.pdf.

Katherine Forbes, Eleni Miltsakaki, Rashmi Prasad, Anoop Sarkar, Aravind K. Joshi, and Bonnie Lynn Webber. 2003. D-LTAG system: Discourse parsing with a Lexicalized Tree-Adjoining Grammar. *Journal of Logic, Language and Information*, 12(3):261–279. Special Issue: Discourse and Information Structure. DOI: 10.1023/A:1024137719751.

Katherine Forbes-Riley, Bonnie Lynn Webber, and Aravind K. Joshi. 2006. Computing discourse semantics: The predicate-argument semantics of discourse connectives in D-LTAG. *Journal of Semantics*, 23(1):55–106. DOI: 10.1093/jos/ffh032.

Claire Gardent and Laura Kallmeyer. 2003. Semantic construction in feature-based TAG. In *Proceedings of the 10th Meeting of the European Chapter of the Association for Computational Linguistics (EACL)*, pages 123–130. ACL anthology: E03-1030.

Claire Gardent. 1997. Discourse tree adjoining grammar. CLAUS Report 89, Universit, Saarbr, April.

Jean-Yves Girard. 1987. Linear logic. *Theoretical Computer Science*, 50(1):1–102. DOI: 10.1016/0304-3975(87)90045-4.

Aravind K. Joshi and Yves Schabes. 1997. Tree-adjoining grammars. In Grzegorz Rozenberg and Arto K. Salomaa, editors, *Handbook of formal languages*, volume 3, chapter 2. Springer.

Aravind K. Joshi, Leon S. Levy, and Masako Takahashi. 1975. Tree adjunct grammars. *Journal of Computer and System Sciences*, 10(1):136–163. DOI: 10.1016/S0022-0000(75)80019-5.

Laura Kallmeyer and Maribel Romero. 2008. Scope and situation binding for LTAG. *Research on Language and Computation*, 6(1):3–52. DOI: 10.1007/s11168-008-9046-6.

Makoto Kanazawa. 2008. A prefix-correct earley recognizer for multiple context-free grammars. In *Proceedings of the Ninth International Workshop on Tree Adjoining Grammars and Related Formalisms (TAG+9)*, pages 49–56, Tuebingen, Germany, June 7–8. http://tagplus9.cs.sfu.ca/papers/Kanazawa.pdf.

Daniel Marcu. 2000. *The Theory and Practice of Discourse Parsing and Summarization*. The MIT Press.

Frédéric Meunier. 1997. *Implantation du formalisme de génération G-TAG*. Ph.D. thesis, Université Paris 7 — Denis Diderot.

Crystal Nakatsu and Michael White. 2010. Generating with discourse combinatory categorial grammar. *Linguistic Issues in Language Technology*, 4. http://journals.linguisticsociety.org/elanguage/lilt/article/view/1277.html.

Rebecca Nancy Nesson and Stuart M. Shieber. 2006. Simpler TAG semantics through synchronization. In *Proceedings of the 11th Conference on Formal Grammar*, Malaga, Spain, 7. CSLI Publications. http://cslipublications.stanford.edu/FG/2006/nesson.pdf.

Sylvain Pogodalla. 2004. Computing Semantic Representation: Towards ACG Abstract Terms as Derivation Trees. In *Seventh International Workshop on Tree Adjoining Grammar and Related Formalisms - TAG+7*, pages 64–71, Vancouver, BC, Canada. HAL open archive: inria-00107768.

Sylvain Pogodalla. 2009. Advances in Abstract Categorial Grammars: Language Theory and Linguistic Modeling. ESSLLI 2009 Lecture Notes, Part II. HAL open archive: hal-00749297.

Livia Polanyi and Martin H. van den Berg. 1996. Discourse structure and discourse interpretation. In Paul J. E. Dekker and Martin Stokhof, editors, *Proceedings of the Tenth Amsterdam Colloquium*. ILLC/Department of Philosophy, University of Amsterdam. http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.56.221.

Owen Rambow, K. Vijay-Shanker, and David Weir. 2001. D-Tree Substitution Grammars. *Computational Linguistics*, 27(1):87–121. ACL anthology: J01-1004. DOI: 10.1162/089120101300346813.

Frank Schilder. 1997. Tree discourse grammar or how to get attached to a discourse? In *In Proceedings of the Tilburg Conference on Formal Semantics (IWCS-1997)*, pages 261–273. ftp://ftp.informatik.uni-hamburg.de/pub/unihh/informatik/WSV/schild97a.ps.gz.

Stuart M. Shieber and Yves Schabes. 1990. Synchronous tree-adjoining grammars. In *Proceedings of the 13th International Conference on Computational Linguistics*, volume 3, pages 253–258, Helsinki, Finland. http://www.eecs.harvard.edu/~shieber/Biblio/Papers/synch-tags.pdf. DOI: 10.3115/991146.991191.

Stuart M. Shieber. 2006. Unifying synchronous tree-adjoining grammars and tree transducers via bimorphisms. In *Proceedings of the 11th Conference of the European Chapter of the Association for Computational Linguistics (EACL-06)*, pages 377–384, Trento, Italy, 3–7 April. ACL anthology: E06-1048.

Radu Soricut and Daniel Marcu. 2003. Sentence level discourse parsing using syntactic and lexical information. In Marti Hearst and Mari Ostendorf, editors, *Proceedings of the 2003 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics (HLT-NAACL 2003)*, pages 149–156. ACL anthology: N03-1030.

Mark Steedman and Jason Baldridge. 2011. Combinatory categorial grammar. In Robert Borsley and Kersti Börjars, editors, *Non-Transformational Syntax: Formal and Explicit Models of Grammar*, chapter 5. Wiley-Blackwell.

Mark Steedman. 2001. *The Syntactic Process*. MIT Press.

Bonnie Lynn Webber and Aravind K. Joshi. 1998. Anchoring a lexicalized tree-adjoining grammar for discourse. In Manfred Stede, Leo Wanner, and Eduard Hovy, editors, *Proccedings of the ACL/COLING workshop on Discourse Relations and Discourse Markers*. ACL anthology: W98-0315.

Bonnie Lynn Webber. 2004. D-LTAG: extending lexicalized TAG to discourse. *Cognitive Science*, 28(5):751–779. DOI: 10.1207/s15516709cog2805_6.

XTAG Research Group. 2001. A Lexicalized Tree Adjoining Grammar for English. Technical Report IRCS-01-03, IRCS, University of Pennsylvania. ftp://ftp.cis.upenn.edu/pub/xtag/release-2.24.2001/tech-report.pdf.