

Grammaires phrastiques et discursives fondées sur les TAG : une approche de D-STAG avec les ACG *

Laurence Danlos^{1,2,3} Aleksandre Maskharashvili^{4,5,6} Sylvain Pogodalla^{4,5,6,7}

(1) Université Paris Diderot (Paris 7), Paris, F-75013, France

(2) ALPAGE, INRIA Paris–Rocquencourt, Paris, F-75013, France

(3) Institut Universitaire de France, Paris, F-75005, France

(4) INRIA, Villers-lès-Nancy, F-54600, France

(5) Université de Lorraine, LORIA, UMR 7503, Vandœuvre-lès-Nancy, F-54500, France

(6) CNRS, LORIA, UMR 7503, Vandœuvre-lès-Nancy, F-54500, France

(7) Heinrich Heine Universität, Düsseldorf, Allemagne

{laurence.danlos}{aleksandre.maskharashvili}{sylvain.pogodalla}@inria.fr

Résumé. Nous présentons une méthode pour articuler grammaire de phrase et grammaire de discours qui évite de recourir à une étape de traitement intermédiaire. Cette méthode est suffisamment générale pour construire des structures discursives qui ne soient pas des arbres mais des graphes orientés acycliques (DAG). Notre analyse s’appuie sur une approche de l’analyse discursive, Discourse Synchronous TAG (D-STAG), qui utilise les Grammaires d’Arbres Adjoint (TAG). Nous utilisons pour ce faire un encodage des TAG dans les Grammaires Catégorielles Abstraites (ACG). Cet encodage permet d’une part d’utiliser l’ordre supérieur pour l’interprétation sémantique afin de construire des structures qui soient des DAG et non des arbres, et d’autre part d’utiliser les propriétés de composition d’ACG pour réaliser naturellement l’interface entre grammaire phrastique et grammaire discursive. Tous les exemples proposés pour illustrer la méthode ont été implantés et peuvent être testés avec le logiciel approprié.

Abstract.

Sentential and Discourse TAG-Based Grammars: An ACG Approach to D-STAG

This article presents a method to interface a sentential grammar and a discourse grammar without resorting to an intermediate processing step. The method is general enough to build discourse structures that are direct acyclic graphs (DAG) and not only trees. Our analysis is based on Discourse Synchronous TAG (D-STAG), a Tree-Adjoining Grammar (TAG)-based approach to discourse. We also use an encoding of TAG into Abstract Categorical Grammar (ACG). This encoding allows us to express a higher-order semantic interpretation that enables building DAG discourse structures on the one hand, and to smoothly integrate the sentential and the discourse grammar thanks to the modular capability of ACG. All the examples of the article have been implemented and may be run and tested with the appropriate software.

Mots-clés : Syntaxe, sémantique, discours, grammaire, grammaire d’arbres adjoints, TAG, D-LTAG, D-STAG, grammaire catégorielle abstraite, ACG.

Keywords: Syntax, semantics, discourse, grammar, Tree-Adjoining Grammar, TAG, D-LTAG, D-STAG, Abstract Categorical Grammar, ACG.

1 Introduction

On considère généralement que la structure interne d’un texte, construite en particulier par l’utilisation de relations de discours, joue un rôle important dans son interprétation. Différentes techniques peuvent être mises en œuvre afin de calculer automatiquement cette structure. Certaines techniques se fondent sur la segmentation du discours en unités discursives élémentaires, puis en l’identification des relations susceptibles de les relier (Marcu, 2000; Soricut & Marcu, 2003). D’autres techniques utilisent des *grammaires du discours*, et en particulier des grammaires d’arbres (Polanyi & van den Berg, 1996; Gardent, 1997; Schilder, 1997). Les propriétés de ces dernières, notamment le besoin d’une opération d’adjonction, a conduit à utiliser le même formalisme grammatical des Grammaires d’Arbres Adjoints (TAG) (Joshi *et al.*, 1975; Joshi

*. Ce travail a bénéficié d’une aide de l’Agence Nationale de la Recherche portant la référence (ANR-12-CORD-0004).

& Schabes, 1997) pour la description des grammaires phrastiques et des grammaires discursives, notamment avec TAG pour le discours (D-LTAG) (Webber & Joshi, 1998; Forbes *et al.*, 2003; Webber, 2004; Forbes-Riley *et al.*, 2006).

Ces approches présentent deux caractéristiques importantes. Premièrement, bien qu’elles s’appuient sur un seul et unique formalisme grammatical, deux grammaires différentes sont utilisées d’une part pour l’analyse syntaxique et d’autre part pour l’analyse discursive. Cette dichotomie, qui implique la présence d’une étape intermédiaire de traitement, complique la modélisation des marqueurs discursifs dont l’usage peut être ambigu entre rôle syntaxique et rôle discursif. Il empêche également le traitement de ces ambiguïtés par les méthodes classiques adoptées en analyse. Deuxièmement, les structures discursives sont données directement par les arbres de dérivation. Or cette représentation sous forme d’arbres ne permet pas de rendre compte d’analyses pour lesquelles une structure de graphe orienté acyclique (DAG) pourrait sembler plus adaptée. Le formalisme grammatical réduit donc le choix des théories linguistiques modélisables.

Pour obvier au premier problème, Nakatsu & White (2010) proposent une grammaire unique rendant compte à la fois des phénomènes phrastiques et des phénomènes discursifs. L’approche se fonde sur une Grammaire Catégorielle Combinatoire du Discours (DCCG), une extension à la Grammaire Catégorielle Combinatoire (CCG) (Steedman, 2001; Steedman & Baldrige, 2011). Elle propose également une interface vers la représentation sémantique. Cependant, elle ne propose pas de modélisation pour les structures discursives représentées par un DAG (les structures « multi-parents » représentées aux figures 1(b) et 1(c)).

Pour répondre au deuxième problème et permettre la construction de structures multi-parents, Danlos (2009, 2011) définit Discourse Synchronous TAG (D-STAG), un formalisme fondé sur les TAG offrant une interprétation sémantique avec de l’ordre supérieur. Ainsi, D-STAG peut engendrer par exemple les structures discursives de la figure 1 qui correspondent aux exemples (1–4) tirés de (Danlos, 2009). Cependant, cette approche nécessite elle aussi un traitement en trois étapes. À la première étape, chaque phrase est analysée. À la deuxième étape, de même qu’en D-LTAG, une suite de clauses et de marqueurs de discours est extraite de la suite des analyses des phrases du texte pour construire ce qui est appelé une « forme normale discursive ». À la troisième étape, une grammaire TAG synchrone discursive analyse cette forme normale, produisant les représentations sémantiques souhaitées.

- (1) [Fred est grognon]₁ parce qu’[il a perdu ses clefs]₂. De plus, [il a raté son permis de conduire]₃.
- (2) [Fred est grognon]₁ parce qu’[il a mal dormi]₄. [Il a fait un cauchemar]₅.
- (3) [Fred est allé au supermarché]₆ parce que [le frigo était vide]₇. Ensuite, [il est allé au cinéma]₈.
- (4) [Fred est grognon]₁ parce que [sa femme est absente une semaine]₉. [Ceci prouve qu’il l’aime beaucoup]₁₀.

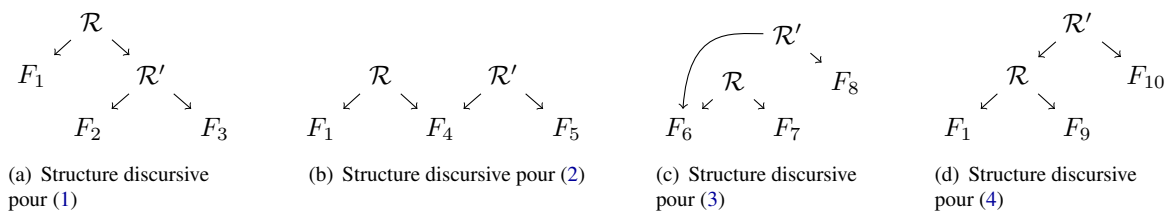


FIGURE 1 – Structures discursives possibles reliant trois unités de discours

Dans cet article, nous présentons une nouvelle approche permettant l’articulation d’une grammaire phrastique et d’une grammaire discursive à la D-STAG. Le but est d’éviter le recours à une étape intermédiaire de traitement résultant de l’indépendance complète entre les deux grammaires. Pour ce faire, nous utilisons le codage des TAG dans les Grammaires Catégorielles Abstraites (ACG) (de Groot, 2001), un cadre grammatical défini à partir de la théorie des types et du λ -calcul permettant l’encodage de nombreux formalismes grammaticaux.

Deux caractéristiques des ACG sont mises en œuvre : d’une part l’encodage des opérations d’adjonction et de substitution des grammaires TAG en ACG et l’interface syntaxe-sémantique qu’il offre pour les TAG (Pogodalla, 2004, 2009), et d’autre part la modularité de la composition d’ACG. Cette dernière est cruciale pour interfacier les deux grammaires et rendre compte des propriétés syntaxiques aussi bien que discursives des connecteurs sans recourir à une analyse en trois étapes. Nous reprenons la forme des arbres élémentaires discursifs des connecteurs de D-STAG, qui tiennent compte d’hypothèses psycho-linguistiques sur l’incrémentalité de l’extension du discours (par adjonction sur le discours déjà construit). Lors de l’interfaçage avec la grammaire phrastique, cette contrainte conduit à l’utilisation d’opérations qui sont au-delà de celles proposées par TAG mais qui s’expriment naturellement dans les ACG.

Cette intégration entre grammaire phrastique et grammaire discursive en une seule étape présente trois avantages. Premièrement, elle permet de réutiliser aussi bien des grammaires phrastiques TAG que des grammaires TAG discursives (nous ne connaissons pas de grammaire CCG à large couverture pour le français). Deuxièmement, elle permet d'éviter le découpage en plusieurs étapes et permet d'envisager à terme l'utilisation des techniques classiques de désambiguïsation prenant en compte à la fois le niveau phrastique et le niveau discursif. Les grammaires discursives sont en effet très ambiguës. Notre approche ne change pas cela. Elle présente la même ambiguïté que les approches en plusieurs étapes. Mais avoir un seul traitement d'analyse permet d'utiliser les techniques de désambiguïsation déjà existantes plutôt que de considérer l'interaction entre un modèle au niveau phrastique et un modèle au niveau discursif. Un troisième avantage est d'offrir un même cadre et une même grammaire non seulement pour l'analyse, mais aussi pour la génération de textes de plusieurs phrases reliées par des connecteurs de discours. Les ACG dites de second ordre sont en effet réversibles (Kanazawa, 2007). Un texte structuré peut donc s'obtenir par réalisation de surface.

Le choix de D-STAG nous permet aussi de construire des structures discursives sous forme de DAG. Pour ce faire, nous nous appuyons au niveau sémantique sur les analyses proposées dans (Danlos, 2009, 2011), et en particulier sur l'utilisation de l'ordre supérieur. C'est pourquoi nos exemples illustrent la chaîne d'interprétation jusqu'au calcul d'une représentation sémantique s'appuyant sur la théorie de représentation du discours segmentée (SDRT) (Asher & Lascarides, 2003) pour laquelle les structures ne sont pas nécessairement des arbres. Mais cela ne restreint pas la généralité de l'approche qui pourrait sans doute construire des représentations fondées sur la théorie de la structure rhétorique (RST) (Mann & Thompson, 1988) et des structures sous forme d'arbre, comme pour D-LTAG. Les λ -termes exemples, notamment sémantiques, pouvant être complexes (même si l'objet de l'article n'est pas d'expliquer comment ils fonctionnent, mais de décrire l'interface entre grammaire phrastique et grammaire discursive), nous proposons des grammaires qui les implantent. Ils peuvent donc être vérifiés en utilisant le logiciel de développement d'ACG proposé par l'équipe Sémagramme du LORIA¹.

Nous présentons brièvement les approches D-LTAG et D-STAG à la section 2. Nous montrons en particulier les difficultés de la prise en compte des propriétés discursives des adverbiaux médiaux qui ont conduit ces deux approches à recourir à un traitement intermédiaire. En section 3 nous présentons les ACG. Nous pouvons alors montrer en section 4 comment les TAG sont encodées dans les ACG, et, en suivant les mêmes principes, comment encoder D-STAG dans les ACG. La section 5 décrit la réalisation de l'interface entre ces grammaires dans les ACG. La section 6 présente finalement la manière d'obtenir la représentation sémantique associée.

2 Grammaires TAG pour le discours

Nous renvoyons le lecteur à (Webber & Joshi, 1998; Forbes *et al.*, 2003; Webber, 2004; Forbes-Riley *et al.*, 2006) pour une présentation en profondeur de D-LTAG et à (Danlos, 2009, 2011) pour une présentation de D-STAG et une comparaison entre ces deux approches. Ici, nous ne soulignons que leurs différences principales et leurs limitations.

D-LTAG D-LTAG propose trois familles principales d'arbres élémentaires pour rendre compte des différents effets des marqueurs discursifs sur la structure du discours. La première famille concerne les conjonctions de subordination. En s'appuyant sur leur structure prédicat-argument au niveau discursif, D-LTAG les modélise à l'aide d'arbres initiaux avec deux nœuds à substitution pour chacun des arguments comme le montre la figure 2(a)². Cela contraste avec la modélisation au niveau phrastique (syntaxique) de ces connecteurs, qui sont habituellement encodés par des arbres *auxiliaires* car ils ne font pas partie du domaine de localité des verbes des clauses dans lesquelles ils apparaissent. Ils sont en effet adjoints aux nœuds S ou V de ces derniers. La seconde famille de connecteurs est utilisée pour étendre une clause avec des arbres auxiliaires ancrés par des conjonctions de coordination (ou par le connecteur vide ϵ). Le premier argument sémantique de la relation de discours correspond alors à l'unité de discours à laquelle l'arbre du connecteur est adjoint. Le second argument sémantique correspond à la clause qui est substituée au nœud pied de l'arbre du connecteur, comme le montre la figure 2(b). La troisième famille consiste également en arbres auxiliaires. Mais ces derniers sont associés à *une unique* clause comme le montre la figure 2(c). L'autre argument des connecteurs qui ancrent de tels arbres s'obtient par résolution anaphorique vers une unité de discours précédente (Webber *et al.*, 2003).

1. Les fichiers ACG d'exemples peuvent être téléchargés depuis <https://hal.inria.fr/hal-01145994/file/taln-2015-dstag-acg-examples.zip>. Le logiciel est disponible à <http://www.loria.fr/equipes/calligramme/acg/#Software>.

2. Le type Du correspondant à une unité de discours. Cette unité peut être élémentaire lorsque c'est une simple clause au niveau phrastique (exemples (2)) et (3), ou complexe lorsqu'il s'agit de plusieurs clauses elles-mêmes reliées par des relations de discours (exemples (1) et (4)).

Le traitement en trois étapes fonctionne de la manière suivante : tout d'abord, chaque phrase reçoit une analyse TAG (arbre dérivé et arbre de dérivation) par une TAG phrastique standard. Puis, dans un deuxième temps, chaque arbre de dérivation est traité pour identifier les connecteurs de discours (DC) possibles et leurs arguments d'un point de vue syntaxique. Ces derniers (un ou deux suivant le connecteur) sont ajoutés comme arbres initiaux de racine Du à la grammaire du discours, de même que l'arbre élémentaire ancré par le connecteur détecté (voir figure 3). Un procédé similaire réalise l'extraction des connecteurs adverbiaux médiaux. Enfin, dans un troisième temps, la séquence ainsi obtenue est analysée à l'aide de la grammaire discursive.

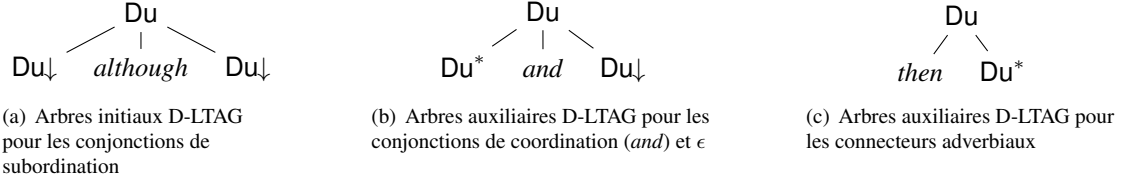


FIGURE 2 – Schémas des arbres élémentaires D-LTAG

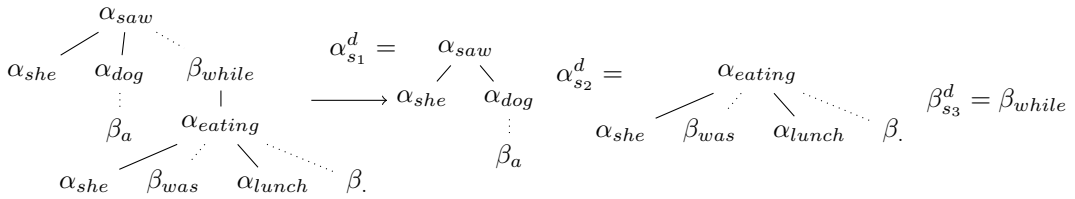


FIGURE 3 – Extraction des arbres élémentaires de la grammaire du discours

D-STAG En D-STAG, tous les connecteurs de discours sont modélisés avec des arbres auxiliaires qui sont adjoints au texte qu'ils étendent. Le contenu de la clause ajoutée est le second argument du connecteur et s'obtient par substitution. La figure 4 montre quelques schémas de ces arbres auxiliaires³. Suivant le principe des TAG synchronisées, chaque arbre élémentaire est associé à un arbre sémantique pour former une paire (arbre syntaxique, arbre sémantique). Ici, les arbres sémantiques ont une racine qui est soit t , le type des propositions, soit le type d'ordre supérieur $(t \rightarrow t) \rightarrow t$ nécessaire pour l'obtention des structures DAG. (Danlos, 2009) introduit deux termes $\Phi'_{\mathcal{R}}$ et $\Phi''_{\mathcal{R}}$ comme en (5) et la sémantique d'un connecteur associé à un arbre tel que ceux de la figure 4 s'obtient alors par : $\lambda p_4 p_3 p_2 p_1 p_0.p_4 (\phi (p_3 p_0) (p_2 p_1))$ où $\phi \in \{\Phi'_{\mathcal{R}}, \Phi''_{\mathcal{R}}\}$ et p_0 correspond à la sémantique de la clause auquel l'arbre est adjoint, p_1 celle de la clause qui est substituée, p_2, p_3 , et p_4 celles des arbres auxiliaires éventuellement adjoints au nœuds ②, ③, et ④ de l'arbre. En D-STAG, les connecteurs ancrent deux telles paires : une dont la partie sémantique met en jeu Φ' , et une dont la partie sémantique met en jeu Φ'' . (Danlos, 2009) détaille les raisons qui permettent à ces formules de rendre compte des phénomènes que nous souhaitons modéliser.

(5)
$$\Phi'_{\mathcal{R}} = \lambda X Y.X(\lambda x.Y(\lambda y.\mathcal{R}(x, y)))$$

$$\Phi''_{\mathcal{R}} = \lambda X Y P.X(\lambda x.Y(\lambda y.\mathcal{R}(x, y) \wedge P(x)))$$

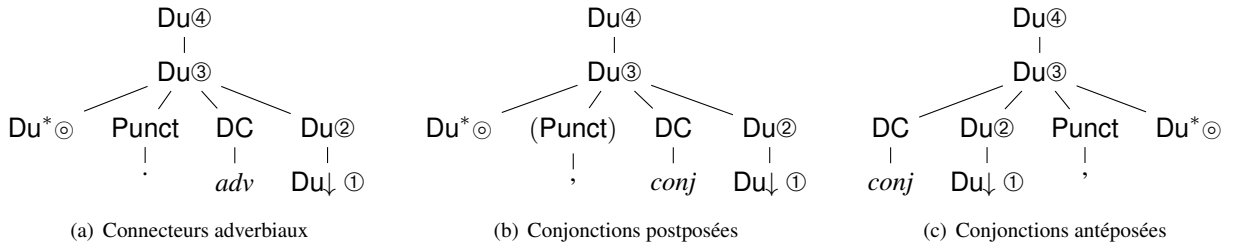


FIGURE 4 – Arbres élémentaires D-STAG

Comme en D-LTAG, avant que la grammaire discursive ci-dessus ne puisse s'appliquer, il est nécessaire de réaliser un pré-traitement pour obtenir une *forme normale discursive* qui peut seulement ensuite être analysée. Elle s'obtient de

3. Le connecteur vide ϵ est traité comme un adverbial.

manière similaire à celle exposée pour D-LTAG et est essentiellement motivée par les mêmes raisons : les connecteurs de discours et les unités de discours élémentaires ont besoin d’être identifiés pour être ancrés dans la grammaire discursive, et l’extraction des adverbiaux médiaux doit s’opérer lors de cette phase.

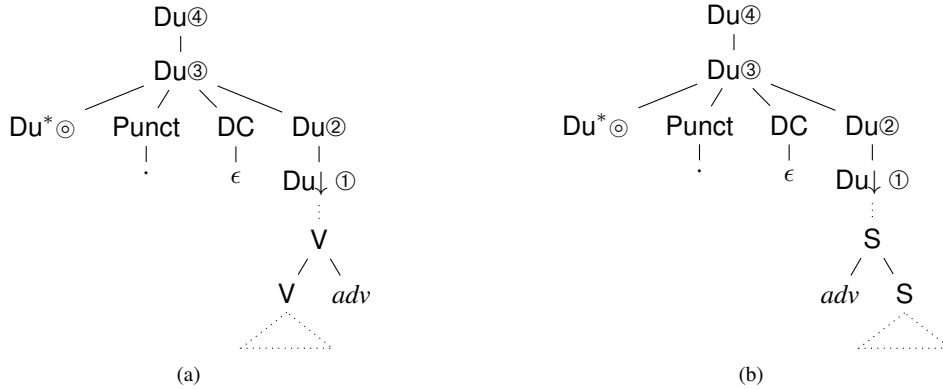


FIGURE 5 – Arbres auxiliaires pour les connecteurs de discours

Extraction des adverbiaux médiaux En observant les arbres de la figure 4 (le problème est similaire en D-LTAG), on constate que la clause hôte du connecteur de discours est *substituée* dans l’arbre élémentaire (au nœud Du^\downarrow). Or, au niveau phrastique, les adverbiaux sont *adjoints* à cette clause. Pour les connecteurs en position initiale, les arbres D-STAG permettent d’engendrer les mêmes formes de surface, le connecteur étant placé juste *avant* la clause hôte, comme pour l’analyse phrastique. Ceci n’est plus vrai si le connecteur n’est pas en position initiale, par exemple en position médiale, et l’arbre discursif ne permet pas d’obtenir la même forme de surface que l’arbre phrastique. Dès lors, une forme intermédiaire comme la forme normale discursive est nécessaire : elle permet de déplacer les connecteurs adverbiaux médiaux en position initiale. Pour s’en passer, il faudrait pouvoir décrire une opération qui substitue une clause dans l’arbre discursif du connecteur vide et simultanément lui adjoint l’arbre auxiliaire de l’adverbial. La figure 5(a) décrit une telle opération. Les lignes en pointillé représenteraient une contrainte de dominance requise dans l’arbre qui est substitué en Du^\downarrow . La description d’une telle contrainte ne nous semble pas possible en TAG⁴. Il est alors naturel d’utiliser la même approche pour les adverbiaux en position initiale afin qu’ils apparaissent également dans leur clause hôte (sous le nœud S , voir la figure 5(b)).

3 Grammaires Catégorielles Abstraites

Les ACG appartiennent à la famille des grammaires de types logiques. Plutôt qu’un formalisme grammatical propre, elles offrent un cadre grammatical dans lequel différents formalismes grammaticaux peuvent être encodés (de Groote & Pogodalla, 2004), notamment les TAG (de Groote, 2002). La définition des ACG s’appuie sur un petit nombre d’opérations de la théorie des types et du λ -calcul. Ces opérations se combinent à l’aide de règles simples de composition, offrant aux ACG une grande souplesse. En particulier, les ACG engendrent des langages de λ -termes linéaires, généralisant à la fois les langages de chaînes de caractères et les langages d’arbres.

Une caractéristique des ACG est de considérer les structures d’analyse des grammaires, la *langage abstrait*, de manière explicite plutôt que comme produit dérivé (ainsi les arbres de dérivation des grammaires non contextuelles ne sont-ils pas définis de manière intrinsèque. D’une manière similaire pour les TAG, les arbres de dérivation résultent de l’analyse mais ne définissent pas les langages engendrés). Ces structures sont ensuite interprétées à l’aide d’un *lexique* pour obtenir le *langage objet* des formes de surfaces. On appelle *vocabulaire* les *signatures d’ordre supérieur* définissant les éléments atomiques des langages (types atomiques à partir desquels sont construits inductivement les types implicatifs $\alpha \rightarrow \beta$ et constantes à partir desquelles sont construits les λ -termes). Étant donné un tel vocabulaire Σ , l’ensemble des λ -termes typés construits sur la signature est $\Lambda(\Sigma)$. Pour une ACG \mathcal{G} de lexique \mathcal{L} , on notera indifféremment $\mathcal{G}(a) = o$, $\mathcal{L}(a) = o$, ou $a := o$ l’interprétation du terme (resp. du type) abstrait a par le terme (resp. le type) objet o . On appelle *analyse ACG* l’opération qui permet de retrouver les structures (termes) abstraites a à partir d’un terme objet o . Cette opération consiste

4. C’est possible a priori avec les D-Tree Substitution Grammars (Rambow et al., 2001), mais les dépendances indiquées dans l’arbre de dérivation seraient différentes, l’interface syntaxe-sémantique des grammaires synchrones serait à définir, et les propriétés de réversibilité à établir.

à trouver le ou les antécédents a de o par le lexique (inversion du lexique). En adoptant ce point de vue, les arbres de dérivation des TAG sont représentés par des termes d'un langage abstrait, alors que les arbres dérivés ou le *yield* (la chaîne de caractères qui s'obtient en lisant les feuilles d'un arbre dérivé) sont représentés par des termes de différents langages objets. Il s'agit d'un langage objet d'arbres pour la représentation des arbres dérivés, et d'un langage objet de chaînes pour le *yield*. La classe des *ACG de second ordre*⁵ permet des analyses polynomiales dont les bornes de complexité correspondent aux meilleures connues (Kanazawa, 2008).

En définissant la manière dont les structures abstraites sont interprétées, le lexique joue un rôle crucial dans notre proposition. Tout d'abord, on peut noter qu'il est possible que deux interprétations partagent un même vocabulaire abstrait. Ainsi, un même terme peut être projeté sur deux structures objets différentes reposant sur deux vocabulaires objets différents, typiquement une forme de surface (chaîne de caractères) et une forme sémantique (formule logique). Cette *composition* permet par exemple l'interprétation sémantique des arbres de dérivation. Elle est illustrée par $\mathcal{G}_{derived\ trees}$ et $\mathcal{G}_{TAG\ sem.}$ qui partagent le vocabulaire Σ_{TAG} dans la figure 6. Ensuite, un autre mode de composition, correspondant à la composition fonctionnelle, est possible. Il s'agit de faire interpréter une deuxième fois, par un deuxième lexique, une structure intermédiaire résultant d'une première interprétation. Cette fois-ci, c'est le vocabulaire objet d'une première ACG qui est aussi le vocabulaire abstrait de la deuxième ACG. Cette composition est illustrée par $\mathcal{G}_{yield} \circ \mathcal{G}_{derived\ trees}$ à la figure 6. Nous utilisons cette composition pour relier les arbres de dérivation du niveau discursif aux arbres de dérivation phrastiques par le biais de $\mathcal{G}_{disc-clause\ int.}$, et, par la suite, aux arbres dérivés et aux chaînes de caractères.

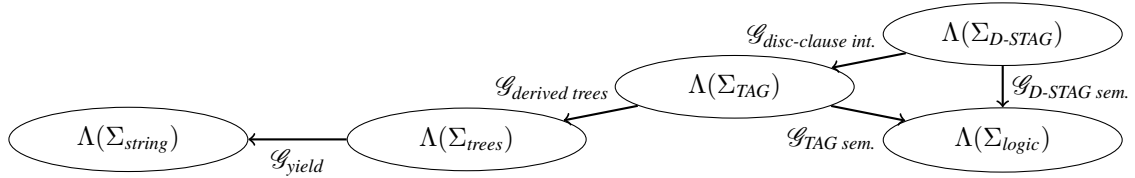


FIGURE 6 – Architecture ACG pour l'interface entre grammaire phrastique et grammaire discursive

4 Encodages des grammaires

TAG dans les ACG Pour encoder une TAG dans une ACG⁶, nous utilisons une signature d'ordre supérieur Σ_{TAG} dont les types atomiques comprennent les types suivant : $S, V, NP, S_A, V_A, \dots$ pour tout symbole non terminal X de la grammaire TAG qui peut faire l'objet d'une adjonction ou d'une substitution. Les types X sont utilisés pour les substitutions (soit pour le type de l'arbre qui est substitué, c'est-à-dire le type de sa racine, soit pour le type paramètre représentant un site de substitution), et les types X_A sont utilisés pour les adjonctions (soit pour le type de l'arbre auxiliaire de racine et nœud pied X , soit pour le type paramètre indiquant qu'une adjonction est possible). Pour chaque arbre élémentaire $\gamma_{entrée\ lex.}$, il y a une constante $C_{entrée\ lex.}$ dont le type dépend des sites de substitution et d'adjonction au sein de $\gamma_{entrée\ lex.}$ comme le montre la table 1 donnant l'encodage dans Σ_{TAG} des arbres de la figure 7⁷. Cette signature comprend également pour chaque type X_A une constante $I_X : X_A$ utilisée pour représenter une adjonction factice, c'est-à-dire lorsque que dans une dérivation TAG, aucune adjonction ne se produit sur un nœud X alors que ce dernier pourrait recevoir une adjonction.

Les termes construits sur cette signature sont interprétés par $\mathcal{G}_{derived\ trees}$ dans une autre signature dont le seul type atomique est le type τ des arbres. Dans cette signature, pour chaque symbole X d'arité n du vocabulaire permettant la construction des arbres, il y a une constante $X_n : \overbrace{\tau \rightarrow \dots \rightarrow \tau}^{n\ \text{fois}} \rightarrow \tau$. Nous ne décrivons pas ici l'interprétation définie par \mathcal{G}_{yield} qui interprète directement les arbres avec leur *yield*. Elle consiste simplement à interpréter les arbres comme des chaînes de caractères et chaque constante n -aire par la concaténation de ses n arguments.

Le lexique de la table 1 permet par exemple d'interpréter comme des chaînes de caractères deux dérivations avec une

5. L'ordre d'un type atomique a est $ord(a) = 1$, l'ordre d'un type complexe est $ord(\alpha \rightarrow \beta) = \max(ord(\alpha) + 1, ord(\beta))$. L'ordre d'un terme est l'ordre de son type, et l'ordre d'une ACG est l'ordre maximum de ses constantes abstraites. Dans une ACG d'ordre 2, tous les arguments des constantes sont des termes de type atomique, et pas des fonctions.

6. Pour une explication plus détaillée de cet encodage, ainsi que de l'interface syntaxe-sémantique, nous renvoyons le lecteur à (de Groot, 2001, 2002; Pogodalla, 2009).

7. Une constante $C_{entrée\ lex.}$ a comme type résultat le type X de la racine de $\gamma_{entrée\ lex.}$ si ce dernier est un arbre initial, et X_A si c'est un arbre auxiliaire. Comme paramètres, chaque site d'adjonction de label Y (y compris le nœud racine le cas échéant) introduit un paramètre de type Y_A et chaque site de substitution de label Y introduit un paramètre de type Y . Par convention, l'ordre des paramètres est donné en parcourant en premier les sites d'adjonction, avec un parcours en largeur, puis les sites de substitution, avec un parcours de gauche à droite.

adjonction au nœud S et au nœud V, ainsi que le montrent les équations (6) et (7). À noter que les types X de Σ_{TAG} sont interprétés comme des arbres (voir par exemple C_{Fred} et γ_{Fred}) alors que les types X_A sont interprétés comme des fonctions des arbres dans les arbres, permettant la modification du sous-arbre auquel elles sont appliquées.

$$(6) \quad t_1 = C_{allé\ à} (C_{ensuite}^S I_S) (C_{être\ aux.} I_V) C_{Fred} C_{Paris}$$

$$\mathcal{G}_{yield} \circ \mathcal{G}_{derived\ trees}(t_1) = ensuite+, +Fred + est + allé + à + Paris$$

$$(7) \quad t_2 = C_{allé\ à} I_S (C_{être\ aux.} (C_{ensuite}^V I_V)) C_{Fred} C_{Paris}$$

$$\mathcal{G}_{yield} \circ \mathcal{G}_{derived\ trees}(t_2) = Fred + est + ensuite + allé + à + Paris$$

Si l'on représente les termes t_1 et t_2 comme des arbres (un nœud étant étiqueté par le foncteur et ses fils par ses arguments qui sont éventuellement d'autres arbres), on obtient les arbres de la figure 8. Ceux-ci sont tout à fait similaires aux arbres de dérivation TAG à ceci près que toutes les adjonctions sont représentées (éventuellement par l'adjonction de l'identité I_X) et que l'ordre des paramètres est fixe, correspondant à chacun des sites, plutôt que variable avec l'adresse de Gorn du site concerné indiquée sur l'arête. L'arbre de gauche correspond aux adjonctions de l'adverbe et de l'auxiliaire sur le verbe, et l'arbre de droite correspond à une adjonction de l'adverbe sur l'auxiliaire, le tout étant adjoint au verbe.

Remarque. La notion de forme de surface f qui ancre un arbre γ_f en TAG correspond au fait que la constante objet f est un sous-terme de l'interprétation d'une constante abstraite C_f (par exemple « ensuite » ancre $C_{ensuite}^S$ car c'est un sous-terme de $\mathcal{G}_{yield} \circ \mathcal{G}_{derived\ trees}(C_{ensuite}^S) = \lambda S x.S(ensuite + x)$).

Constantes et types de Σ_{TAG}	Leur interprétation par $\mathcal{G}_{derived\ trees}$
S, V, NP...	τ
$S_A, V_A, NP_A \dots$	$\tau \rightarrow \tau$
$C_{Fred} : NP$	$\gamma_{Fred} : \tau$ $\gamma_{Fred} = NP_1 Fred$
$C_{être\ aux.} : V_A \rightarrow V_A$	$\gamma_{est} : (\tau \rightarrow \tau) \rightarrow \tau \rightarrow \tau$ $\gamma_{est} = \lambda A x.V_2(A(V_1 est)) x$
$C_{allé\ à} : S_A \rightarrow V_A \rightarrow NP \rightarrow NP \rightarrow S$	$\gamma_{allé\ à} : (\tau \rightarrow \tau) \rightarrow (\tau \rightarrow \tau) \rightarrow \tau \rightarrow \tau \rightarrow \tau$ $\gamma_{allé\ à} = \lambda S A s c.S(S_2 s (VP_2(A(V_1 allé)) (PP_2(Prep_1 à) c)))$
$C_{ensuite}^S : S_A \rightarrow S_A$	$\gamma_{ensuite}^S : (\tau \rightarrow \tau) \rightarrow \tau \rightarrow \tau$ $\gamma_{ensuite}^S = \lambda A x.A(S_2(Adv_1 ensuite) x)$
$C_{ensuite}^V : V_A \rightarrow V_A$	$\gamma_{ensuite}^V : (\tau \rightarrow \tau) \rightarrow \tau \rightarrow \tau$ $\gamma_{ensuite}^V = \lambda A x.A(V_2 x(Adv_1 ensuite))$

TABLE 1 – Exemple de lexique pour l'encodage de la grammaire TAG de la figure 7

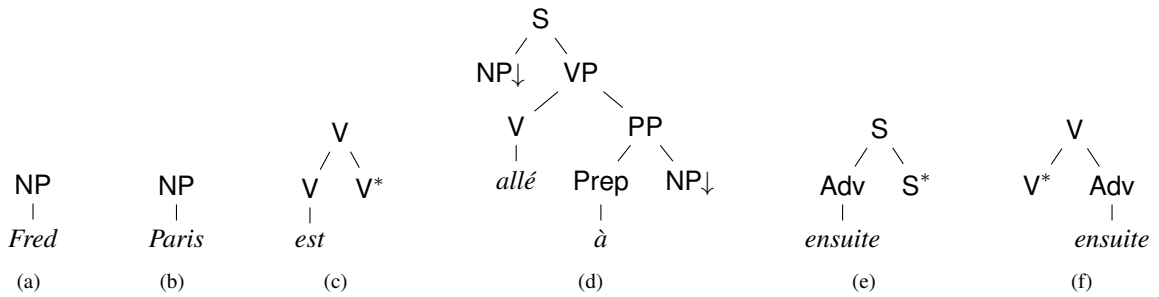


FIGURE 7 – Exemples d'arbres de la grammaire phrastique TAG

D-STAG dans les ACG L'encodage de D-STAG dans les ACG suit les mêmes principes que ceux énoncés dans le paragraphe précédent. Les arbres dérivés et de dérivation correspondent maintenant aux arbres de la grammaire de discours de la figure 4. On obtient donc l'équivalent des arbres de dérivation de la grammaire D-STAG. Les différences principales par rapport à l'encodage TAG apparaissent dans les interprétations :

- $\mathcal{G}_{disc-clause\ int.}$ implante l'interface entre la grammaire discursive et la grammaire phrastique en évitant, par simple composition, l'étape intermédiaire d'extraction et de construction de la forme normale de discours.

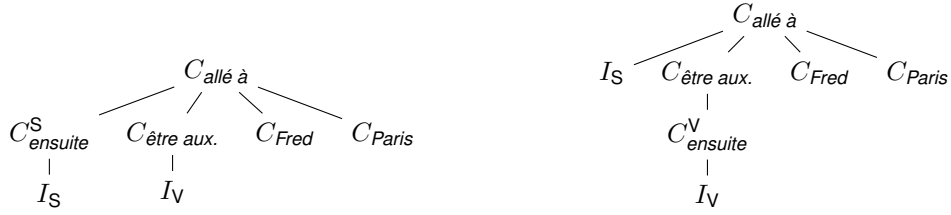


FIGURE 8 – Termes abstraits et arbres de dérivation TAG

- $\mathcal{G}_{TAG\ sem.}$ implante l’interprétation des structures de discours. Par rapport à (Danlos, 2011), afin d’avoir un système plus unifié, tous les discours sont interprétés avec un ordre supérieur. De plus, nous introduisons un nouveau type atomique ℓ pour les labels. De cette façon, nous pouvons construire des formules comme $l_1 : \text{grumpy}(F) \wedge \exists x. l_2 : \text{keys}(x) \wedge l_3 : \text{lose}(F, x) \wedge l_3 : \phi_{\text{Expl.}}(l_1, l_2)$ ⁸. De cette manière, ce sont les quantificateurs qui ont portée sur les relations de discours, plutôt que l’inverse comme dans une formule du type $\phi_{\text{Expl.}}(\text{grumpy}(F), \exists x. \text{keys}(x) \wedge \text{lose}(F, x))$.

5 Interface entre grammaires phrastiques et discursives

Le vocabulaire abstrait Σ_{D-STAG} utilise les mêmes types atomiques que Σ_{TAG} pour la grammaire phrastique (NP, V, V_A etc.), ainsi que de nouveaux types atomiques propres à la description du niveau discursif : Du, le type pour les unités de discours, et le type Du_A correspondant pour les sites d’adjonction. Une constante abstraite de Σ_{D-STAG} introduisant un marqueur discursif comme $d_{\text{parce que}}$ a typiquement le type $Du_A \rightarrow Du_A \rightarrow Du_A \rightarrow Du \rightarrow Du_A$ (que l’on notera par définition $DC \triangleq Du_A \rightarrow Du_A \rightarrow Du_A \rightarrow Du \rightarrow Du_A$). Ce type reflète la forme des arbres auxiliaires de D-STAG de la figure 4. Nous utilisons également un type T pour un texte (par opposition à phrase).

Le cœur de notre proposition réside dans l’ancrage de l’arbre, ou du terme, d’un marqueur de discours $d_{\text{conn.}}$ de type DC non pas par la chaîne de caractère *conn.* mais par l’arbre auxiliaire correspondant $C_{\text{conn.}}$ (voir la remarque de la section 4) par l’intermédiaire de l’ACG $\mathcal{G}_{\text{disc-clause int.}}$. Cela nous permet de spécifier que, dans l’arbre qui est substitué (l’argument de type Du qui est paramètre de $d_{\text{conn.}}$, ou, pour le dire dans les termes de D-STAG, celui qui est substitué au nœud Du_{\downarrow}), il faut qu’il y ait également adjonction de l’arbre $C_{\text{conn.}}$ au niveau phrastique. Cela se traduit par exemple dans $\mathcal{G}_{\text{disc-clause int.}}$ (voir table 2) par le fait que l’interprétation du terme abstrait d_{ensuite}^V fait appel à une adjonction sur son paramètre s de l’arbre auxiliaire C_{ensuite}^V . C’est ainsi que l’arbre discursif d_{ensuite}^V est relié à l’arbre phrastique C_{ensuite}^V de l’adverbe en position médiale. On rend ainsi compte de la contrainte de dominance requise (voir figure 5) dans l’arbre qui est substitué en Du_{\downarrow} . Cette contrainte, qui n’est pas à notre connaissance exprimable en TAG, l’est naturellement avec les ACG.

Pour permettre cette adjonction, le type Du des unités de discours est interprété comme une clause qui peut encore être l’objet d’une adjonction d’un arbre auxiliaire de conjonction de subordination ou d’un adverbial en position initiale ou médiale. Cela revient à l’interpréter comme un type de second ordre $S_A \rightarrow V_A \rightarrow S$ où des adjonctions sur les nœuds S et V sont encore possibles⁹. Plus précisément, nous interprétons Du comme $S_A \rightarrow (V_A \rightarrow V_A) \rightarrow S$ pour rendre compte de la possibilité de ce que les adverbiaux médiaux peuvent être entre d’autres adverbes¹⁰. Dès lors, au niveau de la grammaire discursive, les verbes intransitifs par exemple auront le type $S_A \rightarrow V_A \rightarrow V_A \rightarrow NP \rightarrow S$ plutôt que $S_A \rightarrow V_A \rightarrow NP \rightarrow S$ pour permettre l’adjonction au-dessus ou au-dessous du marqueur discursif adverbial. On obtient alors les types et les interprétations décrites dans la table 2¹¹. Bien qu’un même symbole (S par exemple) puisse apparaître aussi bien à gauche qu’à droite du signe $:=$, il est important de noter que le symbole de gauche appartient au

8. La notation $l; p$ se lit l est un label de p . Un même label peut étiqueter plusieurs propositions. Cela signifie que ces dernières appartiennent à une même clause. Par ailleurs, les prédicats des formules sémantiques sont donnés en anglais pour les distinguer de la partie syntaxiques en français.

9. Une autre solution serait de définir $DC \triangleq Du_A \rightarrow Du_A \rightarrow Du_A \rightarrow (S_A \rightarrow V_A \rightarrow Du) \rightarrow Du_A$ avec un type fonctionnel comme quatrième paramètre, au lieu de Du. Mais l’ACG ne serait plus de second ordre. L’analyse ACG resterait décidable car la grammaire serait toujours lexicalisée pour les constantes d’ordre supérieur, et donc inversible. Cependant, la complexité d’analyse de ces grammaires n’est en général pas polynomiale.

10. Une analyse plus précise pour vérifier que cette configuration existe en français devra être menée. Cela généralise en tout cas la possibilité pour l’adverbe marqueur de discours de se trouver aussi bien avant qu’après un autre adverbe (comme les occurrences de *généralement ensuite* ou de *ensuite doucement* sur le web semblent l’indiquer) et permet également de traiter les auxiliaires par une adjonction (*a ensuite doucement...*). Bien entendu, il faudrait faire de même pour les adverbiaux en position initiale, mais pour des raisons de concision nous ne le faisons pas ici.

11. *mod* et *cons* sont deux opérateurs qui ne servent qu’à juxtaposer les arbres TAG de dérivation phrastique des unités de discours élémentaires. Ils sont interprétés de la manière suivante : $\text{mod} := \lambda s\ m.\ m\ s$ (c’est-à-dire qu’il réalise l’adjonction effective sur l’arbre dérivé) et $\text{cons} := \lambda s_1\ s_2\ s_3\ s.\ s_1(s_2(S_3\ x.\ (s_3\ s)))$ qui construit un arbre dérivé en insérant un point (quand c’est nécessaire) entre les arbres dérivés correspondant aux unités élémentaires de discours.

vocabulaire abstrait Σ_{D-STAG} tandis que les types ou termes de droite sont construits sur le vocabulaire objet Σ_{TAG} .

Si d'un point de vue théorique la complexité de l'analyse augmente à cause de l'interprétation de types atomiques par des types d'ordre plus élevés, l'étude précise de la complexité reste à faire. La complexité est polynomiale puisque les ACG de second ordre sont équivalentes aux grammaires non contextuelles multiples (MCFG) et aux systèmes de réécriture linéaires (m -LCFRS). Mais d'autres propriétés comme le bon parenthésage (*well-nestedness*), le rang, ou l'étendue (*fan-out*) de la nouvelle grammaire pourrait permettre de raffiner ce résultat.

	NP	:= NP	N	:= N	Du	:= $S_A \rightarrow (V_A \rightarrow V_A) \rightarrow S$
	NP _A	:= NP _A	N _A	:= N _A	Du _A	:= S _A
	V	:= V	T	:= S	S	:= $S_A \rightarrow (V_A \rightarrow V_A) \rightarrow S$
	V _A	:= $V_A \rightarrow V_A$			S _A	:= $S_A \rightarrow S_A$
I_X	:	X_A				:= $\lambda P.P$
d_{Fred}	:	NP				:= C_{Fred}
$d_{\text{être aux.}}$:	$V_A \rightarrow V_A$:= $\lambda a v.C_{\text{être aux.}}(a v)$
$d_{\text{allé à}}$:	$S_A \rightarrow V_A \rightarrow V_A \rightarrow NP \rightarrow NP \rightarrow S$:= $\lambda S a_1 a_2 s o c m.C_{\text{allé à}}(S c)(a_2(m(a_1 I_V))) s o$
$d_{\text{initial anchor}}$:	$S \rightarrow Du_A \rightarrow T$:= $\lambda s m.mod(s I_S(\lambda x.x)) m$
d_{anchor}	:	$S \rightarrow Du_A \rightarrow Du$:= $\lambda s m d_s d_v.mod(s d_s d_v) m$
d_{ensuite}^S	:	DC				:= $\lambda d_1 d_2 d_3 s.cons d_1 d_2 d_3 (s (C_{\text{ensuite}}^S I_S)(\lambda x.x))$
d_{ensuite}^V	:	DC				:= $\lambda d_1 d_2 d_3 s.cons d_1 d_2 d_3 (s I_S C_{\text{ensuite}}^V)$

TABLE 2 – Interprétation par $\mathcal{G}_{\text{disc-clause int.}}$ pour l'interface entre phrase et discours

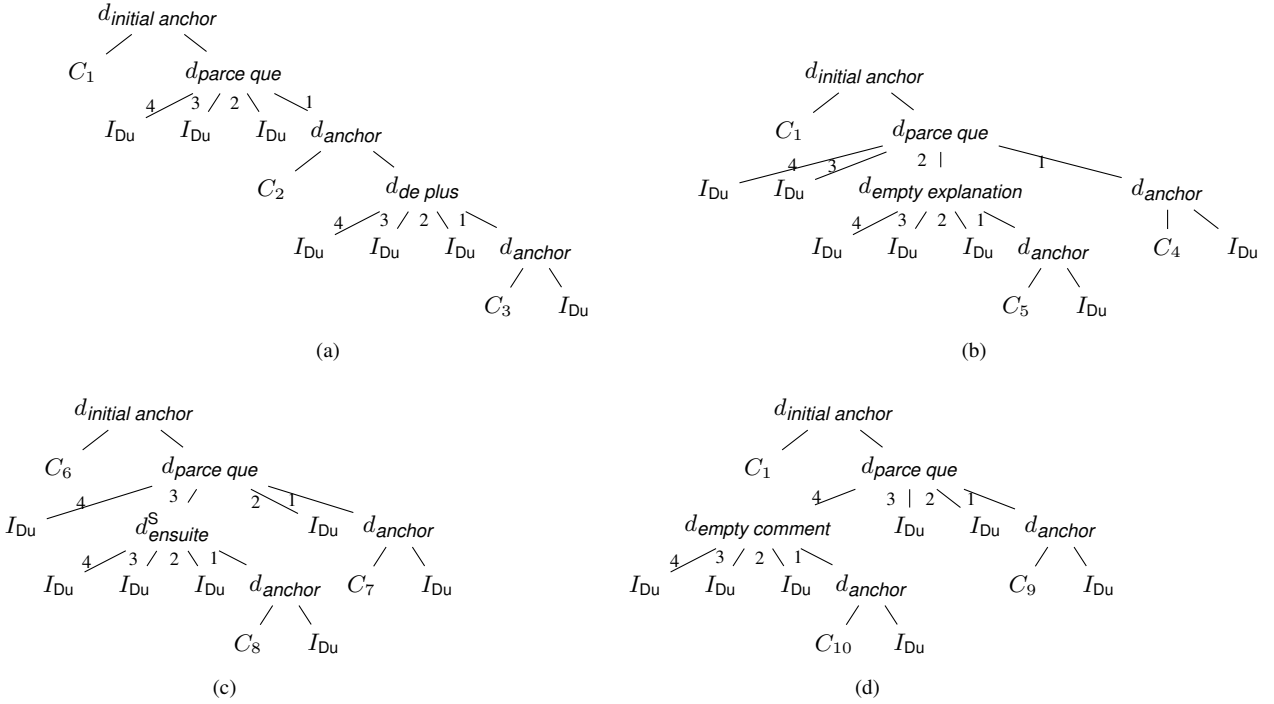


FIGURE 9 – Arbres de dérivation pour le discours. $\mathcal{G}_{D-STAG \text{ sem.}}$ les interprète, le cas échéant, en DAG (voir (21))

Nous ne pouvons donner ici la définition de tous les termes C_i qui définissent les arbres de dérivation discursifs pour chacune des clauses correspondant à l'unité discursive i dans les exemples (1–4) car il faudrait définir le lexique entier¹². Nous ne donnons que C_8'' en (8) (qui diffère de C_8 seulement par le complément de lieu, ici un nom propre). On voit qu'en appliquant le résultat à $(C_{\text{ensuite}}^S I_S)$ et $(\lambda x.x)$ (resp. à I_S et C_{ensuite}^V), c'est-à-dire en remplaçant adv_s et adv_v , comme le fait d_{ensuite}^S (resp. d_{ensuite}^V), on obtient le terme t_1 donné en (6) (resp. t_2 donné en (7)).

$$(8) \quad \begin{aligned} C_8'' &= d_{\text{allé à}} I_S I_V (d_{\text{être aux.}} I_V) d_{\text{Fred}} d_{\text{paris}} \\ &:= \lambda^0 adv_s adv_v.C_{\text{allé à}} adv_s (C_{\text{être aux.}}(adv_v I_V)) C_{\text{Fred}} C_{\text{paris}} \end{aligned}$$

12. Cela est fait dans les fichiers d'exemples.

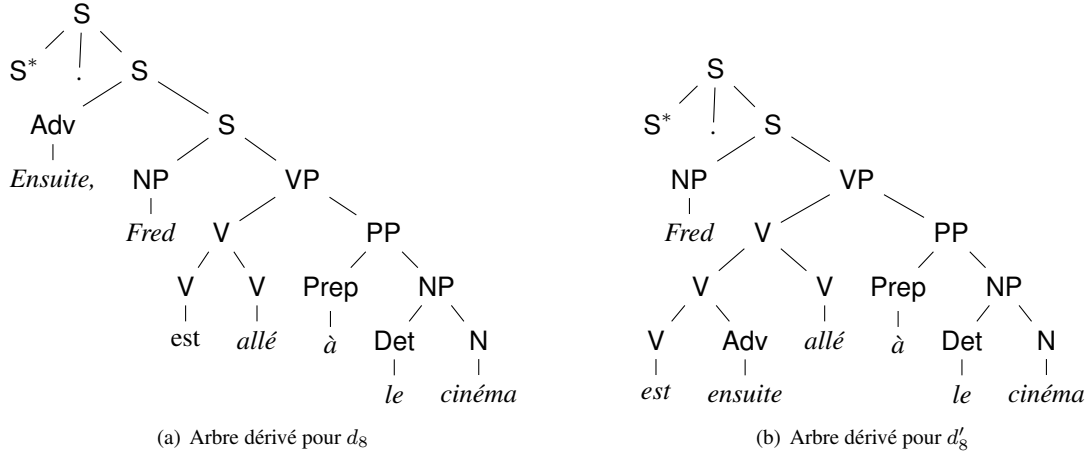


FIGURE 10 – Interprétation comme arbres dérivés de d_8 et d'_8

À l'aide des lexiques définis dans les tables 1 et 2, et avec les définitions (9) and (10), on peut vérifier que l'on obtient les interprétations (11)–(16). Le terme d_8 (resp d'_8) correspond à l'arbre auxiliaire discursif (comme le montre le type Du_A) qui étend le discours précédent par *Ensuite, il est allé au cinéma* (resp. *Il est ensuite allé au cinéma*).

- (9) $d_8 = d_{ensuite}^S I_{Du} I_{Du} I_{Du} (d_{anchor} C_8 I_{Du}) : Du_A$
- (10) $d'_8 = d_{ensuite}^V I_{Du} I_{Du} I_{Du} (d_{anchor} C_8 I_{Du}) : Du_A$
- (11) $\mathcal{G}_{disc-clause\ int.}(d_8) = \text{cons } I_S I_S I_S (\text{mod } (C_{allé\ à} (C_{ensuite}^S I_S) (C_{être\ aux.} I_V) C_{Fred} (C_{le} (C_{cinéma} I_N))) I_S) : S_A$
- (12) $\mathcal{G}_{derived\ trees} \circ \mathcal{G}_{disc-clause\ int.}(d_8) = [\text{voir la représentation figure 10(a)}]$
- (13) $\mathcal{G}_{yield} \circ \mathcal{G}_{derived\ trees} \circ \mathcal{G}_{disc-clause\ int.}(d_8) = \lambda x.x + . + \textit{Ensuite} + , + \textit{Fred} + \textit{est} + \textit{allé} + \textit{à} + \textit{le} + \textit{cinéma} : \sigma \rightarrow \sigma$
- (14) $\mathcal{G}_{disc-clause\ int.}(d'_8) = \text{cons } I_S I_S I_S (\text{mod } (C_{allé\ à} I_S (C_{être\ aux.} (C_{ensuite}^V I_V)) C_{Fred} (C_{le} (C_{cinéma} I_N))) I_S) : S_A$
- (15) $\mathcal{G}_{derived\ trees} \circ \mathcal{G}_{disc-clause\ int.}(d'_8) = [\text{voir la représentation figure 10(b)}]$
- (16) $\mathcal{G}_{yield} \circ \mathcal{G}_{derived\ trees} \circ \mathcal{G}_{disc-clause\ int.}(d'_8) = \lambda x.x + . + \textit{Fred} + \textit{est} + \textit{ensuite} + \textit{allé} + \textit{à} + \textit{le} + \textit{cinéma} : \sigma \rightarrow \sigma$

De la même manière, les arbres de dérivation discursifs de la figure 9¹³, correspondant aux analyses des exemples (1–4), sont donnés par les termes d_1, d_2, d_3 et d_4 des équations (17)–(20) ($d_{empty\ explanation}$ et $d_{empty\ comment}$ ancrent tous les deux le connecteur vide, mais avec la sémantique *Explanation* pour le premier, et *Comment* pour le second).

- (17) $d_1 = d_{initial\ anchor} C_1 (d_{parce\ que} I_{Du} I_{Du} I_{Du} (d_{anchor} C_2 (d_{de\ plus} I_{Du} I_{Du} I_{Du} (d_{anchor} C_3 I_{Du}))))$
- (18) $d_2 = d_{initial\ anchor} C_1 (d_{parce\ que} I_{Du} I_{Du} (d_{empty\ explanation} I_{Du} I_{Du} I_{Du} (d_{anchor} C_5 I_{Du})) (d_{anchor} C_4 I_{Du}))$
- (19) $d_3 = d_{initial\ anchor} C_6 (d_{parce\ que} I_{Du} (d_{ensuite}^S I_{Du} I_{Du} I_{Du} (d_{anchor} C_8 I_{Du})) I_{Du} (d_{anchor} C_7 I_{Du}))$
- (20) $d_4 = d_{initial\ anchor} C_1 (d_{parce\ que} (d_{empty\ comment} I_{Du} I_{Du} I_{Du} (d_{anchor} C_{10} I_{Du})) I_{Du} I_{Du} (d_{anchor} C_9 I_{Du}))$

6 Sémantique

Pour l'interprétation sémantique, nous avons modifié les types (et par conséquent les formules) proposés dans (Danlos, 2009). Ainsi, en plus des types e pour les entités, t pour les valeurs de vérité, et ℓ pour les labels, nous définissons $qnp \triangleq (e \rightarrow \ell \rightarrow t) \rightarrow \ell \rightarrow t$ et $ltt \triangleq (\ell \rightarrow t) \rightarrow t$. Nous utilisons les connecteurs logiques habituels, et les constantes sémantiques sont données en anglais. La table 3 montre quelques interprétations¹⁴. L'objet de cet article n'étant pas de montrer en quoi les formules de (Danlos, 2009, 2011) fonctionnent, nous laissons le lecteur vérifier, éventuellement à

13. On a ajouté sur les arêtes les numéros de sites d'adjonction ou de substitution des arbres élémentaires des connecteurs de la figure 4.

14. Notons que $d_{ensuite}^S$ et $d_{ensuite}^V$ ont maintenant la même interprétation.

l'aide des fichiers exemples, que l'égalité (21) est correcte. Elle correspond à la structure discursive de la figure 1(c) : les propositions étiquetées par l_6 (resp. l_7 et l_8) correspondent à la clause C_6 (resp. C_7 et C_8), et $l_{\mathcal{R}}$ et $l_{\mathcal{R}'}$ aux relations \mathcal{R} et \mathcal{R}' . Cela décrit bien un DAG. Nous ne montrons pas ici les trois autres dérivations qui peuvent également être interprétées par $\mathcal{G}_{D-STAG sem.}$ avec des formules représentant les trois autres structures de la figure 1.

$$(21) \quad \mathcal{G}_{D-STAG sem.}(d_3) = \exists \ell l_6 l_{\mathcal{R}} l_{\mathcal{R}'}. \exists ! x. l_6: \text{supermarket}(x) \wedge l_6: \text{go_to}(\text{fred}, x) \wedge \\ (\exists \ell l_8. (\exists ! x. l_8: \text{movies}(x) \wedge l_8: \text{go_to}(\text{fred}, x)) \wedge ((\exists \ell l_7. (\exists ! x. l_7: \text{fridge}(x) \wedge \\ l_7: \text{empty}(x)) \wedge (l_{\mathcal{R}}: \phi_{\text{Expl.}}(l_6, l_7) \wedge \top)) \wedge l_{\mathcal{R}'}: \phi_{\text{Nar}}(l_6, l_8))))$$

NP	:= qnp	N	:= e → ℓ → t	Du	:= ℓ tt
NP _A	:= qnp → qnp	N _A	:= (e → ℓ → t) → e → ℓ → t	Du _A	:= ℓ tt → ℓ tt
T	:= t	V _A	:= t → t	S	:= (t → t) → ℓ → t
				S _A	:= t → t
I_X	:= λP.P				
d_{Fred}	:= λP ℓ.P fred ℓ				
$d_{\text{allé à}}$:= λS a ₁ a ₂ s o m ℓ.S (s (λx ℓ ₁ .o (λy ℓ ₂ .a ₂ (m (a ₁ (ℓ ₂ :go_to x y)))) ℓ ₁) ℓ)				
$d_{\text{initial anchor}}$:= λs m.∃ℓℓ.m (λQ.(s (λx.x) ℓ) ∧ (Q ℓ)) (λℓ'.T)				
d_{anchor}	:= λs m P.∃ℓℓ.m (λQ.(s (λx.x) ℓ) ∧ (Q ℓ)) P				
d_{ensuite}^S	:= λd ₄ d ₃ d ₂ s f.d ₄ (λP.∃ℓℓ.d ₃ f (λx.d ₂ s (λy.(P x) ∧ (ℓ:φ _{Nar} (x, y))))))				
d_{ensuite}^V	:= λd ₄ d ₃ d ₂ s f.d ₄ (λP.∃ℓℓ.d ₃ f (λx.d ₂ s (λy.(P x) ∧ (ℓ:φ _{Nar} (x, y))))))				

TABLE 3 – Interprétation par $\mathcal{G}_{D-STAG sem.}$ pour la sémantique du discours

7 Conclusion

Cet article montre comment interfacer deux grammaires phrastiques et discursives fondées sur les TAG sans recourir à un découpage du processus d'analyse. Cette approche est suffisamment générale pour permettre d'engendrer des structures de graphe qui ne soient pas de simples arbres. Nous avons aussi montré comment implanter cette interface avec des grammaires ACG pour associer formes de surface et représentations sémantiques pour des textes de plusieurs phrases. Ainsi, nous bénéficions d'un mécanisme autorisant l'analyse et la génération prenant en compte les relations de discours. Nous n'avons pas exposé ici comment il est possible de modéliser les modifications des connecteurs (... *probablement parce qu'il pleut*). Notre travail futur portera sur les connecteurs multiples (... *parce qu'ensuite il s'est rendu compte qu'il était fauché*) et sur la manière d'intégrer des modèles de désambiguïsation phrastiques et discursifs pour tirer partie de l'analyse en une seule phase. Enfin, nous pensons que l'utilisation d'une même grammaire permettra de mieux modéliser les problèmes à l'interface entre syntaxe et sémantique lorsqu'on prend en compte les verbes d'attitude propositionnelle et leur position dans la phrase (Danlos, 2013).

Références

- ASHER N. & LASCARIDES A. (2003). *Logics of conversation*. Cambridge University Press.
- DANLOS L. (2009). D-STAG : un formalisme d'analyse automatique de discours basé sur les TAG synchrones. *T.A.L.*, 50(1), 111–143. <http://hal.inria.fr/inria-00524743/en/>.
- DANLOS L. (2011). D-STAG : a formalism for discourse analysis based on SDRT and using Synchronous TAG. In P. DE GROOTE, M. EGG & L. KALLMEYER, Eds., *14th conference on Formal Grammar - FG 2009*, volume 5591 of *LNCS/LNAI*, p. 64–84 : Springer. doi :10.1007/978-3-642-20169-1_5.
- DANLOS L. (2013). Connecteurs de discours adverbiaux : Problèmes à l'interface syntaxe-sémantique. *Linguistica Investigaciones*, 36(2), 261–275. doi :10.1075/li.36.2.05dan.
- DE GROOTE P. (2001). Towards Abstract Categorical Grammars. In *Association for Computational Linguistics, 39th Annual Meeting and 10th Conference of the European Chapter, Proceedings of the Conference*, p. 148–155. <http://aclweb.org/anthology/P01-1033>.
- DE GROOTE P. (2002). Tree-Adjoining Grammars as Abstract Categorical Grammars. In *Proceedings of the Sixth International Workshop on Tree Adjoining Grammars and Related Frameworks (TAG+6)*, p. 145–150 : Università di Venezia. <http://www.loria.fr/equipes/calligramme/acg/publications/2002-tag+6.pdf>.

- DE GROOTE P. & POGODALLA S. (2004). On the expressive power of Abstract Categorical Grammars : Representing context-free formalisms. *Journal of Logic, Language and Information*, **13**(4), 421–438. doi :[10.1007/s10849-004-2114-x](https://doi.org/10.1007/s10849-004-2114-x).
- FORBES K., MILTSAKAKI E., PRASAD R., SARKAR A., JOSHI A. K. & WEBBER B. L. (2003). D-LTAG system : Discourse parsing with a Lexicalized Tree-Adjoining Grammar. *Journal of Logic, Language and Information*, **12**(3), 261–279. doi :[10.1023/A:1024137719751](https://doi.org/10.1023/A:1024137719751). Special Issue : Discourse and Information Structure.
- FORBES-RILEY K., WEBBER B. L. & JOSHI A. K. (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](https://doi.org/10.1093/jos/ffh032).
- GARDENT C. (1997). *Discourse Tree Adjoining Grammar*. CLAUS Report 89, Universit, Saarbr. <ftp://ftp.coli.uni-sb.de/pub/coli/clus/clus89.ps>.
- JOSHI A. K., LEVY L. S. & TAKAHASHI M. (1975). Tree adjunct grammars. *Journal of Computer and System Sciences*, **10**(1), 136–163. doi :[10.1016/S0022-0000\(75\)80019-5](https://doi.org/10.1016/S0022-0000(75)80019-5).
- JOSHI A. K. & SCHABES Y. (1997). Tree-adjoining grammars. In G. ROZENBERG & A. K. SALOMAA, Eds., *Handbook of formal languages*, volume 3, chapter 2. Springer.
- KANAZAWA M. (2007). Parsing and generation as datalog queries. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics (ACL)*, p. 176–183, Prague, Czech Republic : Association for Computational Linguistics. <http://www.aclweb.org/anthology/P07-1023>.
- KANAZAWA M. (2008). A prefix-correct early recognizer for multiple context-free grammars. In *Proceedings of the Ninth International Workshop on Tree Adjoining Grammars and Related Formalisms (TAG+9)*, p. 49–56, Tuebingen, Germany. <http://tagplus9.cs.sfu.ca/papers/Kanazawa.pdf>.
- MANN W. C. & THOMPSON S. A. (1988). Rhetorical Structure Theory : Toward a functional theory of text organization. *Text*, **8**(3), 243–281. doi :[10.1515/text.1.1988.8.3.243](https://doi.org/10.1515/text.1.1988.8.3.243).
- MARCU D. (2000). *The Theory and Practice of Discourse Parsing and Summarization*. The MIT Press.
- NAKATSU C. & WHITE M. (2010). Generating with discourse combinatory categorial grammar. *Linguistic Issues in Language Technology*, **4**. <http://elanguage.net/journals/lilt/article/view/1277>.
- POGODALLA S. (2004). Computing Semantic Representation : Towards ACG Abstract Terms as Derivation Trees. In *Seventh International Workshop on Tree Adjoining Grammar and Related Formalisms - TAG+7*, p. 64–71, Vancouver, BC, Canada. <https://hal.inria.fr/inria-00107768>.
- POGODALLA S. (2009). Advances in Abstract Categorical Grammars : Language Theory and Linguistic Modeling. ESSLLI 2009 Lecture Notes, Part II. <https://hal.inria.fr/hal-00749297>.
- POLANYI L. & VAN DEN BERG M. H. (1996). Discourse structure and discourse interpretation. In P. J. E. DEKKER & M. STOKHOF, Eds., *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>.
- RAMBOW O., VIJAY-SHANKER K. & WEIR D. (2001). D-Tree Substitution Grammars. *Computational Linguistics*, **27**(1), 87–121. doi :[10.1162/089120101300346813](https://doi.org/10.1162/089120101300346813).
- SCHILDER F. (1997). Tree discourse grammar or how to get attached to a discourse? In *Proceedings of the Tilburg Conference on Formal Semantics (IWCS-1997)*, p. 261–273. <ftp://ftp.informatik.uni-hamburg.de/pub/unihh/informatik/WSV/schild97a.ps.gz>.
- SORICUT R. & MARCU D. (2003). Sentence level discourse parsing using syntactic and lexical information. In M. HEARST & M. OSTENDORF, Eds., *Proceedings of the 2003 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics (HLT-NAACL 2003)*, p. 149–156. <http://aclweb.org/anthology/N03-1030>.
- STEEDMAN M. (2001). *The Syntactic Process*. MIT Press.
- STEEDMAN M. & BALDRIDGE J. (2011). Combinatory categorial grammar. In R. BORSLEY & K. BÖRJARS, Eds., *Non-Transformational Syntax : Formal and Explicit Models of Grammar*, chapter 5. Wiley-Blackwell.
- WEBBER B. L. (2004). D-LTAG : extending lexicalized TAG to discourse. *Cognitive Science*, **28**(5), 751–779. doi :[10.1207/s15516709cog2805_6](https://doi.org/10.1207/s15516709cog2805_6).
- WEBBER B. L. & JOSHI A. K. (1998). Anchoring a lexicalized tree-adjoining grammar for discourse. In M. STEDE, L. WANNER & E. HOVY, Eds., *Proceedings of the ACL/COLING workshop on Discourse Relations and Discourse Markers*. <http://aclweb.org/anthology/W98-0315>.
- WEBBER B. L., STONE M., JOSHI A. K. & KNOTT A. (2003). Anaphora and discourse structure. *Computational Linguistics*, **29**(4), 545–587. <http://aclweb.org/anthology/J03-4002>.