

D-STAG : un formalisme d'analyse automatique de discours basé sur les TAG synchrones

Laurence Danlos

▶ **To cite this version:**

Laurence Danlos. D-STAG : un formalisme d'analyse automatique de discours basé sur les TAG synchrones. Revue TAL : traitement automatique des langues, 2009, 50 (1), pp.111–143. inria-00524743

HAL Id: inria-00524743
https://inria.hal.science/inria-00524743

Submitted on 11 Oct 2010

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

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

D-STAG : un formalisme d'analyse automatique de discours fondé sur les TAG synchrones

Laurence Danlos

ALPAGE
Université Paris 7
Institut Universitaire de France
30, rue Château des Rentiers, F-75013 Paris
Laurence.Danlos@linguist.jussieu.fr

RÉSUMÉ. *Nous proposons* D-STAG, *un nouveau formalisme pour l'analyse automatique de la structure discursive des textes. Les analyses produites par* D-STAG *sont des structures de discours hiérarchiques annotées de relations de discours, qui sont compatibles avec les structures de discours produites en* SDRT. *L'analyse discursive prolonge l'analyse phrastique, sans modifier celle-ci, ce qui rend envisageable la mise en œuvre d'un analyseur de discours.*

ABSTRACT. *We propose* D-STAG, *a new formalism for the automatic analysis of the discourse structure of texts. The analyses computed by* D-STAG *are hierarchical discourse structures annotated with discourse relations, that are compatible with discourse structures computed in* SDRT. *The discourse analysis extends the sententiall analysis, without modifying it, which makes conceivable the realization of a discourse analyzer.*

MOTS-CLÉS : *discours,* SDRT, *grammaires d'arbres adjoints (synchrones), syntaxe-sémantique*

KEYWORDS: *Discourse,* SDRT, *(Synchronous) Tree Adjoining Grammar, Syntax-Semantics*

1. Introduction

Nous proposons un nouveau formalisme d'analyse automatique de textes écrits, appelé D-STAG pour *Discourse Synchronous TAG*. Ce formalisme prolonge l'analyse phrastique au niveau discursif : il s'appuie sur un analyseur phrastique qui produit une analyse syntaxique et sémantique de chaque phrase composant le texte, et complète ces résultats par une analyse discursive. L'analyse discursive consiste à construire la « structure de discours » du texte donné en entrée. Une structure de discours repose sur des « relations de discours » (aussi appelées « relations rhétoriques ») qui relient des segments de discours – ou plus précisément les représentations sémantiques de ces segments de discours. Un discours n'est cohérent que si chaque élément d'information apporté dans le discours est relié rhétoriquement à un autre élément d'information, ce qui débouche sur une structure connexe pour le discours dans son ensemble.

Pour la partie discursive de notre analyseur de textes, nous nous sommes appuyée sur SDRT – *Segmented Discourse Representation Theory* (Asher, 1993 ; Asher et Lascarides, 2003) – qui est une des théories les plus élaborées pour le discours. Notre formalisme produit des structures de discours qui sont compatibles avec celles produites par SDRT. De ce fait, D-STAG peut bénéficier des résultats apportés par cette théorie du discours. Par exemple, D-STAG inclut l'implémentation de la contrainte de la frontière droite qui a été mise en avant en SDRT et qui limite grandement le nombre d'analyses possibles.

Les chercheurs travaillant dans le cadre de SDRT fournissent un travail théorique de grande qualité, mais n'ont pas concentré leurs efforts sur la question de l'implémentation robuste, efficace et à grande échelle d'un analyseur de textes calculant la structure d'un discours. Or, c'est notre objectif. Pour cet aspect du travail, nous nous sommes tournée vers un formalisme qui a un bon pouvoir expressif tout en restant efficace, à savoir TAG – *Tree Adjoining Grammar* (Joshi, 1985). TAG a d'abord été utilisé avec succès pour la réalisation d'analyseurs syntaxiques dans différentes langues. Ce formalisme a ensuite été étendu dans deux directions : passage de la syntaxe à la sémantique – avec entre autres STAG (Shieber, 1994 ; Shieber et Schabes, 1990 ; Nesson et Shieber, 2006) –, et passage du niveau phrastique au niveau discursif. Le passage au niveau discursif a concerné d'abord la génération automatique de textes – avec entre autres G-TAG (Danlos, 1998) –, puis l'analyse automatique de textes – avec entre autres D-LTAG (Forbes-Riley *et al.*, 2006). Le nouveau formalisme D-STAG que nous proposons aujourd'hui s'appuie bien évidemment sur tous ces travaux, en particulier sur D-LTAG qui a été conçu avec un objectif similaire. Ainsi D-STAG et D-LTAG reposent sur la même architecture avec trois modules :

1) un analyseur phrastique, qui produit pour chaque phrase du discours donné en entrée une analyse syntaxique et sémantique ;

2) une interface phrase-discours, qui est un module nécessaire si on veut (et c'est ce que nous voulons) ne rien changer à l'analyse phrastique ;

3) un analyseur discursif, qui calcule la structure du discours donné en entrée.

Cet article est organisé comme suit. La section 2 présente les prérequis linguistiques, théoriques et formels sur lesquels D-STAG s'appuie : d'abord un résumé des données de linguistique discursive qui ont guidé la conception de D-STAG, ensuite une présentation de TAG et STAG. Elle se termine par une description du module intermédiaire permettant de passer de l'analyse phrastique à l'analyse discursive. La section 3 constitue le cœur de l'article, elle décrit la partie discursive de D-STAG. La section 4 compare D-STAG et D-LTAG. Enfin, nous concluons sur les perspectives d'implémentation de D-STAG pour le français.

2. Prérequis linguistiques, théoriques et formels

2.1. *Données de linguistique discursive*

Une relation de discours est souvent lexicalisée par un « connecteur de discours ». L'ensemble des connecteurs de discours comprend les conjonctions de subordination et de coordination (*parce que*, *ou*), et les adverbiaux de discours (*ensuite*, *par conséquent*). Un connecteur peut être ambigu, c'est-à-dire lexicaliser plusieurs relations de discours. Ainsi l'adverbial *ensuite* lexicalise la relation $Narration$ dans un récit narratif (« *Fred est allé au supermarché. Ensuite, il est allé au cinéma.* ») et la relation $Continuation$ dans une énumération (« *Le premier chapitre de la thèse expose la problématique. Ensuite, le second chapitre présente un état de l'art.* »). Une relation de discours n'est pas toujours lexicalisée par un connecteur. Par exemple, la relation $Explication$ dans le discours sans connecteur « *Fred est tombé. Marc lui a fait un croche-pied.* » doit être inférée sur la base de connaissances (extra)-linguistiques. Pour de tels cas, nous posons l'existence d'un connecteur adverbial vide noté ϵ, suivant en cela la position de (Harris, 1986). Par conséquent, nous posons que le discours « *Fred est tombé. Marc lui a fait un croche-pied.* » est de forme « P_1. ϵ P_2. », et, par abus de langage, nous disons que le connecteur vide « lexicalise » $Explication$. Avec le recours au connecteur vide, une relation de discours peut être considérée comme un prédicat sémantique avec deux arguments – qui sont les représentations sémantiques discursives de deux segments de discours (contigus) – qui est lexicalisée par un connecteur de discours avec deux arguments – qui sont les représentations syntaxiques discursifs de ces deux mêmes segments de discours. C'est sur ce principe que repose la partie discursive de D-STAG (section 3).

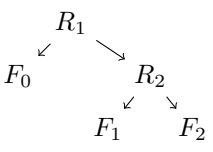
SDRT pose qu'il existe deux types de relations de discours, les relations cordonnantes ($Narration$, $Continuation$) et subordonnantes ($Explication$, $Commentaire$). RST – *Rhetorical Structure Theory* (Mann et Thompson, 1988), l'autre théorie dominante pour le discours – pose qu'il existe deux types d'arguments pour les relations de discours, les nuclei et les satellites. Ces distinctions sont équivalentes en considérant qu'une relation coordonnante (multinucléaire en RST) relie deux nuclei tandis qu'une relation subordonnante (nucleus-satellite en RST) relie un nucleus et un satellite. Un nucleus est un élément essentiel du discours tandis qu'un satellite est de moindre importance. Grâce à cette distinction, SDRT peut faire appel à la « contrainte de la frontière droite » (désormais RFC pour *Right Frontier Constraint*)

lors de la procédure incrémentale de construction des structures de discours. Cette contrainte stipule que le premier argument d'une relation coordonnante n'est pas ouvert pour l'attachement d'une information nouvelle. La RFC simplifie grandement la construction des structures de discours en limitant le nombre de points d'attachement possibles d'une nouvelle information. En D-STAG, nous avons évidemment profité de cet apport de la SDRT et la section 3.1.5 présentera l'implémentation de la RFC.

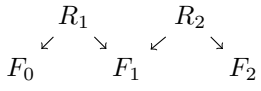
La question se pose de savoir à quoi correspondent les structures discursives représentées en graphes de dépendances (dans lesquels un prédicat domine ses arguments). L'idée est largement répandue que les graphes de dépendances représentant les structures discursives sont arborescents : c'est le principe de base de la RST (Mann et Thompson, 1988 ; Marcu, 2000), théorie sur laquelle se sont appuyés de nombreux systèmes d'analyse ou de génération de textes depuis une vingtaine d'années. Ce principe a aussi guidé la conception de D-LTAG qui ne peut calculer que des structures arborescentes. Pourtant, cette structure arborescente est plus un mythe qu'une réalité, comme montré dans (Wolf et Gibson, 2006) et dans une série de travaux antérieurs (Danlos, 2004a ; Danlos, 2004b ; Danlos, 2006). Les structures de discours en SDRT ne sont pas représentées comme des graphes de dépendances, cependant dans nos travaux antérieurs et dans cet article, nous convertissons les structures de discours construites en SDRT en graphes de dépendances. Ces graphes de dépendances sont des DAG – *Directed Acyclic Graphs* – non forcément arborescents. Ces DAG respectent néanmoins des contraintes fortes qui éliminent bon nombre de DAG ne correspondant à aucune structure de discours. Nous allons brièvement montrer ce point sur les discours de forme C_0 *parce que* C_1. Adv_2 C_2 dans lesquels *parce que* lexicalise $Explication$, le symbole C_i représente la *i*ème clause (la forme logique de C_i est notée F_i), et Adv_2 représente un connecteur adverbial. Ces discours reçoivent quatre types d'interprétation – mais pas plus que quatre – qui sont illustrés dans les exemples de (1).

(1) a. Fred est de mauvaise humeur parce qu'il a perdu ses clefs. De plus, il a raté son permis de conduire.
b. Fred est de mauvaise humeur parce qu'il a mal dormi. Il a fait des cauchemars.
c. Fred est allé au supermarché parce que son frigo était vide. Ensuite, il est allé au cinéma.
d. Fred est de mauvaise humeur parce que sa femme est absente pour une semaine. Ceci prouve qu'il l'aime vraiment beaucoup.

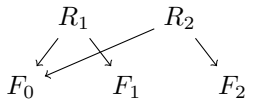
En (1a), $Adv_2 = de\ plus$ lexicalise la relation $Continuation$. Le segment de discours C_1 Adv_2 C_2 forme un constituant complexe dont la forme logique, i.e. $Continuation(F_1, F_2)$, est le second argument de $Explication$. La structure de discours est donc $Explication(F_0, Continuation(F_1, F_2))$, qui correspond à un DAG de dépendances arborescent, voir ci-contre avec $R_1 = Explication$ et $R_2 = Continuation$. Dans cet exemple, le second argument discursif de la conjonction *parce que* dépasse une frontière de phrase.



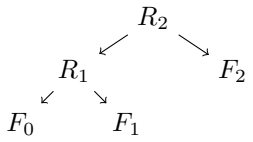
En (1b), $Adv_2 = \epsilon$ lexicalise la relation $Explication$. La structure de discours est $Explication(F_0, F_1) \wedge Explication(F_1, F_2)$, qui correspond à un DAG de dépendances non arborescent, voir ci-contre avec $R_2 = Explication$.



En (1c), $Adv_2 = ensuite$ lexicalise la relation $Narration$. La structure de discours est $Explication(F_0, F_1) \wedge Narration(F_0, F_2)$, qui correspond à un DAG de dépendances non arborescent, voir ci-contre avec $R_2 = Narration$.



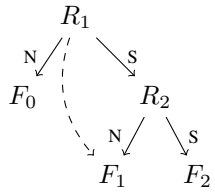
En (1d), $Adv_2 = \epsilon$ lexicalise la relation $Commentaire$. Le segment de discours C_0 *parce que* C_1 forme un constituant complexe dont la forme logique est le premier argument de $Commentaire$. La structure de discours est donc $Commentaire(Explication(F_0, F_1), F_2)$, qui correspond à un DAG de dépendances arborescent, voir ci-contre avec $R_2 = Commentaire$.



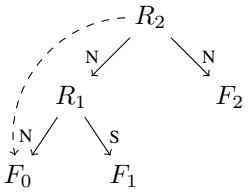
Signalons qu'à cause du mythe sur l'arborescence des structures discursives, l'existence de ces quatre types d'interprétation a été largement passée sous silence[1]. Plus précisément, les structures non arborescentes où un élément a deux parents sont représentées en RST sous forme d'arbres, ce qui est rendu possible en interprétant ces arbres avec le « principe de nucléarité » (Marcu, 2000). Ainsi les graphes de dépendances pour (1b) et (1c) construits en RST sont les arbres de la figure 1 où les arcs sont décorés par les symboles N pour nucleus et S pour satellite – rappelons que la relation $Explication$ est subordonnante ou nucleus-satellite, $Narration$ est coordonnante ou multinucléaire. Ces arbres doivent être interprétés avec le principe de nucléarité qui dit que si un nœud étiqueté par une relation de discours R_i a comme fils un nœud étiqueté par une relation nucleus-satellite R_j (d'où part donc un arc décoré par N et un arc décoré par S), alors R_i a pour argument le nucleus de R_j (et non R_j comme l'indiquerait l'interprétation standard des arbres). L'interprétation donnée par le principe de nucléarité est schématisée par une flèche en pointillés dans la figure 1.

Passons à (1a) et (1d) dont les structures RST avec leur interprétation se trouvent à la figure 2. Le principe de nucléarité n'entre pas en jeu pour (1a) car $R_2 = Continuation$ est coordonnante ou multi-nucléaire : cet arbre est interprété de façon standard, ce qui débouche sur la bonne interprétation, à savoir $Explication(F_0, Continuation(F_1, F_2))$. En revanche, le principe de nucléarité entre en jeu pour (1d), ce qui débouche sur l'interprétation $Explication(F_0, F_1) \wedge Commentaire(F_0, F_2)$ qui est fausse. Pour déboucher sur l'interprétation correcte, à

1. Toutefois, récemment dans (Lee *et al.*, 2008), il en est fait état à partir d'observations sur le *Penn Discourse Tree Bank* (PDTB), qui est un corpus anglais annoté manuellement pour les relations de discours et leurs arguments (PDTB Group, 2008). Nous reviendrons sur ce point dans la section 4.

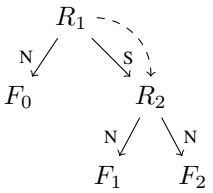


(1b) avec $R_1 = R_2 = Explication$

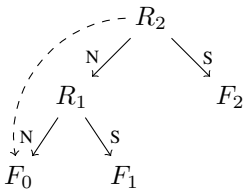


(1c) avec $R_1 = Explication$ et $R_2 = Narration$

Figure 1. *Structures de discours en* RST *pour (1b) et (1c) avec leur interprétation*



(1a) avec $R_1 = Explication$ et $R_2 = Continuation$



(1d) avec $R_1 = Explication$ et $R_2 = Commentaire$

Figure 2. *Structures de discours en* RST *pour (1a) et (1d) avec leur interprétation*

savoir $Commentaire(Explication(F_0, F_1), F_2)$, il faut interpréter cet arbre de façon standard.

En résumé, le principe de nucléarité permet de représenter sous forme d'arbres des interprétations où un élément d'information figure dans deux relations de discours, mais il bloque une interprétation pourtant valide qui demande d'interpréter les arbres de façon standard. Par conséquent, RST ne peut pas rendre compte du fait que les discours de forme C_0 *parce que* C_1. Adv_2 C_2 reçoivent quatre types d'interprétation. Pour prendre en compte cette réalité, il faut admettre le principe suivant : les graphes de dépendances pour les structures de discours sont des DAG non forcément arborescents, qui doivent être interprétés de façon standard[2].

Ce principe a guidé la conception de D-STAG. Plus précisément, à partir de ces données empiriques (et d'autres), nous avons établi les contraintes ci-dessous qui régissent

2. Le fait que RST impose de distinguer structures de discours et interprétation de ces structures prête souvent à confusion, même chez les auteurs utilisant ou s'inspirant de RST (nous reviendrons sur ce point à la section 4). En SDRT ou dans cet article, il ne peut pas y avoir de telle confusion dans la mesure où les structures de discours s'interprètent directement (de façon standard).

les arguments d'un connecteur/relation de discours[3], en utilisant la terminologie suivante. La clause où apparaît un connecteur est appelé sa « clause hôte ». Un connecteur adverbial apparaît en tête de sa clause hôte ou à l'intérieur de son noyau verbal. Une conjonction de subordination apparaît toujours en tête de sa clause hôte qui est appelée « subordonnée adverbiale ». Au niveau phrastique, une subordonnée adverbiale modifie une « principale ». Elle est située sur sa droite, sur sa gauche ou avant son noyau verbal. Quand elle est située sur sa droite, la conjonction de subordination est dite, par abus de langage, « postposée », sinon elle est dite « préposée ». Un connecteur/relation de discours a deux arguments qui sont les représentations syntaxiques/sémantiques de deux segments de discours, appelés le « segment hôte » et le « segment convié ». Ces segments sont régis par les contraintes suivantes.

Contrainte 1 *Le segment hôte d'un connecteur est identique ou commence à sa clause hôte (avec un éventuel dépassement de frontière de phrase).*

Contrainte 2 *Le segment convié d'un adverbial est n'importe où sur la gauche de son segment hôte (avec en général un dépassement de frontière de phrase)*[4].

Contrainte 3 *Le segment convié d'une conjonction postposée est sur la gauche de son segment hôte sans dépasser un connecteur adverbial (et donc en général sans dépassement d'une frontière de phrase)*[5].

Contrainte 4 *Le segment convié d'une conjonction préposée est identique ou commence à la principale (avec un éventuel dépassement de frontière de phrase).*

2.2. *Introduction à TAG et aux TAG synchrones*

Les parties entre guillemets de cette section sont traduites de (Nesson et Shieber, 2006). Les figures 3, 5 et 6 sont des adaptations pour le français de figures présentées dans cet article. L'exemple (très simple) que nous utilisons dans cette section est :

(2) Jean apparemment aime Marie

3. Ces contraintes ne concernent pas les conjonctions de coordination, qui demandent un traitement spécial, entre autres pour les coordinations corrélatives (section 3.4).

4. Cependant, le segment convié doit repecter la contrainte de la frontière droite. Ainsi, l'interprétation $R_1(F_0, F_1) \wedge R_2(F_0, F_2)$ n'est pas disponible lorsque R_1 est coordonnante, puisque le premier argument de R_2 ne peut pas être F_0 car C_0 n'est pas ouvert pour l'attachement d'une information nouvelle.

5. Dans une phrase de forme $C_0\ Conj_1\ C_1$, le segment convié de la conjonction de subordination $Conj_1$ ne peut être que C_0. Mais dans une phrase de forme $C_0\ Conj_1\ C_1\ Conj_2\ C_2$, le segment convié de $Conj_2$ peut être C_1, C_0 ou $C_0\ Conj_1\ C_1$ (Danlos, 2004b) .

2.2.1. *Introduction à TAG*

« Une grammaire d'arbres adjoints (TAG) consiste en un ensemble de structures d'arbres élémentaires et de deux opérations, la substitution et l'adjonction, utilisées pour combiner ces structures. Les arbres élémentaires peuvent être d'une profondeur arbitraire. Chaque nœud interne est étiqueté par un symbole non terminal. Les nœuds sur la frontière peuvent être étiquetés soit par des symboles terminaux soit par des symboles non terminaux avec un des signes $\downarrow$ ou $*$. L'emploi du signe $\downarrow$ sur un nœud frontière indique un *nœud à substitution*. L'opération de *substitution* s'applique quand un arbre élémentaire dont la racine est étiquetée par le symbole non terminal A est substitué à un nœud non terminal étiqueté A. Les arbres auxiliaires sont des arbres élémentaires dans lesquels la racine et un nœud frontière appelé *nœud pied* et marqué par le signe * sont étiquetés par un même symbole non terminal. L'opération d'*adjonction* consiste à insérer un arbre auxiliaire de racine et de nœud pied étiquetés A dans un arbre élémentaire, à un nœud aussi étiqueté A. La figure 3 montre des illustrations des opérations de substitution et d'adjonction concernant des arbres élémentaires simples. »

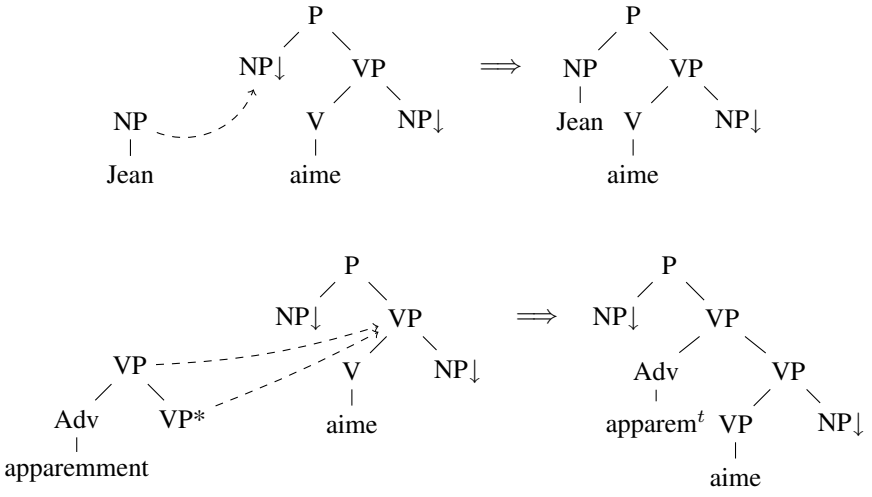


Figure 3. *Illustrations des opérations de substitution et d'adjonction en* TAG

À chaque nœud d'un arbre élémentaire peut être associée une structure de traits bipartite, divisée en une partie « amont» (en anglais « *top* ») et une partie « aval » (en anglais « *bottom* »). Lorsque deux arbres élémentaires sont combinés, deux types de règles sont définies pour l'unification des traits, l'une pour une combinaison d'arbres par adjonction, l'autre pour la substitution. À la fin d'une dérivation, les parties *top* et *bottom* doivent s'unifier à chaque nœud.

La sortie de l'analyse d'une phrase produit deux résultats : un « arbre dérivé » et un « arbre de dérivation ». L'arbre dérivé correspond à l'analyse syntagmatique. L'arbre de dérivation trace l'historique de la combinaison des arbres élémentaires. La

figure 4 présente l'arbre de dérivation pour la phrase (2) . L'opération de substitution est représentée par un arc en trait pointillé, celle d'adjonction par un arc en trait plein. Les arcs sont décorés par les adresses des nœuds où ont lieu les opérations de substitution ou d'adjonction, en suivant la convention de Gorn. Par exemple, la figure 4 nous dit que l'arbre auxiliaire modifieur dont le nom est βapparemment s'est adjoint au nœud dont l'adresse est 2 dans l'arbre initial nommé αaime. Le préfixe α est utilisé pour les noms d'arbres initiaux, le préfixe β pour les noms d'arbres auxiliaires.

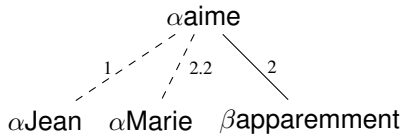


Figure 4. *Arbre de dérivation pour la phrase (2)*

2.2.2. *Introduction à* STAG

« Les TAG synchrones (STAG) prolongent les TAG en considérant les structures élémentaires comme des paires d'arbres TAG avec des liens entre certains nœuds de ces arbres. Une STAG est un ensemble de triplets $\langle t_L, t_R, \frown \rangle$ où t_L et t_R sont des arbres élémentaires TAG et $\frown$ est une relation de liage entre certains nœuds de t_L et certains nœuds de t_R (Shieber, 1994 ; Shieber et Schabes, 1990). La dérivation se déroule comme en TAG excepté que toutes les opérations doivent être appariées. En d'autres termes, un arbre ne peut être substitué ou adjoint à un nœud que si l'arbre apparié est simultanément substitué ou adjoint au nœud lié. Nous notons les liens en utilisant des indices dans des cercles (par exemple ①) qui viennent décorer les nœuds liés. »

STAG a été utilisé avec succès dans une interface syntaxe-sémantique pour l'anglais qui peut traiter de phrases soulevant des problèmes délicats de portée (Nesson et Shieber, 2006). Cette interface est illustrée à la figure 5 par les paires d'arbres élémentaires pour l'analyse de la phrase (2).

Dans l'arbre de dérivation, « les arcs sont décorés avec des liens qui spécifient un numéro de liage dans la paire d'arbres élémentaires. Ces liens donnent l'adresse des opérations dans l'arbre syntaxique et dans l'arbre sémantique. Ces opérations doivent opérer sur des nœuds liés dans la paire d'arbres élémentaires concernée. » La figure 6 donne l'unique arbre de dérivation pour les analyses syntaxique et sémantique de (2). L'arbre dérivé sémantique débouche sur la forme logique (simplifiée) $apparemment(aime(jean, marie))$.

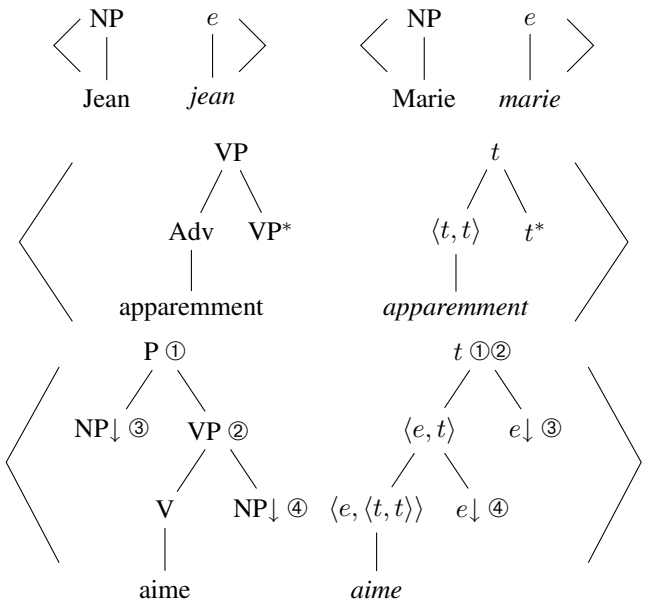


Figure 5. *Paire d'arbres en* STAG *syntaxe–sémantique pour analyser (2)*

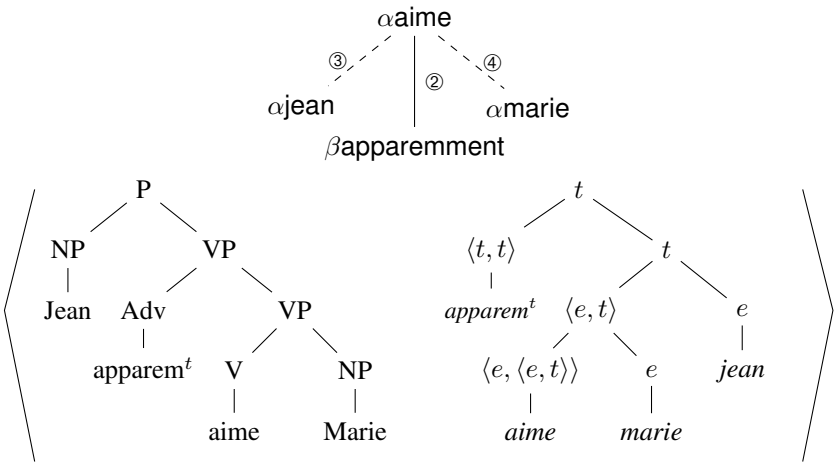


Figure 6. *Arbre de dérivation (unique) pour les arbres dérivés syntaxique et sémantique de (2)*

2.3. *Interface phrase-discours*

Expliquons d'abord pourquoi cette interface est nécessaire. L'idée de D-STAG est de prolonger un analyseur phrastique au niveau discursif *sans rien changer à l'analyseur phrastique*. Mais on ne peut pas passer directement de la phrase au discours car les arguments d'un connecteur au niveau discursif présentent de fortes disparités avec ses arguments au niveau phrastique. D'abord, au niveau discursif un connecteur adverbial a obligatoirement deux arguments, tandis qu'au niveau phrastique, il n'a, comme tout autre adverbial, qu'un seul argument (de catégorie P s'il est placé en tête de sa clause hôte ou de catégorie V s'il se situe à l'intérieur du noyau verbal de sa clause hôte). Ensuite, au niveau discursif un argument d'une conjonction de subordination peut dépasser une frontière de phrase (voir (1a) pour une conjonction postposée et (8) ci-dessous pour une conjonction préposée) alors qu'il ne saurait en être question au niveau phrastique.

En conclusion, il est nécessaire de passer par une interface phrase-discours qui donne aux frontières de phrases le simple rôle d'un signe de ponctuation et qui permet de recalculer les (deux) arguments d'un connecteur. Une telle interface est aussi utilisée en D-LTAG et nous nous en sommes inspirée. À partir de l'analyse syntaxique phrastique, cette interface produit de façon déterministe une « forme normalisée de discours », abrégée en FND, qui est une suite de « mots de discours » où chaque mot de discours est, par exemple, un connecteur, un identifiant C_i pour une clause (sans connecteur) ou un signe de ponctuation. Les analyses syntaxiques et sémantiques des clauses sont alors celles obtenues par l'analyseur phrastique en supprimant les connecteurs. Dans une FND, un connecteur adverbial est toujours placé en tête de sa clause hôte, avec éventuellement un trait, noté vp, gardant la trace de sa position s'il modifiait un nœud V du noyau verbal de sa phrase hôte – ce trait est nécessaire car le rôle discursif d'un connecteur peut dépendre de sa position (Bras, 2008). Dans le même ordre d'idées, une subordonnée adverbiale qui apparaît entre le sujet et le noyau verbal de sa clause hôte est placée en tête de sa phrase hôte avec un trait gardant la trace de sa position médiane. Enfin, si une phrase sous forme normalisée (sauf la première phrase du discours) ne commence pas par un connecteur adverbial, le connecteur adverbial vide ϵ est introduit. À titre d'illustration, pour le discours (3), la FND est : $C_0.\ Ensuite^{vp}\ C_1\ parce\ que\ C_2.\ \epsilon\ comme\ C_3,\ C_4.$

(3) Fred est allé au cinéma. Il a ensuite dévoré un steak parce qu'il avait faim. Comme il avait travaillé comme un dingue, il n'avait rien mangé depuis hier.

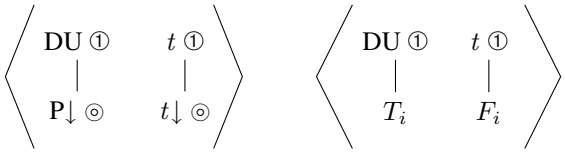
Une FND suit une grammaire régulière décrivant la séquence de ses éléments. Dans la section 3.1, consacrée aux connecteurs adverbiaux et aux conjonctions de subordination postposées, les FND suivent l'expression régulière suivante (en faisant abstraction des traits de position) : $C\ (Punct\ Conn\ C)^*$, où la séquence $Punct\ Conn$ est soit $.\ Adv$ soit $(,)\ Conj$ où la virgule est facultative. Une FND qui suit cette expression régulière est notée (en omettant les signes de ponctuation) $C_0\ Conn_1\ C_1 \ldots\ Conn_n\ C_n$, avec $Conn_i = Adv_i$ ou $Conj_i$. Une FND comportant au moins une conjonction pré-

posée comporte un élément C précédé de l'expression $Conj\ C((,)\ Conj\ C)^*$. Les connecteurs Adv_i et $Conj_i$ peuvent être facultativement suivis ou précédés d'un modifieur (section 3.3). Les conjonctions de coordination sont étudiées à la section 3.4. Le cas des « connecteurs multiples » où deux connecteurs se partagent la même phrase hôte (par exemple, dans une FND de forme $C_0\ Conn_1\ C_1\ Conj_2\ Adv_3\ C_2$ où C_2 est la clause hôte de $Conj_2$ et Adv_3) est examiné dans la section 4.

Cette grammaire régulière ne prend pas en compte le fait qu'une clause peut comporter relatives, complétives ou incises. Or ces sous-clauses jouent souvent un rôle discursif important. Ainsi la relation $Attribution$, qui relie une incise comme *,a annoncé l'AFP,* et le contenu de cette annonce, doit être détectée pour de nombreuses applications du TAL (Prasad *et al.*, 2006). Néanmoins, nous n'avons pas encore intégré la relation $Attribution$ dans notre grammaire discursive, et plus généralement, nous considérons les clauses comme des unités qui ne sont jamais décomposées en sous-clauses. Dans les perspectives de recherches futures, nous projetons de compléter la grammaire régulière des FND et d'étendre la partie discursive de D-STAG en conséquence.

3. Partie discursive de D-STAG

Pour une clause C_i (sans connecteur), rappelons que l'interface phrase-discours fournit son arbre syntaxique de racine P noté T_i, son arbre sémantique de racine t noté F_i et son arbre de dérivation noté η_i. Pour immerger les analyses phrastiques dans les analyses discursives, nous utilisons la paire α*P-to-D* donnée à la figure 7-a, où le symbole DU représente la catégorie « unité de discours » (*Discours Unit*). Dans la suite de cet article, nous notons τ_i l'arbre de dérivation α*P-to-D* dans lequel η_i est substitué au lien ◎ ; τ_i correspond à la paire donnée à la figure 7-b. Enfin, nous utilisons la convention suivante : les arbres de notre grammaire comportant au plus un nœud à substitution, celui-ci, quand il existe, porte systématiquement le lien ◎. Cette convention nous permet d'éviter de préciser systématiquement le lien auquel une opération de substitution a lieu.



(a) α*P-to-D* (b) τ_i

Figure 7. *Paires d'arbres αP-to-D et τ_i*

Lorsqu'un connecteur donné $Conn_i$ n'exprime qu'une seule relation de discours R_i, le principe de base de la grammaire STAG discursive consiste à élaborer une paire d'arbres, nommée $Conn_i \div R_i$, dont l'arbre syntaxique est ancré par $Conn_i$ et dont l'arbre sémantique est ancré par un lambda-terme associé à R_i. Par abus de langage,

nous parlons d'arbre sémantique ancré par R_i. Lorsqu'un connecteur est ambigu, c'est-à-dire qu'il permet d'exprimer plusieurs relations de discours, sont créées autant de paires d'arbres qu'il y a de relations de discours exprimées par le connecteur (dans chacune de ces paires, l'arbre syntaxique est ancré par le connecteur ambigu). La gestion de l'ambiguïté est abordée à la section 5.

Nous commençons par présenter la grammaire STAG pour les adverbiaux et les conjonctions postposées, connecteurs qui ont un comportement discursif similaire (section 3.1), puis nous passerons aux conjonctions préposées (section 3.2). Ensuite, nous examinerons les modifieurs de connecteurs (section 3.3) et les conjonctions de coordination (section 3.4).

3.1. *Connecteurs adverbiaux et conjonction postposées*

3.1.1. *Arbres syntaxiques*

Les arbres syntaxiques ancrés par un connecteur adverbial et une conjonction postposée sont donnés à la figure 8, où un connecteur est de catégorie DC pour *Discours Connective*. Ils ne diffèrent, en faisant abstraction des traits qui seront expliqués ultérieurement, que par les coancres lexicales qui sont des signes de ponctuation de catégorie $Punct$, le sous-arbre de racine $Punct$ étant facultatif pour une conjonction préposée.

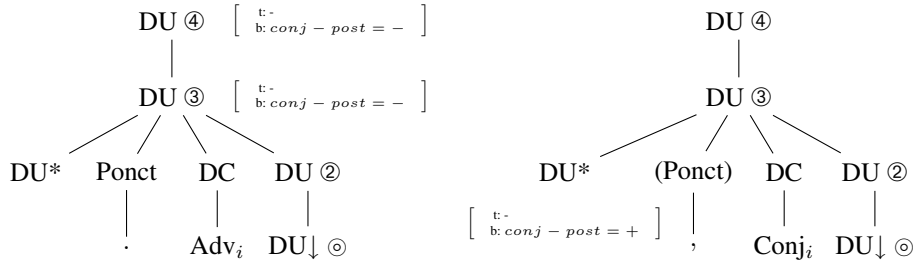


Figure 8. *Arbres syntaxiques pour les adverbiaux et les conjonctions postposées*

Ces arbres suivent les principes suivants : ce sont des arbres **auxiliaires** à deux arguments représentés par un un nœud à substitution DU↓ et nœud pied DU*. Le nœud à substitution DU↓, qui porte le lien ◎, correspond à l'argument hôte du connecteur. Il sert à substituer l'arbre comportant un nœud DU qui domine l'analyse syntaxique de la clause hôte du connecteur, cet arbre ayant éventuellement reçu une adjonction à sa racine (si c'est le cas, le segment hôte du connecteur commence – sans être identique – à sa clause hôte, voir la contrainte 1 posée à la section 2.1). Le nœud pied DU* correspond à l'argument convié du connecteur. Il est situé sur sa gauche, ce qui respecte les contraintes 2 et 3 posées à la section 2.1. Le fait que le segment convié

d'une conjonction postposée ne peut pas dépasser un connecteur adverbial et donc une frontière de phrase (contrairement à celui d'un adverbial) est pris en compte par les traits $[conj - post = \pm]$ expliqués en 3.1.3. L'implémentation de la RFC est décrite en 3.1.5.

Nous postulons une procédure incrémentale d'analyse où la séquence des éléments d'une FND de forme $C_0\ Conn_1\ C_1 \ldots Conn_n\ C_n$ (en omettant les signes de ponctuation) est traitée de gauche à droite. Après avoir analysé $C_0\ Conn_1\ C_1 \ldots Conn_{n-1}\ C_{n-1}$, l'attachement du nouveau connecteur $Conn_n$ se fait par adjonction de l'arbre ancré par $Conn_n$ sur un nœud DU de **la frontière droite** de l'arbre syntaxique représentant $C_0 \ldots C_{n-1}$ (pour respecter l'ordre linéaire de la FND). L'attachement de la nouvelle clause C_n s'effectue par substitution de l'arbre syntaxique de la paire τ_n au nœud à substitution DU$\downarrow$ de l'arbre ancré par $Conn_n$. Soulignons que cette procédure ne tient pas compte du découpage du texte en phrases. L'unité d'analyse discursive n'est pas la phrase mais une paire $Conn_n\ C_n$.

Les arbres ancrés par un adverbial ou une conjonction postposée comportent trois nœuds étiquetés DU, avec le lien ②, ③ ou ④, sur leur frontière droite. Ces nœuds portent des liens différents qui vont permettre de varier l'interprétation sémantique obtenue, comme montré ci-dessous. Nous avons créé trois nœuds DU avec des liens différents, et non un seul nœud DU portant trois liens différents, pour pouvoir faire plusieurs adjonctions à différents nœuds du même arbre, par exemple une adjonction à DU③ pour attacher $Conn_n$ et une adjonction à DU④ pour attacher $Conn_{n+1}$. Soulignons que si l'on a fait une adjonction à DU ③ pour attacher $Conn_n$, DU② n'est plus sur la frontière droite de l'arbre syntaxique. On ne peut donc plus faire d'adjonction à DU② pour attacher $Conn_{n+1}$. Cette contrainte sera généralisée en 3.1.4 (voir la contrainte 5).

3.1.2. *Arbres sémantiques*

À première vue, on pourrait penser qu'une relation de discours R_i est associée au foncteur $\mathcal{R}_i = \lambda xy.R_i(x, y)$ avec $x, y : t$, $R_i(x, y) : t$, et $\mathcal{R}_i : \langle t, \langle t, t\rangle\rangle$, $\mathcal{R}_i$ ancrant un arbre avec un nœud pied $t*$ et un nœud à substitution $t\downarrow$. Ceci est uniquement approprié pour analyser une simple FND à deux clauses, par exemple une FND de forme $C_0\ parce\ que\ C_1$ comme montré dans la figure 9 dans laquelle $\beta_1 = parce\ que_{post} \div Explication$.

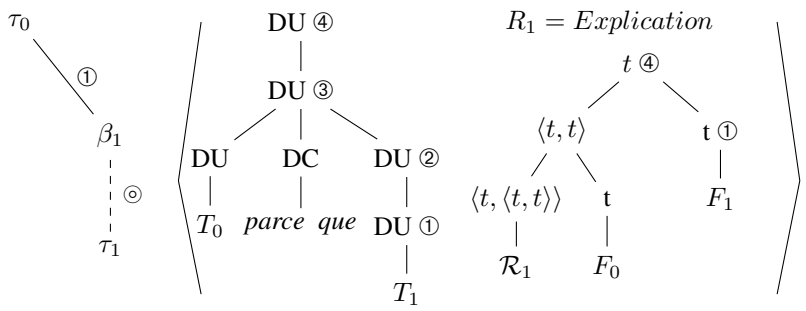


Figure 9. *Arbre de dérivation et paire d'arbres dérivés pour une* FND *de forme* C_0 *parce que* C_1 *(avec le foncteur* $\mathcal{R}_1$ *dans l'arbre sémantique)*

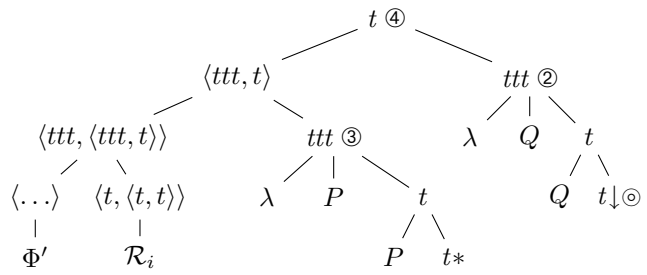
Toutefois, avec le simple foncteur $\mathcal{R}_i$, il est impossible d'obtenir pour les FND à trois clauses (de forme $C_0\ Conn_1\ C_1\ Conn_2\ C_2$) quatre types d'interprétation, dont deux avec une conjonction de formules, voir section 2.1. De ce fait, nous avons défini deux opérateurs à montée de type Φ' et Φ'' : ils prennent $\mathcal{R}_i$ comme argument et retournent deux nouveaux foncteurs $\mathcal{R}'_i$ et $\mathcal{R}''_i$ associés à la relation de discours R_i. Ci-dessous leur définition.

Définition 1 $\Phi' = \lambda\mathcal{R}_i XY.X(\lambda x.Y(\lambda y.\mathcal{R}_i(x,y)))$
$\Phi'(\mathcal{R}_i) = \mathcal{R}'_i = \lambda XY.X(\lambda x.Y(\lambda y.\mathcal{R}_i(x,y)))$
avec $X, Y : ttt = \langle\langle t,t\rangle, t\rangle$ et $x, y : t$

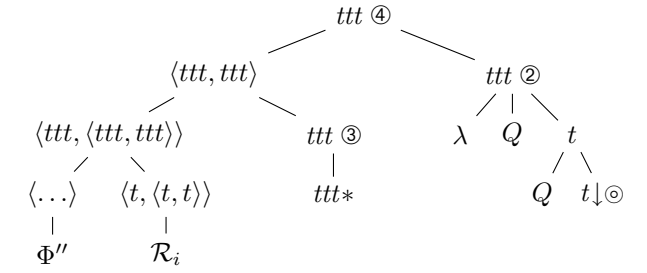
Φ' s'accompagne d'une montée de type. Il débouche sur le foncteur $\mathcal{R}'_i$ de type $\langle ttt, \langle ttt, t\rangle\rangle$ dans lequel ttt symbolise le type $\langle\langle t,t\rangle, t\rangle$. Il coancre l'arbre (A), donné dans la figure 10-a, dont le nœud pied est de type t. (A) est utilisé pour les adjonctions aux liens ① et ④. Si le premier argument de $\mathcal{R}'_i$ est $\lambda P.P(F_0)$ de type ttt, le second $\lambda Q.Q(F_1)$ de type ttt, alors le résultat est $R_i(F_0, F_1)$ de type t. Pour une FND à deux clauses, $\mathcal{R}'_i$ débouche donc sur le même résultat que $\mathcal{R}_i$. Cependant, la montée de type est nécessaire pour introduire les nœuds ttt② et ttt③ auxquels l'arbre (B) peut s'adjoindre.

Définition 2 $\Phi'' = \lambda\mathcal{R}_i XYP.X(\lambda x.Y(\lambda y.\mathcal{R}_i(x,y) \wedge P(x)))$
$\Phi''(\mathcal{R}_i) = \mathcal{R}''_i = \lambda XYP.X(\lambda x.Y(\lambda y.\mathcal{R}_i(x,y) \wedge P(x)))$
avec $X, Y : ttt = \langle\langle t,t\rangle, t\rangle,\ P : \langle t,t\rangle$ et $x, y : t$

Φ'' introduit une conjonction de termes. Il débouche sur le foncteur $\mathcal{R}''_i$ de type $\langle ttt, \langle ttt, ttt\rangle\rangle$. Il coancre l'arbre (B), donné dans la figure 10-b, dont le nœud pied est de type ttt. (B) est utilisé pour les adjonctions aux liens ② et ③. Si le premier argument de $\mathcal{R}''_i$ est $\lambda P.P(F_0)$, le second $\lambda Q.Q(F_1)$, alors le résultat est $\lambda P.(R_i(F_0, F_1) \wedge P(F1))$ de type ttt.



(a) Arbre (A)



(b) Arbre (B)

Figure 10. *Arbres sémantiques (A) et (B) ancrés par* $\mathcal{R}'_i = \phi'(\mathcal{R}_i)$ *et* $\mathcal{R}''_i = \phi''(\mathcal{R}_i)$

3.1.3. *Analyse des* FND *à trois clauses*

Pour les FND à trois clauses, on doit pouvoir calculer quatre types d'interprétation, qui ont été illustrés dans les exemples (1) de forme C_0 *parce que* C_1. Adv_2 C_2 présentés à la section2.1. Nous allons expliquer l'analyse de ces exemples. Nous notons β_1 la paire d'arbres $parce\ que_{post} \div Explication$ et β_2 la paire $Adv_2 \div R_2$. Après avoir analysé C_0 *parce que* C_1, l'arbre syntaxique est celui montré à la figure 9. La frontière droite de cet arbre contient quatre nœuds DU qui portent le lien ① venant de l'arbre syntaxique de τ_1 ou le lien ②, ③ ou ④ venant de l'arbre syntaxique ancré par *parce que*. Les analyses des quatre exemples de (1) s'obtiennent en adjoignant β_2 à un de ces liens.

Commençons par (1a), répété en (4), avec $\beta_2 = de\ plus \div Continuation$, pour lequel la structure de discours est $Explication(F_0, Continuation(F_1, F_2))$.

(4) Fred est de mauvaise humeur parce qu'il a perdu ses clefs. De plus, il a raté son permis de conduire.

Cet exemple s'analyse en adjoignant β_2 au lien ① de τ_1, Le nœud avec le lien ① dans l'arbre sémantique de τ_1 est de type t. Pour la satisfaction des types, on doit donc utiliser

pour R_2 l'arbre (A) ancré par $\mathcal{R}'_2$ dont le nœud pied est de type t. L'arbre dérivé sémantique pour (1a) = (4) est donné à la figure 11. Le sous-arbre de racine ttt à l'adresse de Gorn 2 débouche sur le lambda-terme $\lambda P.P(Continuation(F_1, F_2))$ avec $P : \langle t, t\rangle$. $Continuation(F_1, F_2)$ est donc le second argument de $R_1 = Explication$, qui a pour premier argument F_0, d'où la formule $Explication(F_0, Continuation(F_1, F_2))$.

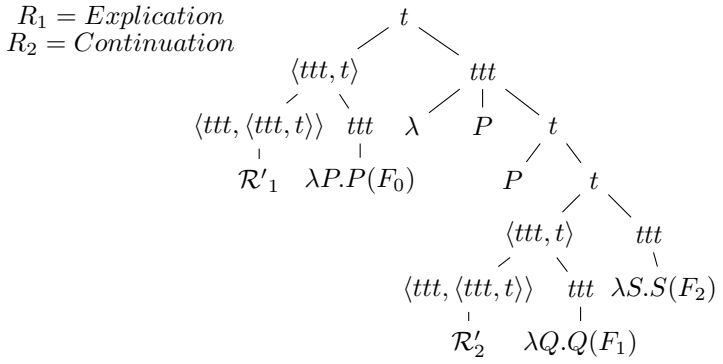


Figure 11. *Arbre sémantique dérivé pour (1a) = (4) avec l'interprétation* $R_1(F_0, R_2(F_1, F_2))$

Continuons par (1b), répété en (5), avec $\beta_2 = \epsilon \div Explication$, pour lequel la structure de discours est $Explication(F_0, F_1) \wedge Explication(F_1, F_2)$ avec une conjonction de formules.

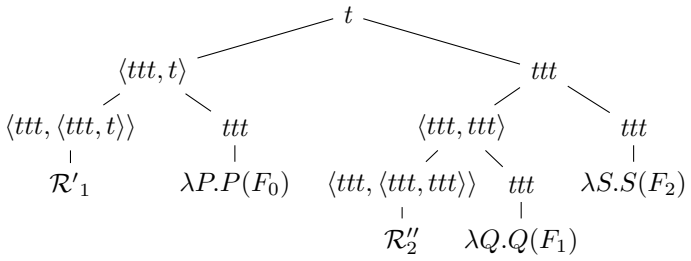
(5) Fred est de mauvaise humeur parce qu'il a mal dormi. Il a fait des cauchemars.

Cet exemple s'analyse en adjoignant β_2 au lien ② de β_1. Le nœud avec le lien ② dans l'arbre sémantique de β_1 est de type ttt. Pour la satisfaction des types, on doit donc utiliser pour R_2 l'arbre (B) ancré par $\mathcal{R}''_2$ dont le nœud pied est de type ttt. L'arbre dérivé sémantique pour (1b) = (5) est donné à la figure 12-a. Le sous-arbre de racine ttt à l'adresse de Gorn 2 débouche sur le lambda-terme $\lambda P.(Explication(F_1, F_2) \wedge P(F_1))$ avec $P : \langle t, t\rangle$. Comme seule F_1 est sous la portée de P, F_1 est le second argument de $R_1 = Explication$, qui a pour premier argument F_0, d'où la formule $Explication(F_0, F_1) \wedge Explication(F_1, F_2)$.

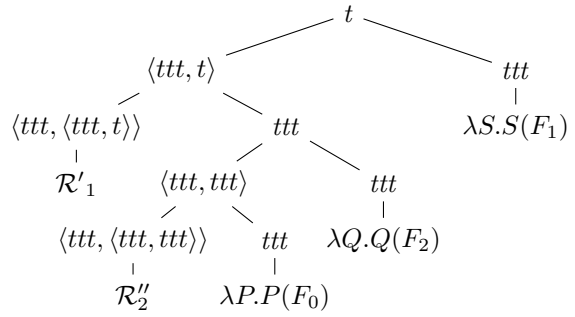
Passons à (1c), répété en (6), avec $\beta_2 = ensuite \div Narration$, pour lequel la structure de discours est $Explication(F_0, F_1) \wedge Narration(F_0, F_2)$ avec aussi une conjonction de formules.

(6) Fred est allé au supermarché parce que son frigo était vide. Ensuite, il est allé au cinéma.

Cet exemple s'analyse en adjoignant β_2 au lien ③ de β_1. Ce cas étant similaire au cas précédent, nous nous contentons de donner l'arbre dérivé sémantique à la figure 12-b.



(a) (1b) avec $R_1 = R_2 = Explication$



(b) (1c) avec $R_1 = Explication$ et $R_2 = Narration$

Figure 12. *Arbres dérivés sémantiques pour (1b) = (5) et (1c) = (6) avec les interprétations* $R_1(F_0, F_1) \wedge R_2(F_1, F_2)$ *et* $R_1(F_0, F_1) \wedge R_2(F_0, F_2)$

Terminons par (1d), répété en (7), avec $\beta_2 = \epsilon \div Commentaire$, pour lequel on veut calculer la formule sémantique $Commentaire(Explication(F_0, F_1), F_2))$.

(7) Fred est de mauvaise humeur parce que sa femme est absente pour une semaine.
ϵ Ceci prouve qu'il l'aime vraiment beaucoup.

Cet exemple s'analyse en adjoignant β_2 au lien ④ de β_1, lien porté par un nœud t dans l'arbre sémantique de β_1. Pour la satisfaction des types, on doit utiliser pour R_2 l'arbre (A) de nœud pied t^* ancré par $\mathcal{R}'_2$. L'arbre dérivé sémantique pour (1d) = (7) est donné à la figure 13. Le sous-arbre de racine *ttt* à l'adresse de Gorn 1.2 débouche sur le lambda-terme $\lambda P.P(Explication(F_0, F_1))$ avec $P : \langle t, t\rangle$. $Explication(F_0, F_1)$ est donc le premier argument de $R_2 = Commentaire$, qui a comme second argument F_2, d'où la formule $Commentaire(Explication(F_0, F_1), F_2)$.

En conclusion, les quatre types d'interprétation des FND de forme $C_0\ Conj_1\ C_1.\ Adv_2\ C_2$ s'obtiennent grâce aux quatre sites d'adjonction qui sont sur la frontière droite de l'arbre syntaxique de $C_0\ Conj_1\ C_1$ et grâce aux arbres sémantiques (A) et (B) respectivement de nœud pied t et ttt et ancrés par $\mathcal{R}'_i$ et $\mathcal{R}''_i$.

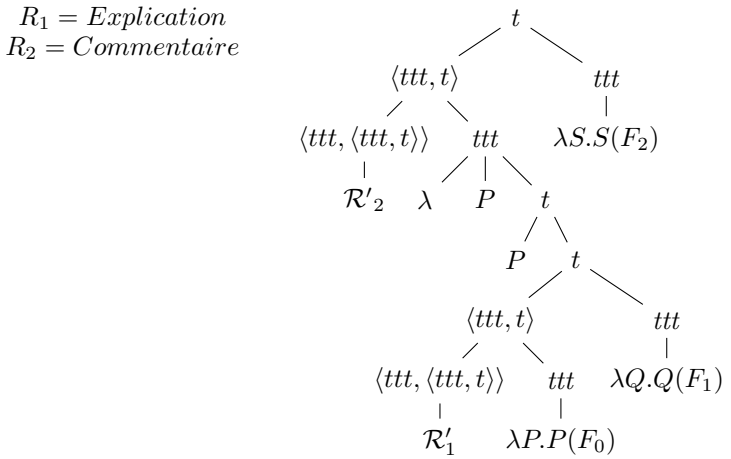


Figure 13. *Arbre dérivé sémantique pour (1d) = (7) avec l'interprétation* $R_2(R_1(F_0, F_1), F_2)$

Pour les FND à trois clauses de forme simplifiée $C_0\ Conn_1\ C_1\ Conn_2\ C_2$, nous venons d'examiner en détail les discours de forme $C_0\ Conj_1\ C_1.\ Adv_2\ C_2$ où $Conn_1$ est une conjonction postposée et $Conn_2$ un connecteur adverbial. Il reste trois cas :

– $Conn_1$ est une conjonction postposée et $Conn_2$ aussi ;

– $Conn_1$ est un adverbial et $Conn_2$ aussi ;

– $Conn_1$ est un adverbial et $Conn_2$ une conjonction postposée.

Les deux premiers cas ne posent aucun problème nouveau. Ils s'obtiennent par les mécanismes développés en détail pour $C_0\ Conj_1\ C_1.\ Adv_2\ C_2$. Le troisième cas, c'est-à-dire les FND de forme $C_0.\ Adv_1\ C_1\ Conj_2\ C_2$, soulève la question de l'implémentation de la contrainte 3 de la section 2.1 stipulant que le segment convié de la conjonction postposée $Conj_2$ ne peut pas dépasser un connecteur adverbial et donc une frontière de phrase (indiquons que cette contrainte n'est pas implémentée en SDRT). Cette contrainte est implémentée grâce aux traits $[conj - post = \pm]$ décorant certains nœuds des arbres syntaxiques ancrés par un adverbial ou une conjonction postposée, voir les arbres de la figure 8. Plus précisément :

– le nœud pied d'un arbre ancré par une conjonction postposée est décoré par le trait *bottom* $[conj - post = +]$ ce qui se glose par « cet arbre auxiliaire sert à adjoindre une conjonction postposée » ;

– les nœuds de catégorie DU portant les liens ③ et ④ dans un arbre ancré par un connecteur adverbial sont décorés par un trait *bottom* $[conj - post = -]$; cette décoration se glose par « ce nœud ne peut pas être le site d'une adjonction pour un arbre de conjonction postposée ».

Ces traits bloquent l'adjonction de $Conj_2 \div R_2$ aux liens ③ et ④ de $Adv_1 \div R_1$ grâce à l'échec d'unification $[conj-post = +] \cup [conj-post = -]$. Par conséquent, $Conj_2 \div R_2$ ne peut s'adjoindre qu'au lien ② de $Adv_1 \div R_1$ et au lien ① de τ_1, ce qui débouche respectivement sur les interprétations $R_1(F_0, F_1) \wedge R_2(F_1, F_2)$ et $R_1(F_0, R_2(F_1, F_2))$, où le premier argument de R_2 est F_1. Ces interprétations respectent la contrainte 3 : l'argument convié de R_2 est F_1, donc le segment convié de $Conj_2$ est C_1 sans dépasser Adv_1 (précédé d'une frontière de phrase).

3.1.4. *Analyse des* FND *à n clauses ($n > 3$) de forme $C_0\ Conn_1\ C_1 \ldots\ Conn_n\ C_n$*

Pour les FND à n clauses avec $n > 3$, aucun mécanisme nouveau n'est mis en jeu. Attacher $Conn_n$ puis C_n revient, dans l'arbre de dérivation représentant $C_0 \ldots C_{n-1}$, à adjoindre $Conn_n \div R_n$ – dans lequel on a substitué τ_n – au lien ① de τ_{n-1} ou au lien (i) avec $i \in \{2, 3, 4\}$ d'un nœud $\beta_k = Conn_k \div R_k$, le nœud portant le lien (i) dans l'arbre syntaxique de β_k devant être sur la frontière droite de l'arbre syntaxique de $C_0 \ldots C_{n-1}$ (pour respecter l'ordre linéaire de la FND). Comme il est fastidieux de calculer (la frontière droite de) l'arbre dérivé syntaxique, il est pratique de définir une notion de frontière droite sur l'arbre de dérivation. Un arbre de dérivation étant intrinsèquement non ordonné[6], on doit avoir recours à une convention graphique qui représente un arbre de dérivation comme ordonné. La convention suivante est suffisante : les nœuds τ_k projetés sur une droite sont ordonnés en suivant l'ordre linéaire des C_k dans la FND. Avec cette relation d'ordre notée $\prec$ sur les τ_k, les nœuds β_k de l'arbre de dérivation qui permettent d'attacher $Conn_n\ C_n$ sont ceux qui sont situés sur la frontière droite de l'arbre de dérivation en respectant la contrainte 5 régissant deux adjonctions aux liens (n) et (m) d'un même nœud[7] :

Contrainte 5 *Si β_j – dans lequel on a substitué τ_j – est adjoint au lien (n) d'un nœud β_i, alors on ne peut adjoindre β_k – dans lequel on a substitué τ_k – au lien (m) du même nœud β_i qu'en respectant la règle suivante : $\tau_j \prec \tau_k \Rightarrow n < m$ (avec n et m appartenant à $\{2, 3, 4\}$).*

Cette contrainte généralise celle que nous avions formulée dans la section 3.1.1 : si on a fait une adjonction au nœud DU③ d'un arbre syntaxique ancré par $Conn_i$, alors on peut faire une nouvelle adjonction dans cet arbre au nœud DU④ mais pas au nœud DU②.

3.1.5. *Implémentation de la RFC*

Rappelons que la RFC postulée en SDRT (section 2.1) repose sur une distinction entre deux types de relations de discours, les relations coordonnantes et les relations

6. Les adjonctions multiples à un même nœud devant se faire à des liens différents, elles peuvent être effectuées dans n'importe quel ordre tout en donnant le même résultat.

7. Cette contrainte est valide car nous avons pris le soin d'attribuer les liens ②, ③ et ④ de façon soignée.

subordonnantes. Elle stipule qu'il est interdit d'attacher une information nouvelle au premier argument d'une relation coordonnante. Par conséquent, l'interprétation $R_1(F_0, F_1) \wedge R_2(F_0, F_2)$ est interdite lorsque R_1 est coordonnante, par exemple.

L'implémentation de la RFC en D-STAG demande en premier lieu de pouvoir distinguer les arbres sémantiques ancrés par une relation coordonnante *versus* subordonnante. Ceci est obtenu en créant deux copies des arbres (A) et (B), copies qui se distinguent par un trait *top* $[coord = \pm]$ sur leur nœud pied. La RFC se traduit alors par le fait que toute adjonction au lien ③ de (A) ou (B) dont le nœud pied est décoré du trait $[coord = +]$ est interdite. En effet, c'est le lien ③ qui sert à obtenir des interprétations de type $R_1(F_0, F_1) \wedge R_2(F_0, F_2)$ qui doivent être bloquées lorsque R_1 est coordonnante.

En fait, la RFC induit d'autres contraintes que celle que nous venons de traiter et il existe d'autres contraintes sémantiques fondées sur la distinction entre relations coordonnantes *versus* subordonnantes, par exemple la contrainte de « *Continuous Discourse Pattern* » (Asher et Vieu, 2005). Nous n'avons pas la place de décrire ces contraintes, mais nous pouvons dire qu'elles sont facilement prises en compte en D-STAG grâce à des jeux de traits dans les arbres (A) et (B)[8].

Une remarque : comme les paires notées $Conn_i \div R_i$ associent un connecteur $Conn_i$ avec une relation de discours particulière R_i, il est possible de transférer l'information sur la nature coordonnante *versus* subordonnante de R_i vers le connecteur. Ceci demande de créer deux copies des schémas d'arbres syntaxiques ancrés par un connecteur, copies qui se distinguent par un trait top $[coord = \pm]$ sur leur nœud pied. Avec ces copies, tous les traits qui ont été ajoutés dans les deux copies des arbres sémantiques (A) et (B) pour prendre en compte la RFC ou d'autres contraintes sémantiques peuvent être transférés dans les arbres syntaxiques ancrés par les connecteurs. Ceci permet de prendre en compte les contraintes sémantiques au niveau syntaxique, ce qui peut s'avérer fort utile, voir la section 5.

3.2. *Conjonctions préposées*

L'arbre syntaxique ancré par une conjonction préposée est donné à la figure 14. Il permet de respecter la contrainte 4 établie à la section 2.1. Il se distingue des arbres ancrés par un connecteur adverbial ou une conjonction postposée, entre autres, par le fait que le nœud pied DU* est immédiatement dominé par un nœud DU portant le lien ⑤. Pour prendre en compte ce nouveau lien, on ajoute dans les deux copies de l'arbre sémantique (A) un nœud t⑤ dominant immédiatement le nœud pied $t*$. Nous allons examiner uniquement les adjonctions au lien ⑤ d'un arbre de conjonction préposée, les adjonctions aux liens ②, ③ et ④ se déroulant comme établi précédemment.

L'adjonction au lien ⑤ d'un arbre de conjonction préposée sert lorsque la conjonction introduit un « cadre de discours » (Charolles, 2005) comme illustré en (8) de

8. Pour la contrainte de *Continuous Discourse Pattern*, ces traits évitent le recours à un processus non monotone qui est un point de passage obligé en SDRT.

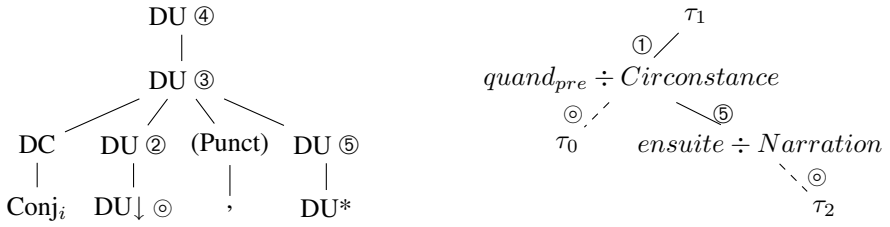


Figure 14. *Arbre ancré par une conjonction préposée et arbre de dérivation pour (8)*

forme $Quand\ C_0,\ C_1.\ Ensuite,\ C_2$. Dans ce discours, la conjonction *quand* a son segment convié qui passe une frontière de phrase. En effet, la structure de discours est $Circonstance(Narration(F_1, F_2), F_0)$[9], en posant que *quand* exprime $Circonstance$ et *ensuite* $Narration$.

(8) Quand il était à Paris, Fred est allé à la tour Eiffel. Ensuite, il a visité le Louvre.

L'arbre de dérivation pour (8) est présenté à la figure 14. Pour la satisfaction des types, les arbres ancré par $Circonstance$ et $Narration$ sont tous deux (A). Le foncteur $\mathcal{R}'_i$ avec $R_i = Circonstance$ ayant comme arguments $\lambda P.P(Narration(F_1, F_2))$ et $\lambda Q.Q(F_0)$, $Narration(F_1, F_2)$ est le premier argument de $Circonstance$, F_0 le second, d'où la formule $Circonstance(Narration(F_1, F_2), F_0)$.

Une remarque : les exemples comme (8) font que, malgré les apparences, la grammaire syntaxique discursive de D-STAG est une TAG qui ne peut être vue comme une TIG (*Tree Insertion Grammar*). Rappelons que la définition d'une TIG impose les contraintes suivantes (Schabes et Waters, 1995) :

– les arbres auxiliaires ne doivent être que des arbres auxiliaires droits ou gauches, en excluant les arbres enveloppants ;

– il est interdit d'adjoindre un arbre droit (resp. gauche) sur le « *spine* » d'un arbre gauche (resp. droit)[10].

Notre grammaire obéit à la première contrainte mais pas à la seconde. En effet, les arbres auxiliaires sont droits pour les connecteurs adverbiaux et les conjonctions postposées, et gauches pour les conjonctions préposées (figures 8 et 14) : il n'existe aucun arbre auxiliaire enveloppant, ce qui respecte la première contrainte. Par contre,

9. La structure de discours ne prend pas en compte l'ordre linéaire du discours. Ainsi, la sémantique de $Circonstance(F_i, F_j)$ est que F_j décrit les circonstances de F_i, quel que soit l'ordre linéaire des segments de discours correspondant à F_i et F_j. En revanche, l'ordre linéaire est pertinent pour la RFC et la définition de cette contrainte dans des discours comme (8) est abordée dans (Danlos et Hankach, 2008).

10. Le « *spine* » d'un arbre auxiliaire est défini comme le chemin de la racine au nœud pied.

pour traiter un exemple comme (8), il faut adjoindre l'arbre droit ancré par *ensuite* sur le *spine* de l'arbre gauche de la conjonction *quand* préposée, ce qui enfreint la seconde contrainte.

3.3. *Modifieurs de connecteurs/relations de discours*

À notre connaissance, la modification des connecteurs/relations de discours est un phénomène qui a été négligé, même en SDRT. Pourtant, c'est un phénomène fréquent, comme illustré en (9).

(9) a. Fred est de mauvaise humeur *seulement/même/sauf* quand il fait beau.
b. Tu ne dois pas faire confiance à Jean parce que, *par exemple*, il ne rend jamais ce qu'il a emprunté. (Exemple traduit de (Webber *et al.*, 2003))
c. Jean s'est cassé le bras. De ce fait, *par exemple*, il ne peut pas conduire. (Exemple traduit de (Webber *et al.*, 2003))

Dans (Webber *et al.*, 2003), l'adverbial *par exemple* n'est pas considéré comme un modifieur de connecteur, mais comme un connecteur, dont l'interprétation est « parasite » de la relation de discours exprimée par le connecteur qui le précède. Cette position, qui n'est pas justifiée, amène à des calculs laborieux pour obtenir l'interprétation d'un discours comme (9b), voir (Forbes-Riley *et al.*, 2006, p. 31-35). À rebours, en D-STAG, nous posons que *par exemple* dans (9b) ou (9c) est un modifieur du connecteur qui le précède au même titre que *seulement, même* et *sauf* dans (9a) sont des modifieurs du connecteur qui les suit. Outre le fait que cette position paraît plus justifiée linguistiquement, elle permet d'obtenir l'interprétation d'un discours comme (9b) de façon très simple, comme nous allons le montrer.

En D-STAG, les modifieurs de connecteur ancrent des arbres (syntaxiques) auxiliaires de nœud pied DC (arbre gauche pour *seulement, même* et *sauf*, arbre droit pour *par exemple*). Pour adjoindre ces arbres, il est nécessaire d'ajouter un lien ⑥ sur le nœud DC dans les arbres syntaxiques de connecteurs (figures 8 et 14). Sur le plan sémantique, nous considérons que la contribution d'un modifieur de relation de discours consiste à transformer un foncteur $\mathcal{R}_i$ de type $\langle t, \langle t, t\rangle\rangle$ en un autre foncteur du même type. De ce fait, les nœuds dominant $\mathcal{R}_i$ portent le lien ⑥ dans les deux copies de (A) et (B). Nous allons illustrer les adjonctions au lien ⑥ pour l'exemple (9b) de forme $C_0\ parce\ que\ par\ exemple\ C_1$. Comme expliqué dans (Webber *et al.*, 2003), l'interprétation de (9b) est $Exemplification(F_1, \lambda r.Explanation(F_0, r))$ avec $r : t$. Pour obtenir cette interpretation, nous donnons au foncteur $\mathcal{P}ar\text{-}ex$ la définition ci-dessous. La paire nommée βpar-ex est donnée à la figure 15, qui montre aussi l'arbre de dérivation pour (9b). Le foncteur $\Phi'(\mathcal{P}ar\text{-}ex\ (\mathcal{R}_i))$ avec $R_i = Explanation$ débouche sur l'interprétation attendue.

Définition 3 $\mathcal{P}ar\text{-}ex = \lambda\mathcal{R}_i pq.Exemplification(q, \lambda r.\mathcal{R}_i(p, r))$
avec $\mathcal{R}_i : \langle t, \langle t, t\rangle\rangle$ et $p, q, r : t$.

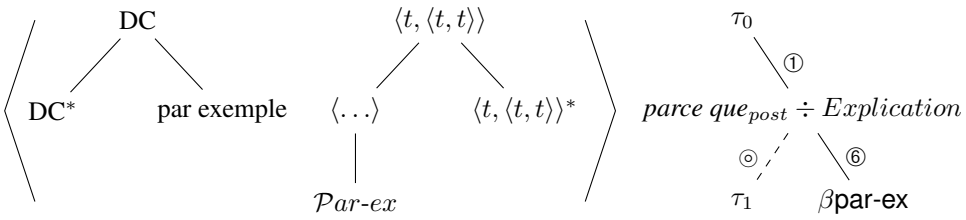


Figure 15. *Paire βpar-ex et arbre de dérivation pour (9)*

3.4. *Conjonctions de coordination*

Pour des raisons qui deviendront évidentes ultérieurement, nous commençons par les coordinations corrélatives de subordonnées adverbiales, qui sont illustrées en (10). Ces coordinations corrélatives se traitent facilement en D-STAG en considérant que les éléments *ni . . . ni*, *soit . . . soit*, *et . . . et* sont des modifieurs de la conjonction de subordination qui les suit. Cette position n'est pas classique dans la mesure où ces éléments sont généralement considérés comme des conjonctions de coordination. Néanmoins, elle permet de calculer, par exemple pour (10a) de forme $C_0\ ni\ quand\ C_1\ ni\ quand\ C_2$, l'interprétation $\neg Circonstance(F_0, F_1) \wedge \neg Circonstance(F_0, F_2)$ sans rien changer à la grammaire STAG discursive : il suffit de donner aux modifieurs *ni* la sémantique de la négation et d'adjoindre la seconde conjonction/relation au lien ③ de la première. Le fait qu'un modifieur comme *ni* ne peut pas modifier une conjonction sans qu'il y ait un autre *ni* qui modifie un autre conjonction – la partie corrélative des constructions en jeu – est pris en compte par un jeu de traits dans les arbres syntaxiques des connecteurs (nous n'avons pas la place de présenter ce jeu de traits). Pour (10b) de forme $C_0\ soit\ si\ C_1\ soit\ si\ C_2$, l'interprétation $Condition(F_0, F_1) \vee Condition(F_0, F_2) = \neg(\neg Condition(F_0, F_1) \wedge \neg Condition(F_0, F_2))$ est obtenue grâce à un arbre sémantique associé au premier *soit* qui comporte deux composants, l'un pour la portée locale de la négation, l'autre pour la portée globale de la négation sur la conjonction de formules[11]. Pour (10c) de forme $C_0\ et\ quand\ C_1\ et\ quand\ C_2$, l'interprétation $Circonstance(F_0, F_1) \wedge Circonstance(F_0, F_2)$ s'obtient à donnant aux modifieurs *et* la sémantique de l'identité.

(10) a. Fred n'est content *ni* **quand** il fait beau *ni* **quand** il pleut.
b. Fred viendra *soit* **s'**il fait beau *soit* **s'**il pleut.
c. Fred est content *et* **quand** il fait beau *et* **quand** il pleut.

Dans (10c), le premier *et* est facultatif. En l'omettant, on obtient un cas de coordination non corrélatif de subordonnées adverbiales, qui correspond à une FND de

11. C'est le seul cas qui demande un arbre à plusieurs composants.

forme $C_0\ quand\ C_1\ et\ quand\ C_2$. Mais cette FND peut être automatiquement convertie en $C_0\ et\ quand\ C_1\ et\ quand\ C_2$. Autrement dit, l'interface phrase-discours peut automatiquement transformer une coordination non corrélative de subordonnées adverbiales en une coordination corrélative. Nous pouvons de ce fait traiter en D-STAG les coordinations non corrélatives de subordonnées adverbiales.

Les coordinations non corrélatives de subordonnées adverbiales constituent le seul cas de coordination qui aurait pu poser problème pour intégrer les conjonctions de coordination dans la grammaire discursive des connecteurs de D-STAG. En effet, les autres cas ne posent pas problème : les coordinations de phrases principales à deux clauses de type $C_0\ Coord_1\ C_1$ s'obtiennent avec un arbre syntaxique ancré par la conjonction de coordination $Coord_1$ similaire à celui ancré par une conjonction de subordination (voir la figure 8). Les coordinations à n clauses avec $n > 2$ de type $C_0,\ C_1(,)\ Coord_2\ C_2$ s'obtiennent par les mécanismes classiques de coordination multiple. En résumé, les conjonctions de coordination sont traitées en D-STAG soit de façon analogue aux conjonctions de subordination pour les coordinations de principales soit comme des modifieurs de conjonctions de subordination pour les coordinations de subordonnées adverbiales en ramenant les coordinations non corrélatives à des coordinations corrélatives.

3.5. *Conclusion sur la grammaire* STAG *discursive*

Nous avons présenté une grammaire STAG discursive qui traite de façon exhaustive les FND comportant des connecteurs de catégorie adverbiale ou conjonction de subordination et coordination, ces connecteurs étant éventuellement modifiés. Cette grammaire est de petite taille. Ainsi, elle ne comporte qu'une dizaine de schémas d'arbres syntaxiques. Un schéma d'arbre est ancré par une catégorie syntaxique (Adv_i) et non par un item lexical de cette catégorie (*ensuite*). Nous avons considéré que les connecteurs d'une catégorie donnée se comportaient tous de la même façon, mais il est possible de prendre en compte les particularités d'un connecteur donné, par exemple le fait que la conjonction *comme* n'ancre qu'un seul arbre, celui d'une conjonction préposée. De même pour les arbres sémantiques et les particularités d'une relation de discours donnée.

Rappelons (section 2.3) que cette grammaire demande à être étendue. Cette extension ne devrait pas soulever de problème insurmontable vu la puissance d'expressivité offerte par STAG.

4. Comparaison entre D-STAG et D-LTAG

D-STAG et D-LTAG (Forbes-Riley *et al.*, 2006) ont *grosso modo* le même objectif et partagent la même architecture (section 1). Les divergences entre ces deux formalismes

se situent principalement dans la partie discursive[12]. D'abord, D-LTAG utilise peu les relations de discours, ignore la distinction entre relations coordonnantes *versus* subordonnantes. Bref, ce formalisme n'est pas élaboré à partir de théories discursives. C'est même un principe, comme en témoigne la citation suivante (Forbes-Riley *et al.*, 2006, p. 1) (les italiques sur « same » sont des auteurs) : « D-LTAG *presents a model of low-level discourse structure and interpretation that exploits the* same *mechanisms used at the sentence level and builds them directly on top of clause structure and interpretation*. » [13]. Ceci empêche D-LTAG de bénéficier des résultats apportés par les théories sur le discours qui apportent, entre autres, des connaissances rhétoriques. Par exemple, D-LTAG ne peut pas faire appel à la RFC qui contraint fortement l'attachement d'informations nouvelles.

Ensuite, en D-STAG, les connecteurs de discours ancrent des arbres élémentaires qui ont tous **deux** arguments, tandis qu'en D-LTAG la plupart des connecteurs adverbiaux (mais pas le connecteur vide ϵ) ancrent des arbres (auxiliaires) avec **un seul** argument (c'est le cas, par exemple, pour *next/ensuite* qui est considéré comme un connecteur «anaphorique » dont le segment hôte est fourni « structurellement », le segment convié « anaphoriquement » (Webber *et al.*, 2003 ; Webber, 2004)). De plus, les conjonctions de subordination ancrent en D-LTAG des arbres avec deux arguments, mais ces arbres sont **initiaux** tandis qu'ils sont **auxiliaires** en D-STAG (les conjonctions de subordination sont considérées en D-LTAG comme des connecteurs « structuraux » dont les segments hôte et convié sont fournis structurellement, en passant sous silence le fait qu'un de leurs arguments peut dépasser une frontière de phrase). Ces différences sur les arbres ancrés par un connecteur induisent des différences majeures sur l'analyse syntaxique et surtout sur la structure de discours. À titre d'illustration, l'arbre syntaxique et l'arbre de dérivation pour (1c), répété en (11), produits par une adaptation française de D-LTAG sont donnés à la figure 16.

(11) Fred est allé au supermarché parce que son frigo était vide. Ensuite, il est allé au cinéma.

L'arbre syntaxique comporte trois nœuds étiquetés DC, un pour *parce que*, un pour ϵ, et un pour *ensuite*. Il est différent de celui calculé en D-STAG qui ne comporte que deux nœuds étiquetés DC, car le connecteur vide ϵ n'est introduit qu'en l'absence d'un autre connecteur adverbial. L'arbre de dérivation débouche sur la structure de discours $Narration(Explication(F_0, F_1), F_2)$, qui n'est pas correcte : l'explication fournie pour la visite de Fred au supermarché (*son frigo était vide*) ne doit

12. Les composants phrastiques de D-STAG et D-LTAG sont aussi différents, car D-STAG repose sur STAG, ce qui n'est pas le cas pour D-LTAG. Néanmoins, le niveau phrastique sort du cadre de cet article et nous invitons le lecteur à lire (Nesson et Shieber, 2006) pour une discussion sur les différentes approches utilisées pour réaliser une interface syntaxe-sémantique phrastique.

13. « D-LTAG présente un modèle pour la structure discursive de bas niveau et pour l'interprétation discursive qui exploite les *mêmes* mécanismes que ceux utilisés au niveau phrastique et qui calcule structures et interprétations discursives directement à partir des structures et interprétations clausales. »

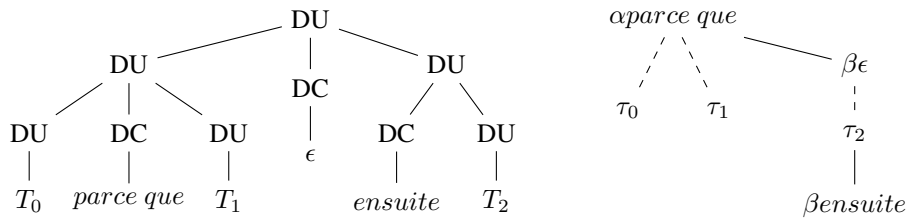


Figure 16. *Arbre syntaxique et arbre de dérivation pour (1c) = (11) en* D-LTAG

pas être sous la portée de $Narration$. Comme expliqué dans la section 2.1, l'interprétation est $Explication(F_0, F_1) \wedge Narration(F_0, F_2)$. Cette interprétation, qui correspond à un graphe de dépendances non arborescent (section 2.1), ne peut pas être obtenue directement en D-LTAG. Néanmoins, il peut être argué que la structure $Narration(Explication(F_0, F_1), F_2)$ est à interpréter avec le principe de nucléarité (section 2.1), ce qui débouche sur l'interprétation voulue. Mais rappelons que le principe de nucléarité exclut de pouvoir donner une interprétation correcte à un discours comme (1d), à savoir une interprétation de type $R_2(R_1(F_0, F_1), F_2)$ où la relation subordonnante R_1 sert à former un constituant complexe qui est le premier argument de R_2. De plus, les chercheurs travaillant dans le cadre de D-LTAG ne se positionnent pas clairement (à notre connaissance) sur le fait qu'ils appliquent ou non le principe de nucléarité. Il en est de même pour les chercheurs travaillant dans le cadre du *Penn Discourse Tree Bank* (PDTB Group, 2008), les deux communautés de chercheurs ayant d'ailleurs une forte intersection. Ainsi, (Lee *et al.*, 2008) présentent des statistiques fort étonnantes pour les quatre interprétations des discours de forme $C_0\ Conj_1\ C_1.\ Adv_2\ C_2$, où $Conj_1$ est une des douze conjonctions de subordination les plus usitées en anglais. Ils indiquent, à partir d'observations faites sur le PDTB , que la première structure $R_2(R_1(F_0, F_1), F_2)$ s'observe dans 83,4 % des cas, la deuxième $R_1(F_0, F_1) \wedge R_2(F_0, F_2)$ dans 12,4 % des cas, la troisième $R_1(F_0, F_1) \wedge R_2(F_1, F_2)$ et la quatrième $R_1(F_0, R_2(F_1, F_2))$ dans 4,2 % des cas[14]. L'écrasante majorité de la première structure nous étonne énormément (en considérant que $Conj_1$ exprime une relation subordonnante dont le nucleus est C_0[15]) :

– si cette structure n'est pas à interpréter avec le principe de nucléarité, ceci revient à dire qu'une structure exclue par RST correspond à 83,4 % des cas (ou un peu moins si on soustrait les quelques cas où la conjonction $Conj_1$ exprime une relation coordonnante,

14. (Lee et al., 2008) distinguent bien les deux dernières structures, mais ne donnent qu'un pourcentage global pour les deux.

15. (Matthiessen et Thompson, 1988) ont posé une relation étroite entre subordination syntaxique et subordination discursive. Toutefois, cette position doit plutôt être vue comme une tendance que comme un principe. En effet, il existe quelques contre-exemples, comme ceux mis en avant dans (Delort, 2006) pour la conjonction *avant (que/de)*.

voir note 15). Ceci serait surprenant, d'autant plus que nous pensons que des exemples comme (1d) ne sont pas si fréquents ;

– on est donc amené à penser que cette structure doit être interprétée avec le principe de nucléarité. Ceci revient à dire qu'elle débouche sur la même interprétation que la deuxième structure, $R_1(F_0, F_1) \wedge R_2(F_0, F_2)$. On peut alors se demander quand et comment les annotateurs ont choisi une structure plutôt que l'autre, sachant que seule l'interprétation détermine les arguments des connecteurs dans des discours de forme $C_0\ Conj_1\ C_1.\ Adv_2\ C_2$.

En résumé, en D-LTAG comme dans le PDTB, on retrouve la confusion introduite en RST par le fait qu'il faut distinguer structures de discours et interprétation de ces structures (voir note 2).

Une autre différence entre D-STAG et D-LTAG s'observe avec le traitement de la modification des connecteurs de discours (section 3.3). Rappelons que D-LTAG ignore la modification des connecteurs et traite *par exemple* en (9b) comme un connecteur, ce qui conduit à des calculs laborieux pour aboutir à la représentation sémantique. En D-LTAG, nous traitons les modifieurs de connecteurs par l'intermédiaire d'arbres auxiliaires, en suivant la solution classique préconisée en TAG pour les modifieurs. D'une manière plus générale, disons que D-STAG profite beaucoup plus que D-LTAG des possibilités offertes par l'adjonction, opération qui fait la richesse du formalisme TAG.

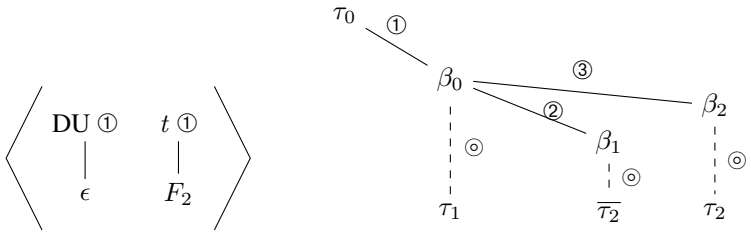
Une motivation majeure dans la conception de D-LTAG a été le cas des « connecteurs multiples », citons (Webber *et al.*, 2003, p 552) : « *The distinction between structural and anaphoric connectives in* D-LTAG *is based on considerations of computational economy and behavioral evidence from cases of multiple connectives.* »[16]. Le cas des connecteurs multiples est illustré en (12), dont la FND est de la forme $C_0.\ Mais\ C_1\ parce\ que\ ensuite\ C_2$, où deux connecteurs de discours partagent la même clause hôte, C_2.

(12) Jean a commandé trois caisses de Barolo. Mais il a dû annuler cette commande parce qu'ensuite il s'est rendu compte qu'il était fauché.
[traduit de (Webber *et al.*, 2003)]

À rebours, nous ne pensons pas que les cas de connecteur multiple comme (12) doivent remettre en question les connaissances fondamentales acquises sur les connecteurs de discours. En effet, nous pouvons avancer une solution simple pour les traiter en D-STAG. Cette solution repose sur l'étape intermédiaire, à savoir sur l'interface phrase-discours qui construit la forme normalisée du discours donné en entrée. Nous proposons que la FND pour (12) soit automatiquement convertie en la FND $C_0.\ Mais\ C_1\ parce\ que\ \overline{C_2}\ ensuite\ C_2$ qui suit le modèle général des

16. « La distinction entre connecteurs structuraux et anaphoriques est fondée sur des considérations d'économie de calcul et de comportements linguistiques à partir des cas de connecteurs multiples. »

FND où une clause hôte n'accueille qu'un seul connecteur. La paire d'arbres $\overline{\tau_2}$ associée à $\overline{C_2}$, donnée à la figure 17-a, est construite de la façon suivante : la feuille syntaxique est la chaîne vide ϵ, la feuille sémantique est F_2, la formule sémantique pour C_2. On peut donc dire que $\overline{C_2}$ est la (fausse) clause hôte de *parce que*, qui n'a qu'une contribution sémantique. L'interprétation de (12), *i.e.* $Contraste(F_0, F_1) \wedge Explanation(F_1, F_2) \wedge Narration(F_0, F_2)$, est obtenue par la procédure décrite dans la section 3.1 qui construit l'arbre de dérivation donné dans la figure 17-b (avec $\beta_0 = mais \div Contraste$, $\beta_1 = parce\ que \div Explanation$ et $\beta_2 = ensuite \div Narration$)[17].



(a) $\overline{\tau_2}$ (b) *Arbre de drivation pour* (12)

Figure 17. *Paire d'arbres $\overline{\tau_2}$ et arbre de dérivation pour (12)*

En conclusion, le cas des connecteurs multiples peut être ramené au cas général où une clause hôte n'accueille qu'un seul connecteur, grâce à l'interface phrase-discours qui construit les FND[18]. Il n'y a donc pas lieu de poser une distinction entre connecteurs anaphoriques et connecteurs structuraux, les premiers ancrant des arbres auxiliaires à un seul argument[19], les seconds des arbres initiaux à deux arguments, comme cela est postulé en D-LTAG. La grammaire discursive de D-STAG montre qu'il est justifié de postuler qu'il n'existe qu'un seul type de connecteur et que tout connecteur ancre un arbre auxiliaire à deux arguments. Le nœud pied DU* de ces arbres auxiliaires correspond au segment convié qui est fourni anaphoriquement si on veut reprendre

17. L'interprétation de (12) est un contre-exemple apparent à la RFC (section 3.1.5) : $Contraste$ étant une relation coordonnante, son premier argument représenté par F_0 ne devrait pas être ouvert pour attacher l'information nouvelle F_2 via $Narration(F_0, F_2)$ (rappelons que la structure $R_1(F_0, F_1) \wedge R_i(F_0, F_i)$ est interdite lorsque R_1 est coordonnante). Néanmoins, on peut considérer que l'argument F_2 de $Narration$ n'est pas une information nouvelle dans la mesure où cette information a déjà été introduite comme argument de $Explication$. Techniquement, en D-STAG on peut ajouter le trait $[info - nouvelle = -]$ sur la racine de l'arbre sémantique de τ_2, puis propager ce trait dans l'arbre sémantique de β_2 de façon à ce que l'adjonction de cet arbre au lien ③ de β_0 (avec une relation coordonnante) ne soit pas bloquée.

18. Rappelons qu'une telle interface existe aussi en D-LTAG et que nous nous en sommes inspirée. Il aurait donc aussi été possible en D-LTAG de ramener le cas des connecteurs multiples au cas général.

19. Les connecteurs anaphoriques sont des connecteurs adverbiaux, ce qui à relier au fait que lorsque deux connecteurs se partagent la même phrase hôte, le second est obligatoirement un connecteur adverbial.

ce terme, le nœud à substitution DU↓ correspond au segment hôte qui commence structurellement à la clause hôte du connecteur sans être forcément identique à cette clause hôte.

5. Conclusion et perspectives : vers l'implémentation de D-STAG pour le français

Rappelons que l'analyse en D-STAG se déroule en trois étapes répétées ci-dessous :

1) analyses syntaxique et sémantique au niveau phrastique ;

2) interface phrase-discours qui produit une FND, *i.e.* une suite de mots de discours ;

3) analyse S-TAG au niveau discursif de cette suite de mots de discours.

Concernant la première étape, il existe plusieurs analyseurs syntaxiques du français (fondés sur TAG ou d'autres formalismes) mais aucun analyseur sémantique (obtenu par une S-TAG ou autrement)[20]. De plus, les analyseurs syntaxiques ne donnent pas de bons résultats pour la segmentation des phrases complexes en clauses, alors que c'est un point crucial pour la seconde étape[21]. Nous nous attaquons quand même à l'implémentation d'un analyseur D-STAG du français sans attendre de meilleurs résultats au niveau phrastique, avec deux chantiers.

D'une part, avec Charlotte Roze, nous avons réalisé une base lexicale de connecteurs du français, appelée LEXCONN (Roze, 2009). Cette base lexicale rassemble un ensemble de conjonctions et d'adverbiaux qui nous ont été fournis par Éric Laporte et Benoît Sagot. Pour chacun de ces éléments, nous avons examiné si c'est effectivement un connecteur, et si c'est le cas, nous avons déterminé quelle(s) relation(s) de discours exprime le connecteur et la nature coordonnante ou subordonnante des relations. Ce travail a été effectué en étroite collaboration avec les chercheurs toulousains qui ont aussi réalisé une base lexicale de connecteurs dans le cadre d'ANNODIS (Péry-Woodley *et al.*, 2009)[22]. Leur base compte 64 entrées, la nôtre 370. Ces bases sont destinées à être fusionnées.

20. (Gardent, 2008) présente un effort dans cette direction, toutefois limité à la sémantique des verbes.

21. Nous avons constaté cette triste réalité sur FRMG, métagrammaire TAG (Villemonte de La Clergerie, 2005), les chercheurs toulousains travaillant sur ANNODIS (voir note 13) l'ont constatée sur l'analyseur Synlex (Bourigault, 2007). Elle s'explique par le fait que, dans les campagnes d'évaluation des analyseurs syntaxiques, notamment Easy et Passage, les métriques utilisées donnent la même importance aux relations (dépendances) à portée courte et à celles à portée large, les frontières de clauses se situant parmi les secondes. Or les premières sont beaucoup plus nombreuses, et contribuent donc de façon plus importante aux scores obtenus par les analyseurs. De plus, celles à portée large sont plus difficiles à calculer correctement, et par là même moins bien prises en compte dans la plupart des analyseurs syntaxiques.

22. Le projet ANNODIS a pour objectif d'annoter un corpus français pour les relations de discours et leurs arguments, voir `http://w3.erss.univ-tlse2.fr:8080/index.jsp?perso=annodis&subURL=ANNODISfr.html`.

D'autre part, au sein de l'équipe ALPAGE, nous travaillons à la réalisation d'un système de segmentation de phrases en clauses. À notre connaissance, il n'existe aucun segmenteur de phrases pour le français sauf le système SIGLé – Système d'Identification de propositions avec une Grammaire Légère (Nakamura-Delloye, 2006) – qui repose sur une analyse syntaxique superficielle et qui exploite des connaissances linguistiques sur les délimiteurs de propositions finies (complémenteurs, pronoms relatifs, conjonctions de subordination). Un tel système va être enrichi par des techniques d'apprentissage et des méthodes probabilistes. S'il donne de bons résultats, il pourra servir pour guider l'analyse syntaxique des phrases. La production de FND à partir d'un segmenteur de phrases en clauses ne devrait pas soulever de problèmes drastiques grâce aux connecteurs adverbiaux répertoriés dans LEXCONN. Indiquons que la production automatique de FND ouvre une avenue de recherche pour les linguistes travaillant sur les connecteurs de discours et souhaitant fonder leurs recherches sur des corpus réels. A l'heure actuelle, ces linguistes sont obligés, par exemple, de passer par des grammaires locales qui recherchent des occurrences d'un connecteur donné dans des bases de données textuelles, ce qui donne lieu à beaucoup de bruit. La production automatique de FND simplifiera grandement leur travail. Nous comptons nous-mêmes mener des études de linguistique discursive sur corpus. Ainsi, nous projetons d'approfondir l'étude amorcée dans (Danlos et Hankach, 2008) sur les subordonnées adverbiales préposées (section 3.2). Nous voulons aussi étudier linguistiquement les cas où deux connecteurs se partagent la même phrase hôte (section 4).

Lorsque nous disposerons d'un système de production automatique de FND, nous pourrons nous attaquer à la troisième étape, la partie discursive de D-STAG. La grammaire STAG sera facile à écrire grâce aux paires d'arbres décrites à la section 3 et à LEXCONN. Toutefois, elle nécessitera d'être complétée, par exemple, pour prendre en compte la relation $Attribution$ (voir section 2.3). Pour l'analyseur, nous devrons nous contenter d'un analyseur asynchrone qui ne produit que l'analyse syntaxique discursive (en utilisant un analyseur TAG standard). Rappelons cependant que l'on peut transférer les informations sur la nature coordonnante ou subordonnante des relations de discours des arbres sémantiques vers les arbres syntaxiques, et donc prendre en compte la RFC ou les autres contraintes sémantiques au niveau syntaxique, ce qui limite le nombre d'analyses possibles (voir section 3.1.5).

L'analyseur syntaxique discursif va produire une forêt d'arbres de dépendances, qui représente l'ensemble des analyses possibles, avec une grande inconnue : quelle est l'étendue du désastre ? Est-ce que la situation est pire ou meilleure que pour l'analyse syntaxique phrastique (avec une forêt qui contient un nombre d'arbres dont l'ordre de grandeur est 2^n pour une phrase de longueur n) ? Soulignons qu'aucun analyseur discursif n'ayant été réalisé pour le français à grande échelle, la réponse à cette question est une véritable inconnue. Il restera à faire le travail d'extraction de la meilleure analyse (ou des n meilleures analyses) à partir de cette forêt d'analyses. Ceci demandera d'identifier des indices utilisables informatiquement permettant de construire des modèles probabilistes de désambiguïsation, à l'image de ce dont on dispose au niveau phrastique. Le corpus annoté manuellement dans le cadre du projet ANNODIS sera fort utile pour cette tâche.

En conclusion, il est envisageable à moyen terme de disposer d'un analyseur discursif du français, cet analyseur implémentant le formalisme D-STAG présenté dans cet article. Ce formalisme repose sur une théorie du discours (SDRT), sur des bases linguistiques sérieuses, et sur le formalisme STAG qui a une grande puissance d'expressivité et dont la complexité informatique est maîtrisée.

Remerciements

Je remercie chaleureusement les relecteurs anonymes de la revue TAL, ainsi que Pascal Denis, Sylvain Pogodalla et Benoît Sagot dont les commentaires ont été très fructueux et qui m'ont beaucoup aidée dans ce travail.

6. Bibliographie

Asher N., *Reference to Abstract Objects in Discourse*, Kluwer, Dordrecht, 1993.

Asher N., Lascarides A., *Logics of Conversation*, Cambridge University Press, Cambridge, 2003.

Asher N., Vieu L., « Subordinating and Coordinating Discourse Relations », *Lingua*, vol. 115, n° 4, p. 591-610, 2005.

Bourigault D., *SYNTEX : analyseur syntaxique pour le français*, Dossier d'HDR, Université de Toulouse le Mirail, 2007.

Bras M., *Entre relations temporelles et relations de discours*, Dossier d'HDR, Université de Toulouse le Mirail, 2008.

Charolles M., « Framing adverbials and their role in discourse cohesion », *Proceedings of SEM-05*, Biarritz, p. 194-201, 2005.

Danlos L., « G-TAG : un formalisme lexicalisé pour la génération de textes inspiré de TAG », *Revue TAL*, 1998.

Danlos L., « Discourse dependency structures as constrained DAGs », *Proceedings of SIGDIAL'04*, Boston, p. 127-135, 2004a.

Danlos L., « Sentences with two subordinate clauses: syntactic and semantic analyses, underspecified semantic representation », *Proceedings of TAG+7*, Vancouver, p. 140-147, 2004b.

Danlos L., « Capacité générative forte de RST, SDRT et des DAG de dépendances pour le discours », *Revue TAL*, 2006.

Danlos L., Hankach P., « Right Frontier Constraint for Discourses in Non Canonical Order », *Proceedings of the Constraints in Discourse Workshop (CID'08)*, Postdam, Germany, 2008.

Delort L., « Clause 'Subordination' and Discourse Relations », *Proceedings of the 28th Annual Meeting of the German Society for Linguistics (DGfS-06), Workshop on Subordination vs. Coordination in Sentence and Text from a Cross-linguistic Perspective*, Bielefeld, Germany, 2006.

Forbes-Riley K., Webber B., Joshi A., « Computing Discourse Semantics: The Predicate-Argument Semantics of Discourse Connectives in D-LTAG », *Journal of Semantics*, 2006.

Gardent C., « Integrating a unification-based semantics in a large scale Lexicalised Tree Adjoining Grammar for French », *Proceedings of COLING'08*, Manchester, UK, 2008.

Harris Z., *Mathematical Structures of Language*, Krieger Pub co, New York, 1986.

Joshi A., « Tree-adjoining grammars », *in* D. Dowty, L. Karttunen, A. Zwicky (eds), *Natural language parsing*, Cambridge University Press, p. 206-250, 1985.

Lee A., Prasad R., Joshi A., Webber B., « Departures from Tree Structures in Discourse: Shared Arguments in the Penn Discourse Tree Bank », *Proceedings of the Constraints in Discourse Workshop (CID'08)*, Postdam, Germany, 2008.

Mann W. C., Thompson S. A., « Rhetorical Structure Theory : Toward a Functional Theory of Text Organization », *Text*, vol. 8, n° 3, p. 243-281, 1988.

Marcu D., « The Rhetorical Parsing of Unrestricted Texts: A Surface-Based Approach », *Computational Linguistics*, vol. 26, n° 3, p. 395-448, 2000.

Matthiessen C., Thompson S., « The Structure of Discourse and 'Subordination' », *in* J. Haiman, S. Thompson (eds), *Clause Combining in Grammar and Discourse*, vol. 18 of *Typological Studies in Language*, John Benjamins, Amsterdam/Philadelphia, p. 275-329, 1988.

Nakamura-Delloye Y., « Détection des propositions syntaxiques du français », *Actes de TALN'06*, Leuven, Belgique, 2006.

Nesson R., Shieber S., « Simpler TAG semantics through Synchronization », *Formal Grammars*, Malaga, 2006.

PDTB Group, The Penn Discourse Treebank 2.0 Annotation Manual, Technical report, Institute for Research in Cognitive Science, University of Philadelphia, 2008.

Péry-Woodley M.-P., Asher N., Enjalbert P., « ANNODIS: une approche outillée de l'annotation de structures discursives », *Proceedings of TALN'09*, Senlis, France, p. 190-196, 2009.

Prasad R., Dinesh N., Lee A., Joshi A., Webber B., « Attribution and its annotation in the Penn Discourse TreeBank », *Revue TAL*, 2006.

Roze C.*, « LEXCONN : Base lexicale des connecteurs discursifs du français »*, Mémoire de Master, Université Paris 7, 2009.

Schabes Y., Waters R., « Tree Insertion Grammar », *Computational Intelligence*, vol. 21, p. 479-514, 1995.

Shieber S., « Restricting the weak-generative capacity of synchronous tree-adjoining grammars », *Computational Intelligence*, vol. 10, n° 4, p. 371-385, 1994.

Shieber S., Schabes Y., « Synchronous tree-adjoining grammars », *Proceedings of the 13th International Conference on Computational Linguistics*, vol. 3, Helsinki, p. 253-258, 1990.

Villemonte de La Clergerie E., « From Metagrammars to Factorized TAG/TIG Parsers », *Proceedings of the Fifth International Workshop on Parsing Technology (IWPT'05)*, Vancouver, Canada, p. 190-191, 2005.

Webber B., « D-LTAG: extending lexicalized TAG to discourse », *Cognitive Science*, vol. 28, n° 5, p. 751-779, 2004.

Webber B., Joshi A., Stone M., Knott A., « Anaphora and Discourse Structure », *Computational Linguistics*, vol. 29, n° 4, p. 545-587, 2003.

Wolf F., Gibson E., *Coherence in Natural Language: Data Structures and Applications*, The MIT Press, London, 2006.