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

Laurence Danlos

ALPAGE

Université Paris 7

Institut Universitaire de France

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 sentential 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 Grammars, 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. On sait en effet qu'un discours, c'est-à-dire une succession cohérente de phrases, ne se réduit pas à une simple suite de phrases : un discours présente une organisation interne où les diverses informations apportées par les phrases sont agencées en suivant des principes rhétoriques. Les théories du discours représentent cette organisation sous forme de structures hiérarchiques où des segments de discours sont reliés par des relations de discours, et ce récursivement.

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 ils ne sont pas à même de fournir une implémentation robuste, efficace et à grande échelle d'un analyseur de textes calculant la structure de 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 et passage du niveau phrastique au niveau discursif. Le passage au niveau discursif a concerné d'abord la génération automatique de textes avec G-TAG (Danlos, 1998), puis l'analyse automatique de textes avec D-LTAG — Discourse Lexicalized TAG (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 le même objectif d'analyse automatique du discours. Ainsi D-STAG et D-LTAG reposent sur la même architecture avec trois composants :

- 1) l'analyse phrastique, qui produit pour chaque phrase du discours donnée en entrée une analyse syntaxique et sémantique ;
- 2) l'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) l'analyse discursive, qui calcule une formule logique représentant la structure de discours.

Pour la troisième étape, D-STAG propose une amélioration par rapport à D-LTAG principalement sur deux points, qui sont d'ailleurs reliés¹. Le premier point est que D-STAG s'appuie sur une théorie de discours, contrairement à D-LTAG où les analyses discursives sont vues comme une simple prolongation des analyses phrastiques, en négligeant les connaissances rhétoriques et pragmatiques nécessaires pour une bonne compréhension de l'organisation interne d'un texte. Le second point est le fait que D-STAG peut produire des structures de discours qui correspondent à des graphes de dépendances non arborescents, alors que D-LTAG se cantonne à des graphes de dépendances arborescents.

Cet article est organisé comme suit. La Section 2 présente les pré-requis théoriques, linguistiques et formels sur lesquels D-STAG s'appuie : d'abord une brève introduction à SDRT, ensuite un résumé des données linguistiques sur les connecteurs et leurs arguments, enfin 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 plus en détail D-STAG et D-LTAG. Enfin, nous concluons sur les perspectives d'implémentation de D-STAG pour le français.

2. Pré-requis théoriques, linguistiques et formels

2.1. Brève introduction à SDRT

SDRT procède en deux étapes pour interpréter un discours. La première étape consiste à construire la forme logique de la structure de discours qui est représentée par une SDRS — Segmented Discourse Representation Structure. Une SDRS est une structure hiérarchique où un ensemble de constituants sémantiques sont interconnectés par des relations de discours. Cette étape repose sur une logique (non monotone) de « l'emballage de l'information ». La seconde étape consiste à interpréter dans un modèle ce qui a été dit dans le discours grâce à la logique du « contenu de l'information ». D-STAG ne s'intéresse qu'à la première étape : l'objectif de ce formalisme est de calculer la forme logique de la structure de discours sous une forme compatible avec celle utilisée en SDRT.

Une relation de discours est ou non lexicalisée par un connecteur de discours. L'ensemble des connecteurs de discours comprend la plupart des 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.*). Lorsqu'une

1. D-STAG est illustré pour le français, tandis que D-LTAG l'est pour l'anglais, mais on pourrait intervertir les langues.

relation de discours n'est pas lexicalisée par un connecteur, comme c'est le cas par exemple pour la relation *Explication* dans « *Fred est tombé. Marc lui a fait un croche-pied.* » qui doit être inférée sur la base de connaissances linguistiques et extralinguistiques, nous posons l'existence d'un connecteur adverbial vide noté ϵ , suivant en cela la position de (Harris, 1986) et celle de D-LTAG. De ce fait, nous posons que le discours « *Fred est tombé. Marc lui a fait un croche-pied.* » est de forme « $P_1 . \epsilon P_2 .$ », et par abus de langage nous disons que le connecteur vide « lexicalise » *Explication*.

SDRT pose qu'il existe deux types de relations de discours, les relations coordonnantes et subordonnantes. 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 Nucleus et les Satellites. Ces distinctions sont équivalentes si on pose qu'une relation coordonnante relie deux Nucleus tandis qu'une relation subordonnante relie un Nucleus et un Satellite. Un Nucleus est un élément essentiel du discours tandis qu'un Satellite est de moindre importance. La distinction entre relations coordonnantes et subordonnantes ou entre Nucleus et Satellite suscite des débats en SDRT (Asher et Vieu, 2005) et en RST (Stede, 2007). Toutefois, en D-STAG, nous utilisons la distinction entre relations coordonnantes et subordonnantes (pour la contrainte de la frontière droite expliquée ci-dessous), et nous supposons que le statut coordonnant ou subordonnant d'une relation de discours donnée est clair et fixe.

Les SDRS reçoivent deux types de représentation, une représentation en boîte à la DRS, et une représentation sous forme de graphe. La première est plus pratique pour calculer la forme logique d'un discours, la seconde pour calculer la frontière droite. Ces représentations sont illustrées à la Figure 1 pour le discours (1) de forme « C_1 parce que C_2 . Ensuite C_3 . », où le symbole C_i identifie la i ème clause. Dans la représentation en boîte, la notation K_{π_i} symbolise la forme logique de la i ème clause, π_i étant une étiquette pour cette forme logique.

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

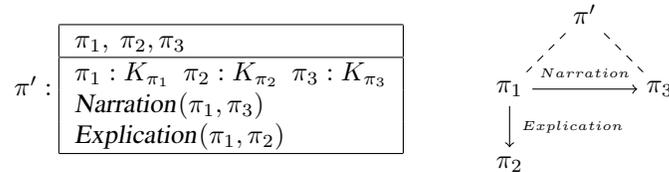


Figure 1. Représentation en boîte et graphe de la SDRS pour le discours (1)

Dans la représentation en boîte, les différentes lignes comportant une relation de discours sont à interpréter comme des conjonctions de formules : la représentation sémantique de (1) — sa structure discursive — est $\text{Explication}(\pi_1, \pi_2) \wedge$

$Narration(\pi_1, \pi_3)$. *Explication* est une relation subordonnante représentée par une flèche verticale dans le graphe de la SDRS, *Narration* une relation coordonnante représentée par une flèche horizontale. La représentation sémantique $Explication(\pi_1, \pi_2) \wedge Narration(\pi_1, \pi_3)$ se glose de la façon suivante : l'événement de Fred allant au cinéma — représenté dans π_3 — s'enchaîne narrativement sur l'événement de Fred allant au supermarché — représenté dans π_1 —, ces deux événements ayant le même degré d'importance ; la seconde clause (*son frigo était vide*) représentée dans π_2 apporte une information non essentielle, elle explique la visite de Fred au supermarché.

Lors de la procédure incrémentale de construction des structures de discours, SDRT fait appel à la « Contrainte de la Frontière Droite » abrégé en RFC (pour Right Frontier Constraint). Cette contrainte stipule que seuls les nœuds sur la frontière droite du graphe — π_3 et π' dans la Figure 1 — sont ouverts pour l'attachement d'une information nouvelle. Outre le respect de l'ordre linéaire du discours, cette contrainte postule que le premier argument d'une relation coordonnante n'est pas ouvert pour l'attachement d'une information nouvelle. Ainsi, dans la Figure 1, le nœud π_1 — premier argument de la relation coordonnante *Narration* — n'est pas ouvert pour l'attachement d'une nouvelle information qui viendrait d'une quatrième clause. 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 3.1.1.3 présentera l'implémentation de la RFC.

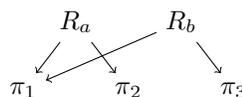
2.2. Données linguistiques et graphes de dépendances pour les structures discursives

SDRT utilise ses propres représentations graphiques, mais la question se pose de savoir à quoi correspondent les structures discursives représentées en graphes de dépendances, dans lesquels une relation de discours domine ses deux arguments. L'idée est largement répandue que les graphes de dépendances des structures discursives ont une structure arborescente : 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. Ce principe a guidé la conception de D-LTAG qui ne peut calculer que des structures arborescentes. Pourtant, cette structure arborescente est un mythe : nous avons montré dans une série de travaux (Danlos, 2004a ; Danlos, 2004b ; Danlos, 2006) que les graphes de dépendances des structures de discours sont des DAG non forcément arborescents, mais respectant cependant des contraintes fortes qui éliminent bon nombre de DAG. Nous allons brièvement montrer ce point sur des discours de forme C_1 parce que C_2 . Adv_b C_3 . (exemple que nous reprendrons quand nous expliquerons la partie discursive de D-STAG, voir Section 3.1.1.2). Ces discours peuvent recevoir quatre types d'interprétation — mais pas plus que quatre — qui sont illustrées dans les exemples de (2).

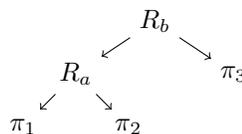
- (2)a. Fred est allé au supermarché parce que son frigo était vide. Ensuite, il est allé au cinéma.

- b. Fred est de mauvaise humeur parce que sa femme est absente pour une semaine. Ceci prouve qu'il l'aime vraiment beaucoup.
- c. Fred est de mauvaise humeur parce qu'il a mal dormi. Il a fait des cauchemars.
- d. Fred est de mauvaise humeur parce qu'il a mal dormi. Il a aussi raté son permis de conduire.

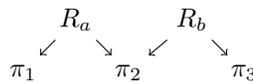
Dans (2a) (= (1)), le premier argument de $Adv_b = \textit{ensuite}$, qui lexicalise la relation *Narration*, est C_1 . Rappelons que la structure de discours est $\textit{Explication}(\pi_1, \pi_2) \wedge \textit{Narration}(\pi_1, \pi_3)$, ce qui correspond à un DAG de dépendances non arborescent, voir ci-contre avec $R_a = \textit{Explication}$ et $R_b = \textit{Narration}$.



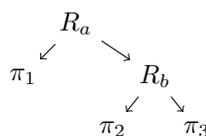
Dans (2b), le premier argument de $Adv_b = \epsilon$, qui lexicalise ici la relation *Commentaire*, est le constituant complexe C_1 parce que C_2 . La structure de discours est $\textit{Commentaire}(\textit{Explication}(\pi_1, \pi_2), \pi_3)$ ce qui correspond à un DAG de dépendances arborescent, voir ci-contre avec $R_b = \textit{Explication}$. Signalons qu'en SDRT, $\textit{Commentaire}(\textit{Explication}(\pi_1, \pi_2), \pi_3)$ s'écrit $\textit{Commentaire}(\pi', \pi_3)$ avec $\pi' : \textit{Explication}(\pi_1, \pi_2)$.



Dans (2c), le premier argument de $Adv_b = \epsilon$, qui lexicalise ici la relation *Explication*, est C_2 . La structure de discours est $\textit{Explication}(\pi_1, \pi_2) \wedge \textit{Explication}(\pi_2, \pi_3)$, ce qui correspond à un DAG de dépendances non arborescent, voir ci-contre avec $R_b = \textit{Explication}$.



Dans (2d), le premier argument de $Adv_b = \textit{aussi}$, qui lexicalise la relation *Parallèle*, est C_2 ; Adv_b sert à former un constituant complexe qui est le second argument de *parce que*. La structure de discours est $\textit{Explication}(\pi_1, \textit{Parallèle}(\pi_2, \pi_3))$, ce qui correspond à un DAG de dépendances arborescent, voir ci-contre avec $R_b = \textit{Parallèle}$. Dans cette structure discursive, le second argument de la conjonction *parce que* dépasse une frontière de phrase.



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. Toutefois, d'une part, les auteurs de D-LTAG ont toujours reconnu qu'il existait des structures non arborescentes même s'ils ne pouvaient pas les prendre en compte (Forbes-Riley *et al.*, 2006, p9), d'autre part, récemment, dans (Lee *et al.*, 2008), il en est fait clairement 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).

En conclusion, ces données empiriques font qu'un formalisme d'analyse automatique de discours doit pouvoir calculer des structures de dépendances non arborescentes. 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 suivantes qui régissent le positionnement des arguments d'un connecteur adverbial ou d'une conjonction postposée (par abus de langage, nous utilisons le terme de conjonction postposée (resp. préposée) pour désigner une conjonction qui introduit une subordonnée adverbiale postposée (resp. préposée)) :

Contrainte 1 *Le premier argument d'un connecteur adverbial ou d'une conjonction postposée se trouve à gauche de sa clause hôte².*

Contrainte 2 *Le second argument d'un connecteur adverbial ou d'une conjonction postposée est une portion contiguë de discours identique ou commençant à sa clause hôte (avec un éventuel dépassement de frontière de phrase).*

Ces contraintes ne font aucune différence entre le positionnement des arguments d'un connecteur adverbial ou d'une conjonction postposée. pourtant, il en existe une résumée dans la Contrainte 3.

Contrainte 3 *Le premier argument d'une conjonction postposée se trouve à gauche de sa clause hôte sans dépassement d'une frontière de phrase. Par contre, le premier argument d'un connecteur adverbial se trouve à gauche de sa clause hôte avec un dépassement d'une frontière de phrase.*

Qu'en est-il pour les autres connecteurs, à savoir les conjonctions préposées et les conjonctions de coordination ? Dans (Danlos et Hankach, 2008), nous avons étudié les discours comportant une subordonnée adverbiale préposée, ce qui nous amène à poser la contrainte suivante :

Contrainte 4 *Le premier argument d'une conjonction préposée (en suivant l'ordre linéaire du texte) est la subordonnée hôte du connecteur, le second est identique ou commence à la principale (avec un éventuel dépassement d'une frontière de phrase).*

Cette contrainte prend en compte le fait qu'une subordonnée préposée peut former un « cadre de discours » (Charolles, 2005), comme illustré en (3) de forme *Quand* C_1 , C_2 . *Ensuite* C_3 . Le second argument de *quand* dépasse une frontière de phrase : la visite de la Tour Eiffel et celle du Louvre sont toutes deux sous la portée de la subordonnée préposée, la structure de discours étant *Circonstance*(*Narration*(π_2 , π_3), π_1).

2. Ce premier argument doit aussi respecter la RFC quand on prend en compte le type coordonnant ou subordonnant des relations de discours lexicalisées par les connecteurs, comme nous l'avons vu dans la section précédente sur la SDRT. Ainsi, l'interprétation $R_a(\pi_1, \pi_2) \wedge R_b(\pi_1, \pi_3)$ n'est pas disponible lorsque R_a est coordonnante, puisque le premier argument de R_b ne peut pas être π_1 qui n'est pas ouvert pour l'attachement d'une information nouvelle.

(3) Quand il était à Paris, Fred a visité la Tour Eiffel. Ensuite, il a été au Louvre.

Nous n'avons pas fait d'étude linguistique poussée sur les conjonctions de coordination jouant le rôle de connecteur (et il n'en existe pas à notre connaissance). Notre grammaire STAG discursive ne traite donc pas encore ces connecteurs. Par contre, elle traite les connecteurs adverbiaux et les conjonctions de subordination (préposées ou postposées) en respectant les quatre contraintes linguistiques que nous avons avancées dans cette section.

2.3. Introduction à TAG et aux TAG synchrones

Les parties entre guillemets de cette section sont traduites de (Nesson et Shieber, 2006). Les figures 2, 4 et 5 sont des adaptations pour le français de figures présentées dans cet article. L'exemple (très simple) que nous utilisons dans cette section est :

(4) Jean apparemment aime Marie

2.3.1. 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 étiqueté 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 2 montre des illustrations des opérations de substitution et d'adjonction concernant des arbres élémentaires simples. »
Le préfixe α est utilisé pour les noms d'arbres initiaux, le préfixe β pour les noms d'arbres auxiliaires.

La sortie de l'analyse d'une phrase produit deux résultats : une « 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 3 présente l'arbre de dérivation pour la phrase (4). 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 3 nous dit que l'arbre auxiliaire modifieur β apparemment s'est adjoint au nœud dont l'adresse est 2 dans l'arbre initial α aime.

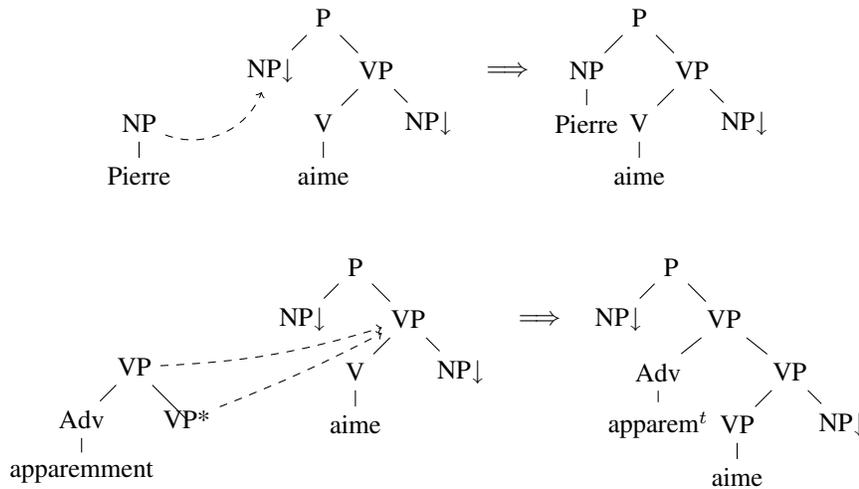


Figure 2. Illustrations des opérations de substitution et d'adjonction en TAG

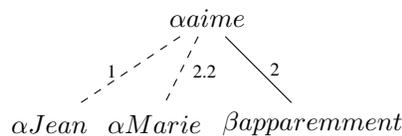


Figure 3. Arbre de dérivation pour la phrase (4)

2.3.2. 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 $\langle t_L, t_R, \sim \rangle$ où t_L et t_R sont des arbres élémentaires TAG et \sim 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 4 par les paires d'arbres élémentaires pour l'analyse de la phrase (4).

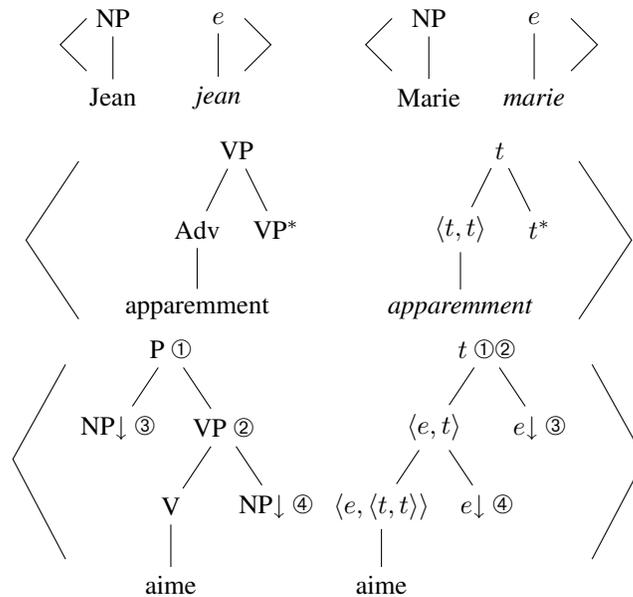


Figure 4. Paire d'arbres en STAG syntaxe–sémantique pour analyser (4)

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 5 donne l'unique arbre de dérivation pour les analyses syntaxique et sémantique de (4).

2.4. Interface phrase-discours

Ayant présenté les pré-requis théoriques, linguistiques et formels sur lesquels s'appuie notre formalisme, nous allons pouvoir entrer dans le vif du sujet. Il nous faut cependant présenter auparavant l'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. Une première illustration de ces disparités est fournie par les connecteurs adverbiaux : au niveau discursif un connecteur adverbial a obligatoirement deux arguments, tandis qu'au niveau phrastique, un connecteur

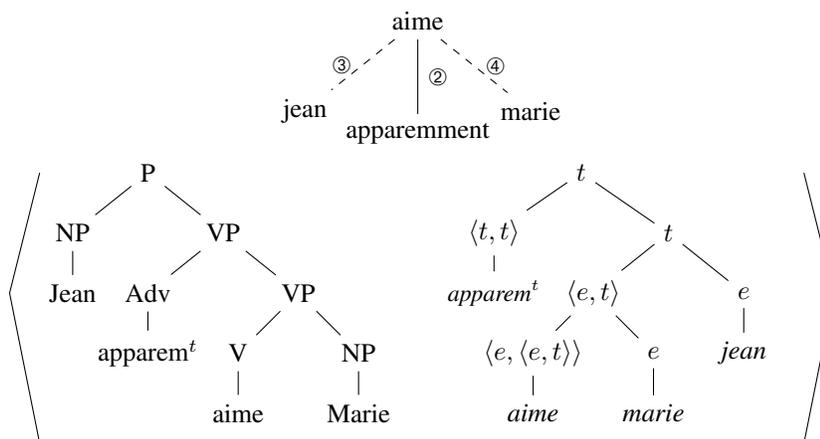


Figure 5. *Arbre de dérivation (unique) pour les arbres dérivés syntaxique et sémantique de (4)*

adverbial n'a qu'un 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). Une autre illustration est le fait qu'au niveau discursif un argument d'une conjonction de subordination peut dépasser une frontière de phrase (voir (2d) pour une conjonction postposée et (3) pour une conjonction préposée) alors qu'il ne serait en être question au niveau phrastique.

En conclusion, il est nécessaire de passer par une interface phrase-discours qui donne aux frontières de phrase le simple rôle d'un signe de ponctuation et qui permet de re-calculer les (deux) arguments d'un connecteur. Une telle interface est aussi utilisée en D-LTAG et nous nous en sommes inspirée. A 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, 2009). Dans le même ordre d'idées, une subordonnée adverbiale qui apparaît entre le NP sujet et le VP de sa clause hôte est placée en tête de sa phrase hôte avec un trait gardant la trace de sa position médiane. Enfin, rappelons que si une phrase ne commençant pas un discours ne contient aucun connecteur dans sa première clause principale, le connecteur

adverbial vide ϵ est introduit. A titre d'illustration, pour le discours (5), la FND est: C_1 . Ensuite^{vp} C_2 parce que C_3 . ϵ comme C_4 , C_5 .

- (5) 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. La partie discursive de D-STAG est illustrée avec les expressions régulières suivantes (en faisant abstraction de traits de position). Une FND sans conjonction préposée suit l'expressions régulière C (*Punct Conn C*)*, où la séquence *Punct Conn* est soit $.$ *Adv* soit $(,)$ *Conj C* où la virgule est facultative. Une FND comportant au moins une conjonction préposée comporte un élément C précédé de l'expression *Conj C*((,) *Conj C*)*. Enfin, les connecteurs *Adv* et *Conj* peuvent être facultativement suivis ou précédés d'un modifieur (Section 3.2).

Cette grammaire régulière demande à être étendue pour prendre en compte les conjonctions de coordination — dans leur emploi simple, voir C_1 parce que C_2 ou C_3 , et corrélé, voir C_1 soit parce que C_2 soit parce que C_3 . De plus, cette grammaire ne prend pas en compte la fait qu'une clause peut comporter des relatives, complétives ou incises. Or ces sous-clauses peuvent jouer 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.

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 τ_i . Pour immerger les analyses phrastiques dans les analyses discursives, nous utilisons la paire αP -to- D donnée à la Figure 6. Dans la suite de cet article, nous utilisons la paire τ'_i pour nommer αP -to- D dans lequel τ_i est substitué au lien \odot . Cette paire est aussi donnée à la Figure 6.

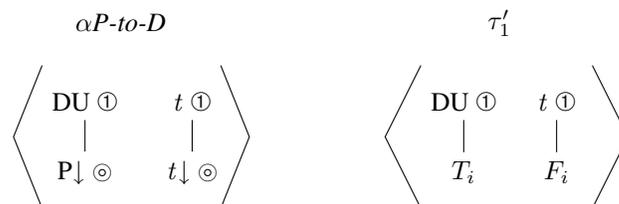


Figure 6. Paires d'arbres αP -to- D et τ'_1

Etant donné la grammaire régulière pour les FND que nous venons de présenter, la grammaire discursive comporte des arbres ancrés par les connecteurs adverbiaux et les conjonctions de subordination. Ceux-ci ont comme co-ancres un signe de ponctuation (Section 3.1). Elle comporte aussi des arbres ancrés par des modificateurs de connecteur (Section 3.2, la section précédente ne mettant en jeu aucun modificateur de connecteur).

Convention : les arbres de notre grammaire comportant au plus un nœud à substitution, celui-ci, quand il existe, porte systématiquement le lien \odot . Cette convention nous permet d'éviter de préciser systématiquement le lien auquel une opération de substitution a lieu.

3.1. Grammaire STAG pour les connecteurs

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 foncteur \mathcal{R}_i représentant le lambda-terme construit sur R_i . Par abus de langage, nous parlons de l'arbre sémantique ancré par R_i . Lorsqu'un connecteur est ambigu, c'est à dire qu'il permet d'exprimer plusieurs relations de discours, on crée 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 dans la Section 5.

Pour les arbres syntaxiques de connecteur, il faut prendre en compte la catégorie du connecteur, adverbial ou conjonction de subordination, et pour une conjonction de subordination, il faut prendre en compte sa position postposée ou préposée. Nous allons commencer (Section 3.1.1) par les adverbiaux et les conjonctions postposées, connecteurs qui ont un fonctionnement proche. Dans cette section, nous présenterons d'abord (Section 3.1.1.1) les paires d'arbres pour ces connecteurs, puis nous illustrerons (Section 3.1.1.2) le fonctionnement de ces paires dans l'analyse des FND ne comportant que ce type de connecteur, soit des FND de forme $C_1 Conn_1 C_2 Conn_2 \dots C_n$ en omettant les signes de ponctuation. Au terme de cette étude, nous prendrons en compte (Section 3.1.1.3) le type coordonnant ou subordonnant des relations de discours en présentant l'implémentation de la RFC posée en SDRT. Nous terminerons (Section 3.1.2) par la présentation des paires d'arbres pour les conjonctions préposées et par une brève description de l'analyse des FND comportant un tel connecteur.

3.1.1. Connecteurs adverbiaux, conjonction postposées, et analyses des FND comportant seulement ce type de connecteur

3.1.1.1. Paires d'arbres pour les adverbiaux et les conjonction postposées

Les arbres syntaxiques ancrés par un connecteur adverbial et une conjonction postposée sont donnés à la Figure 7, 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 co-ancres lexicales qui sont des signes de ponctuation de

catégorie *Punct*, le sous-arbre de racine *Punct* étant facultatif pour une conjonction préposée.

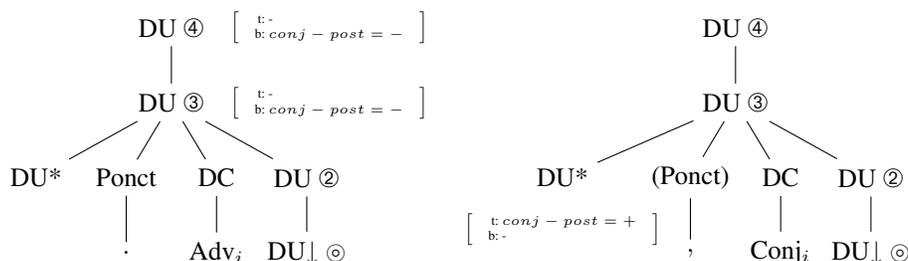


Figure 7. Arbres syntaxiques pour les adverbiaux et les conjonction postposées

Ces arbres suivent les principes suivants : ce sont des arbres **auxiliaires** à deux arguments représentés par un nœud pied DU^* et un nœud à substitution $DU\downarrow$. Le nœud pied DU^* correspond au premier argument du connecteur. Il est situé sur sa gauche, ce qui permet de respecter la Contrainte 1 posée à la Section 2.2. Le nœud à substitution $DU\downarrow$, qui porte le lien \odot , correspond au second argument du connecteur. Il est situé sur sa droite, ce qui permet de respecter la Contrainte 2 posée à la Section 2.2. En effet, il sert à substituer la clause hôte du connecteur, celle-ci ayant éventuellement subi des adjonctions (si c'est le cas, le second argument du connecteur est un DU commençant — à sa clause hôte).

Dans un FND de forme $C_1 Conn_1 C_2 Conn_2 \dots C_n$, l'attachement d'une information nouvelle $Conn_n C_{n+1}$ ne se fait que par **adjonction** de l'arbre ancré par $Conn_n$ — dans lequel on a substitué l'arbre syntaxique de τ'_{n+1} — sur un nœud de l'arbre syntaxique représentant $C_1 \dots C_n$. Le respect de l'ordre linéaire de la FND impose que l'attachement d'une information nouvelle se fasse par adjonction sur un nœud qui se situe sur **la frontière droite de l'arbre syntaxique**. C'est pourquoi les arbres ancrés par un adverbial ou une conjonction postposée comportent trois nœuds DU , avec les liens ②, ③ et ④, qui sont sur leur frontière droite. Ces nœuds DU portent des liens différents qui vont permettre de varier l'interprétation sémantique obtenue. 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, par exemple une adjonction à $DU\textcircled{3}$ pour attacher l'information nouvelle venant de $Conn_n C_{n+1}$ et une adjonction à $DU\textcircled{4}$ pour attacher l'information venant de $Conn_{n+1} C_{n+2}$. Soulignons que si on a fait une adjonction à $DU\textcircled{3}$ pour attacher $Conn_n C_{n+1}$, $DU\textcircled{2}$ n'est plus sur la frontière droite de l'arbre syntaxique. On ne peut donc plus faire d'adjonction à $DU\textcircled{2}$ pour attacher $Conn_{n+1} C_{n+2}$. Cette contrainte sera généralisée à la Contrainte 5 établie ultérieurement.

Passons aux arbres sémantiques. L'arbre sémantique de base ancré par une relation de discours R_i est l'arbre (A), ancré par le foncteur \mathcal{R}'_i , donné dans la partie gauche de la Figure 8. Cet arbre est un arbre auxiliaire avec un nœud pied de catégorie t et un

nœud à substitution aussi de catégorie t . Nous avons effectué une montée de type sur les nœuds pied et à substitution pour pouvoir obtenir des conjonctions de formules, voir Définition 2 ci-dessous. De ce fait, le foncteur \mathcal{R}'_i est de type $\langle ttt, \langle ttt, t \rangle \rangle$ où la notation ttt représente le type $\langle \langle t, t \rangle, t \rangle$. Ci-dessous sa définition³. Si le premier argument de \mathcal{R}'_i est $\lambda P.P(F_1)$ de type ttt , le second $\lambda Q.Q(F_2)$ de type ttt , alors le résultat est $R_i(F_1, F_2)$ de type t , F_1 et F_2 représentant des formes logiques clauseales de type t .

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

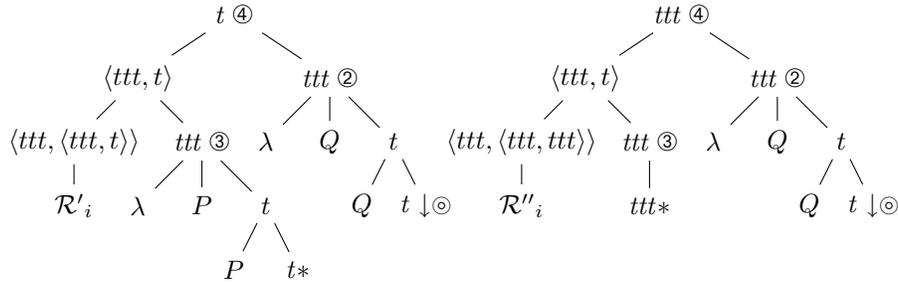


Figure 8. Arbres sémantiques (A) et (B) ancrés par les foncteurs \mathcal{R}'_i et \mathcal{R}''_i

Pour pouvoir faire des adjonctions aux liens ② et ③ de (A) qui sont portés par des nœuds de type ttt , nous définissons un autre arbre sémantique dont la racine et le nœud pied sont de catégorie ttt . Cet arbre, noté (B) et donné dans la partie droite la Figure 8, est ancéré par le foncteur \mathcal{R}''_i , de type $\langle ttt, \langle ttt, ttt \rangle \rangle$, qui est défini ci-dessous. Il contient le symbole \wedge qui va permettre d'obtenir une conjonction de formules. Si le premier argument de \mathcal{R}''_i est $\lambda P.P(F_1)$, le second $\lambda Q.Q(F_2)$, alors le résultat est $\lambda P.(R_i(F_1, F_2) \wedge P(F_1))$ de type ttt , ce qui peut se gloser par l'ensemble des propriétés de F_1 sachant que l'on a par ailleurs $R_i(F_1, F_2)$.

Définition 2 $\mathcal{R}''_i = \lambda XY P.X(\lambda r.Y(\lambda s.R_i(r, s) \wedge P(r)))$
avec $X, Y : ttt = \langle \langle t, t \rangle, t \rangle$; $P : \langle t, t \rangle$ et $r, s : t$

3.1.1.2. Analyse des FND de forme $C_1 Conn_1 C_2 Conn_2 \dots C_n$

Ayant défini nos arbres syntaxiques et sémantiques, tournons-nous vers l'analyse des FND ne comportant que des adverbiaux et des conjonctions postposées en commençant par le cas simple de ceux à deux clauses. Les analyses d'une

3. Si nous n'avions pas effectué de montées de type, le foncteur associé à R_i serait tout simplement $\mathcal{R}_i = \lambda r s.R_i(r, s)$ avec $r, s : t$, \mathcal{R}_i étant de type $\langle t, \langle t, t \rangle \rangle$.

FND à deux clauses de forme C_1 *parce que* C_2 avec avec une conjonction postposée sont donnés dans la Figure 9. Dans l’arbre de dérivation, le nœud étiqueté *parce – que_{post} ÷ Explication* s’adjoint au lien ① de τ'_1 , qui est un lien porté par un nœud t dans l’arbre sémantique. Pour la satisfaction des types, l’arbre sémantique de la paire *parce – que_{post} ÷ Explication* doit correspondre à l’arbre (A) de nœud pied t^* , pour lequel le foncteur est \mathcal{R}'_1 avec $R_1 = \text{Explication}$. L’arbre dérivé sémantique débouche sur la formule $\text{Explication}(F_1, F_2)$. Les analyses d’une FND à deux clauses de forme C_1 . *ensuite* C_2 avec un adverbial sont en tout point similaires à celles obtenues pour une FND de forme C_1 *parce que* C_2 . Il suffit d’utiliser *ensuite ÷ Narration* à la place de *parce – que_{post} ÷ Explication*.

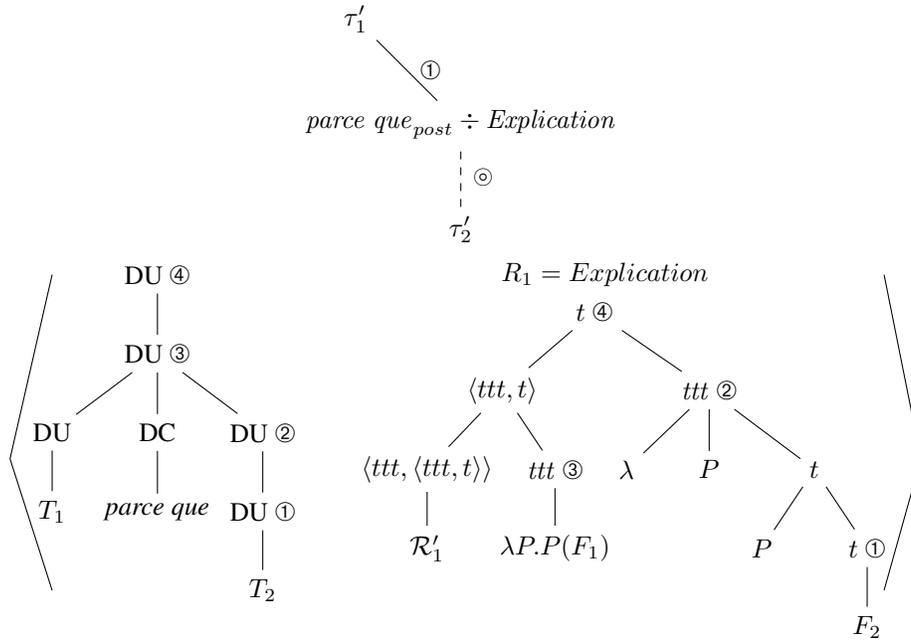


Figure 9. Arbre de dérivation et arbres dérivés syntaxique et sémantique pour une FND de forme C_1 *parce que* C_2

Passons au cas plus complexe des FND à trois clauses dont la forme simplifiée (i.e. sans signe de ponctuation) est notée ici C_1 *Conn_a* C_2 *Conn_b* C_3 . Rappelons que l’on doit être en mesure de produire quatre types d’interprétation qui ont été illustrés dans les exemples (2) de la Section 2.2, de forme normalisée C_1 *parce que* C_2 . *Adv_b* C_3 . Nous allons examiner les analyses discursives de ces quatre exemples. Nous notons β_a la paire d’arbres *parce – que_{post} ÷ Explication* et β_b la paire *Adv_b ÷ R_b*.

Après avoir analysé C_1 *parce que* C_2 , les analyses sont celles de la Figure 9. L’arbre syntaxique contient quatre nœuds *DU* sur sa frontière droite qui peuvent donc

recevoir l'adjonction de l'arbre syntaxique de β_b tout en respectant l'ordre linéaire de la FND. En parcourant cette frontière droite de façon ascendante, ces nœuds portent le lien ① venant de l'arbre syntaxique de τ'_2 et les liens ②, ③ et ④ venant de l'arbre ancré par *parce que*. Les analyses des quatre exemples de (2) s'obtiennent en adjoignant β_b à un de ces quatre liens.

Commençons par (2d), répété en (6), avec $\beta_b = \text{aussi} \div \text{Parallèle}$, pour lequel on veut calculer la formule sémantique $\text{Explication}(F_1, \text{Parallèle}(F_2, F_3))$.

- (6) Fred est de mauvaise humeur parce qu'il a mal dormi. Il a aussi raté son permis de conduire. [de FND C_1 parce que C_2 . aussi C_3 .]

Cet exemple s'analyse en adjoignant β_b au lien ① de τ'_2 , lien porté par un nœud t dans l'arbre sémantique de τ'_2 . Pour la satisfaction des types, on doit utiliser pour R_b l'arbre (A) ancré par \mathcal{R}'_b . L'arbre dérivé sémantique pour (6) est donné à la Figure 10. Le sous-arbre de racine t à l'adresse de Gorn 2 débouche sur le lambda-terme $\lambda P.P(\text{Parallèle}(F_1, F_2))$ avec $P : \langle t, t \rangle$. $\text{Parallèle}(F_1, F_2)$ est donc le second de $R_a = \text{Explication}$, qui a pour premier argument F_1 , d'où la formule $\text{Explication}(F_1, \text{Parallèle}(F_2, F_3))$.

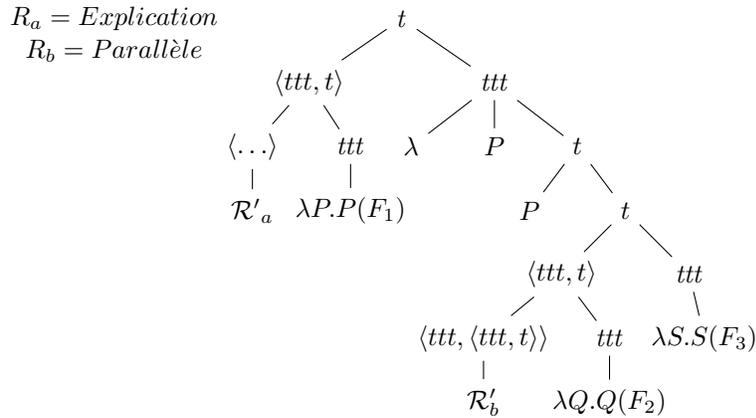


Figure 10. Arbre dérivé sémantique pour (6) avec l'interprétation $R_a(F_1, R_b(F_2, F_3))$

Continuons par (2c), répété en (7), avec $\beta_b = \epsilon \div \text{Explication}$, pour lequel on veut calculer la formule sémantique $\text{Explication}(F_1, F_2) \wedge \text{Explication}(F_2, F_3)$ avec une conjonction de formules.

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

Cet exemple s'analyse en adjoignant β_b au lien ② de β_a , lien porté par un nœud de type t dans l'arbre sémantique de β_a . Pour le respect du typage, on doit utiliser

pour R_b l'arbre (B) de nœud pied t ancré par \mathcal{R}_b'' . L'arbre dérivé sémantique pour (7) est donné dans la partie gauche de la Figure 11. Le sous-arbre de racine t à l'adresse de Gorn 2 débouche sur le lambda-terme $\lambda P.(Explication(F_2, F_3) \wedge P(F_2))$ avec $P : \langle t, t \rangle$, que l'on peut gloser par "l'ensemble des propriétés de F_2 sachant que F_2 est expliquée par F_3 ". Seule F_2 étant sous la porté de P , c'est le second argument de $R_a = Explication$, qui a pour premier argument F_1 , d'où la formule $Explication(F_1, F_2) \wedge Explication(F_2, F_3)$.

Passons à (2a), répété en (8), avec $\beta_b = ensuite \div Narration$, pour lequel on veut calculer la formule sémantique $Explication(F_1, F_2) \wedge Narration(F_1, F_3)$ avec aussi une conjonction de formules.

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

Cet exemple s'analyse en adjoignant β_b au lien ③ de β_a , lien porté par un nœud t dans l'arbre sémantique de β_a . Par respect du typage, on doit utiliser pour R_b l'arbre (B) de nœud pied t ancré par \mathcal{R}_b'' . L'arbre dérivé sémantique pour (8) est donné dans la partie droite de la Figure 11. Le sous-arbre de racine t à l'adresse de Gorn 1.2 débouche sur le lambda-terme $\lambda P.(Narration(F_1, F_3) \wedge P(F_1))$ avec $P : \langle t, t \rangle$, que l'on peut gloser par "l'ensemble des propriétés de F_1 sachant que F_1 a pour suite narrative F_3 ". Seule F_1 étant sous la porté de P , c'est le premier argument de $R_a = Explication$, qui a pour second argument F_2 , d'où la formule $Explication(F_1, F_2) \wedge Narration(F_1, F_3)$.

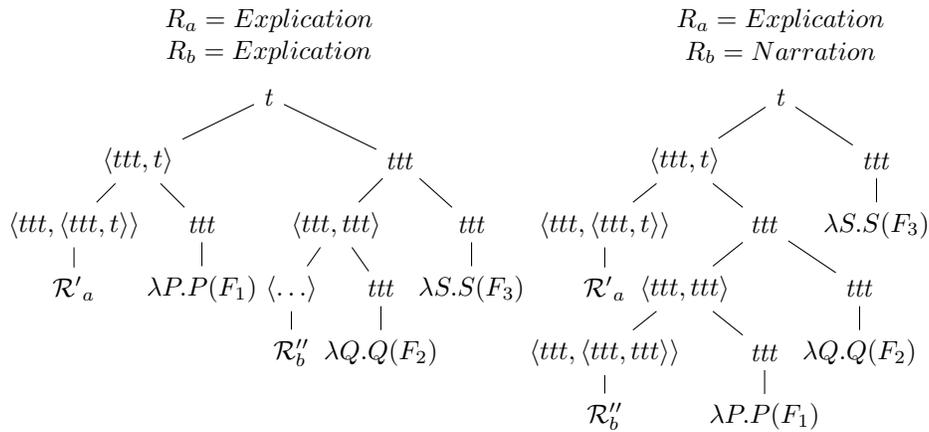


Figure 11. Arbres dérivés sémantiques pour (7) et (8) avec les interprétations $R_a(F_1, F_2) \wedge R_b(F_2, F_3)$ et $R_a(F_1, F_2) \wedge R_b(F_1, F_3)$

Terminons par (2b), répété en (9), avec $\beta_b = \epsilon \div Commentaire$, pour lequel on veut calculer la formule sémantique $Commentaire(Explication(F_1, F_2), F_3)$.

- (9) 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_b au lien ④ de β_a , lien porté par un nœud t dans l'arbre sémantique de β_a . Pour la satisfaction des types, on doit utiliser pour R_b l'arbre (A) de nœud pied t^* ancré par \mathcal{R}'_b . L'arbre dérivé sémantique pour (9) est donné à la Figure 12. Le sous-arbre de racine ttt à l'adresse de Gorn 1.2 débouche sur le lambda-terme $\lambda P.P(\text{Explication}(F_1, F_2))$ avec $P : \langle t, t \rangle$. $\text{Explication}(F_1, F_2)$ est donc le premier argument de $R_b = \text{Commentaire}$, qui a comme second argument F_3 , d'où la formule $\text{Commentaire}(\text{Explication}(F_1, F_2), F_3)$.

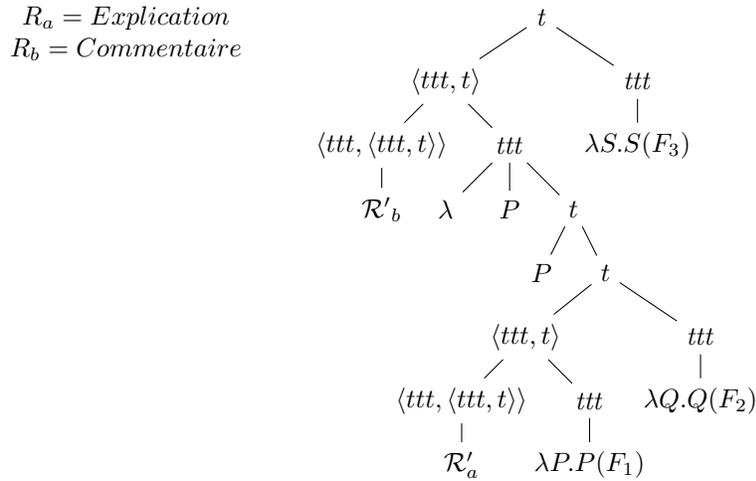


Figure 12. Arbre dérivé sémantique pour (9) avec l'interprétation $R_b(R_a(F_1, F_2), F_3)$

En conclusion, les quatre analyses des FND de forme $C_1 \text{ Conj}_a C_2. \text{Adv}_b C_3$ s'obtiennent grâce aux quatre sites d'adjonction qui sont sur la frontière droite de l'arbre syntaxique de $C_1 \text{ Conj}_a C_2$ et grâce aux arbres sémantiques (A) et (B) respectivement de nœud pied t et ttt et de foncteur \mathcal{R}'_i et \mathcal{R}''_i .

Pour les FND à trois clauses de forme simplifiée $C_1 \text{ Conn}_a C_2 \text{ Conn}_b C_3$, nous venons d'examiner en détail les discours de forme $C_1 \text{ Conj}_a C_2. \text{Adv}_b C_3$ où Conn_a est une conjonction postposée et Conn_b un connecteur adverbial. Il reste trois cas :

- Conn_a est une conjonction post-posée et Conn_b aussi,
- Conn_a est un adverbial et Conn_b aussi,
- Conn_a est un adverbial et Conn_b une conjonction post-posée.

Les deux premiers cas ne posent aucun problème nouveau. Ils s'obtiennent par les mécanismes développés en détail pour $C_1 \text{ Conj}_a C_2. \text{Adv}_b C_3$. Le troisième cas,

c'est-à-dire les FND de forme $C_1. Adv_a C_2 Conj_b C_3$, soulève la question de l'implémentation de la contrainte stipulant que l'argument gauche de la conjonction post-posée $Conj_b$ ne peut pas dépasser la frontière de phrase (voir la Contrainte 3 de la Section 2.2 qui, indiquons-le, 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 7. Plus précisément :

- le nœud pied d'un arbre ancré par une conjonction postposée est décoré par le trait top $conj - post = +$ ce qui se glose par “cet arbre auxiliaire sert à adjoindre une conjonction post-posée” ;

- 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 post-posée”.

Ces traits bloquent l'adjonction de $Conj_b \div R_b$ aux liens ③ et ④ de $Adv_a \div R_a$ grâce à l'échec d'unification $(conj - post = +) \cup (conj - post = -)$. Par conséquent, $Conj_b \div R_b$ ne peut s'adjoindre qu'au lien ② de $Adv_a \div R_a$ et au lien ① de τ'_2 , ce qui débouche respectivement sur les interprétations $R_a(F_1, F_2) \wedge R_b(F_2, F_3)$ et $R_a(F_1, R_b(F_2, F_3))$, où le premier argument de R_b est F_2 . Ces interprétations respectent la Contrainte 3.

Qu'en est-il de l'analyse des FND sans conjonction préposée à n clauses avec $n > 3$? Aucun mécanisme nouveau n'est mis en jeu. Attacher l'information nouvelle venant de $Conn_n C_{n+1}$ revient, dans l'arbre de dérivation représentant $C_1 \dots C_n$, à adjoindre $Conn_n \div R_n$ — dans lequel on a substitué τ'_{n+1} — au lien ① de τ'_n ou au lien \textcircled{i} avec $i \in \{2, 3, 4\}$ d'un nœud $\beta_k = Conn_k \div R_k$, le nœud de l'arbre syntaxique de β_k au lien \textcircled{i} devant être sur la frontière droite de l'arbre syntaxique de $C_1 \dots C_n$ (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 ordonné⁴, on doit avoir recours à une convention graphique qui représente un arbre de dérivation comme ordonné. La convention suivante est suffisante : les nœuds $\tau'k$ 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 $\tau'k$, les nœuds β_k de l'arbre de dérivation qui permettent d'attacher l'information nouvelle venant de $Conn_n C_{n+1}$ sont ceux qui sont situés sur la frontière droite de l'arbre de dérivation en respectant la Contrainte 5 régissant deux adjonctions aux liens \textcircled{n} et \textcircled{m} d'un même nœud⁵ :

4. 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.

5. Cette contrainte est valide car nous avons pris le soin d'attribuer les liens ②, ③ et ④ de façon à ce que les nœuds qui les portent suivent un parcours ascendant de la frontière droite des arbres dérivés syntaxiques, voir celui de la Figure 9.

Contrainte 5 Si β_j — dans lequel on a substitué τ'_{j+1} — est adjoint au lien \textcircled{n} d'un nœud β_i , alors on ne peut adjoindre β_k — dans lequel on a substitué τ'_{k+1} — au lien \textcircled{m} du même nœud β_i qu'en respectant la règle suivante : $\tau'_{j+1} \prec \tau'_{k+1} \Rightarrow n < m$ (avec n et m appartenant à $\{2, 3, 4\}$).

Cette contrainte généralise celle que nous avons formulée dans la Section 3.1.1.1 : si on a fait une adjonction au nœud $DU^{\textcircled{3}}$ d'un arbre syntaxique ancré par $Conn_i$, alors on peut faire une nouvelle adjonction dans cet arbre au nœud $DU^{\textcircled{4}}$ mais pas au nœud $DU^{\textcircled{2}}$.

3.1.1.3. Implémentation de la RFC pour les FND sans conjonction préposée

La RFC posée en SDRT (Section 2.1) induit une contrainte d'ordre syntaxique et une contrainte d'ordre sémantique. La contrainte d'ordre syntaxique revient à respecter l'ordre linéaire du discours, point que nous venons d'examiner. Regardons comment modéliser la contrainte d'ordre sémantique qui interdit d'attacher une information nouvelle au premier élément d'une relation coordonnante. Cette contrainte demande de pouvoir distinguer les arbres sémantiques ancrés par une relation coordonnante versus subordonnante. Ceci est obtenu en créant deux copies des schémas d'arbre (A) et (B), copies qui se distinguent par un trait *top* [*coord* = ±] sur leur nœud pied. La contrainte sémantique se traduit alors en D-STAG par le fait que toute adjonction au lien $\textcircled{3}$ de (A) ou (B) dont le nœud pied est décoré du trait [*coord* = +] est interdite. En effet, c'est le lien $\textcircled{3}$ qui sert à obtenir des interprétations de type $R_a(F_1, F_2) \wedge R_b(F_1, F_3)$ qui doivent être bloquées lorsque R_a est coordonnante (voir note 2). En résumé, l'implémentation de la partie sémantique de la RFC revient à décorer le nœud *ttt* $\textcircled{3}$ d'une contrainte de non adjonction dans les arbres (A) et (B) dont le nœud pied est décoré du trait [*coord* = +].

En fait, la RFC induit des contraintes sémantique plus subtiles que celle nous venons de traiter. Par exemple, la contrainte stipulant que dans une FND de forme $C_1 Conn_a C_2 Conn_b C_3 \dots$ pour lequel on a calculé la sémantique est $R_a(F_1, F_2) \wedge R_b(F_2, F_3)$ obtenue par adjonction de β_b au lien $\textcircled{2}$ de β_a , il est interdit de faire une nouvelle adjonction aux liens $\textcircled{3}$ et $\textcircled{4}$ de β_a si R_b est coordonnante. Cette contrainte se traduit en D-STAG par un jeu de traits dans les arbres (A) et (B), que nous ne détaillerons pas ici, qui bloque l'adjonction aux liens $\textcircled{3}$ et $\textcircled{4}$ de R_a si le lien $\textcircled{2}$ a déjà reçu l'adjonction d'une relation coordonnante R_b . Disons simplement que le mécanisme des traits en TAG est suffisamment puissant pour prendre en compte toute contrainte relative à la RFC de la SDRT, si subtile soit-elle. Signalons aussi que la contrainte de «Continuous Discourse Pattern» se traduit par un simple ajout de traits dans les arbres (A) et (B) ancrés par une relation subordonnante, sans avoir recours à un processus non monotone comme c'est le cas en SDRT.

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 arbres sémantiques (A) et (B) pour prendre en compte la RFC peuvent être transférés dans les arbres syntaxiques ancrés par les connecteurs. Ceci permet de prendre en compte la RFC au niveau syntaxique, ce qui peut s'avérer fort utile, voir Section 5.

3.1.2. Brève analyse des FND comportant une conjonction préposée

L'arbre syntaxique ancré par une conjonction préposée est donné à la Figure 13. Il permet de respecter la Contrainte 4 établie à la Section 2.2. 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 l'arbre sémantique (A) un nœud $t^{\textcircled{5}}$ 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.

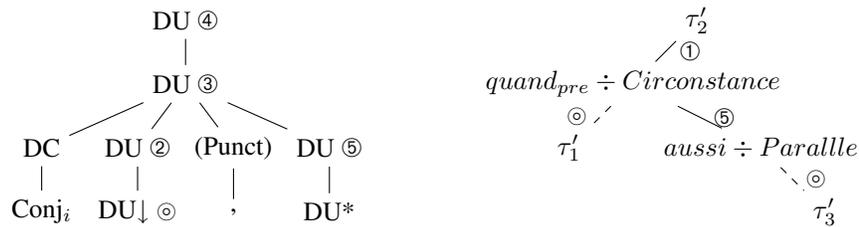


Figure 13. Arbre ancré par une conjonction préposée et arbre de dérivation pour (10)

L'adjonction au lien ⑤ d'un arbre de conjonction préposée sert pour un « cadre de discours » comme illustré en (3) répété ci-dessous en (10), dont la sémantique se calcule à partir de $Circonstance(Narration(F_2, F_3), F_1)$, la conjonction préposée *quand* ayant un argument qui dépasse une frontière de phrase.

(10) Quand il était à Paris, Fred a visité la Tour Eiffel. Ensuite, il a été au Louvre.

L'arbre de dérivation pour (10) est présenté à la Figure 13. Pour la satisfaction des types, les arbres ancrés par *Circonstance* et *Parallèle* sont tous deux le schéma (A). Le foncteur \mathcal{R}'_i avec $R_i = Circonstance$ ayant comme arguments $\lambda P.P(Narration(F_2, F_3))$ et $\lambda Q.Q(F_1), Narration(F_2, F_3)$ est le premier argument de *Circonstance*, F_1 le second, d'où la formule $Circonstance(Narration(F_2, F_3), F_1)$.

Une remarque. Les exemples comme (10) 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)⁶.

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 7 et 13) : il n’existe aucun arbre auxiliaire enveloppant, ce qui respecte la première contrainte. Par contre, pour traiter un exemple comme (10), 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.

3.2. Modification de connecteur

Dans les exemples (11), nous considérons les adverbiaux *par exemple* et *juste* comme des modificateurs du connecteur *parce que*.

- (11)a. 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), de FND [C_1 *parce que par exemple* C_2])
- b. Tu ne dois pas faire confiance à Jean juste parce qu’il ne rend jamais ce qu’il a emprunté.
(de FND [C_1 *juste parce que* C_2])

Ces adverbiaux ancrent des arbres auxiliaires de nœud pied *DC* (arbre gauche pour *juste*, droit pour *par exemple*). Pour adjoindre ces arbres, il est nécessaire d’ajouter un lien © sur le nœud *DC* dans les arbres syntaxiques de connecteurs (Figures 7 et 13). Du côté sémantique, il est nécessaire d’ajouter un nœud portant le lien © dominant les foncteurs \mathcal{R}'_i et \mathcal{R}''_i dans les arbres (A) et (B) et de définir deux versions des arbres sémantiques de modifieur pour le respect du typage. La contribution sémantique des modificateurs ne sera illustrée que pour *par exemple* dans un exemple comme (11a) mettant en jeu l’arbre (A).

Comme expliqué dans (Webber *et al.*, 2003), l’interprétation de (11a) est que le non retour par Jean des objets empruntés est un exemple des raisons pour ne pas lui faire confiance. Nous donnons à (11a) la représentation sémantique suivante qui est inspirée de (Webber *et al.*, 2003) et adaptée à D-STAG :

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

$Exemplification(F_2, \lambda r.Explification(F_1, r))$ avec $r : t$. Pour obtenir cette interprétation, nous définissons le foncteur $\mathcal{P}ar-ex$ de la façon suivante⁷ :

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

La paire nommée β_{par-ex} est donnée à la Figure 14 qui montre aussi l'arbre de dérivation de (11a). L'arbre sémantique dérivé donne l'interprétation voulue.

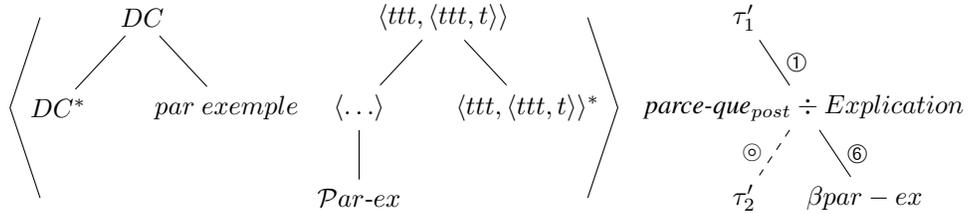


Figure 14. Paire β_{par-ex} et arbre de dérivation pour (11)

3.3. 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, éventuellement modifiés. Cette grammaire est de petite taille. Ainsi, elle ne comporte que 9 (schémas d') arbres syntaxiques⁸. Nous avons considéré que les connecteurs d'une catégorie donnée se comportaient tous de la même façon, mais STAG étant une grammaire lexicalisée, 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 particularités d'une relation de discours donnée.

Rappelons (Section 2.4) que cette grammaire demande à être étendue. Cette extension ne devrait pas soulever de problème insurmontable vu la puissance d'expressivité offerte par STAG.

7. Je remercie Laurent Roussarie de m'avoir aidé à calculer ce lambda-terme. Si le foncteur de relation de discours était simplement $\mathcal{R}_i = \lambda r s.R_i(r, s)$ avec $r, s : t$, sans montée de type (voir note 3), $\mathcal{P}ar-ex$ serait $\lambda R p q.Exemplification(q, \lambda r.R(p, r))$ avec $R : \langle t, \langle t, t \rangle \rangle$ et $p, q : t$.

8. Un pour αP -to-D, trois schémas d'arbres pour les adverbiaux, conjonctions préposées et postposées pour lesquels on fait deux copies, et deux schémas d'arbres pour les modifieurs.

4. Comparaison entre D-STAG et D-LTAG

D-STAG et D-LTAG (Forbes-Riley *et al.*, 2006) ont le même objectif et la même architecture (Section 1). Les divergences entre ces deux formalismes se situent dans la partie discursive. D’abord, D-LTAG utilise peu les relations de discours et ignore la distinction entre relations coordonnantes et subordonnantes. Les formes logiques des discours sont calculées par le même procédé que celui utilisé pour calculer les formes logiques des phrases. Ceci empêche D-LTAG de bénéficier des résultats apportés par les théories sur le discours qui apportent des connaissances rhétoriques et pragmatiques. 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 les connecteurs de type adverbial ancrent des arbres avec un seul argument (qui est fourni *structurellement*, l’autre étant fourni *anaphoriquement* (Webber *et al.*, 2003 ; Webber, 2004)). De plus, les arbres de connecteur avec deux arguments sont pour la plupart des arbres **initiaux** en D-LTAG. C’est en particulier le cas pour les arbres ancres par une conjonction de subordination qui sont considérés comme des connecteurs « structuraux », en passant sous silence le fait qu’un de leurs arguments peut dépasser une frontière de phrase. En revanche, les arbres de connecteur sont tous des arbres **auxiliaires** en D-STAG. Ces différences sur les arbres de connecteur induisent des différences majeures sur l’analyse syntaxique et surtout sur la structure de discours. A titre d’illustration, l’arbre syntaxique et l’arbre de dérivation pour (1) = (2a), répété en (12), produits par une adaptation française de D-LTAG sont donnés à la Figure 15.

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

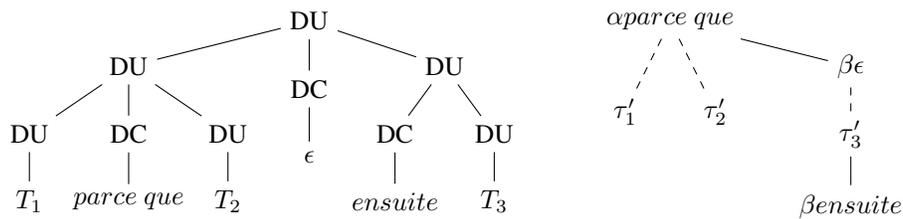


Figure 15. Arbre syntaxique et arbre de dérivation pour (12) en D-LTAG

L’arbre syntaxique comporte trois nœuds étiquetés *DC*, un pour *parce que*, un pour ϵ , et un pour *ensuite*. Elle est différente de l’analyse syntaxique de (12) produite par 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_1, F_2), F_3)*, ce qui

n'est pas correct : l'explication fournie pour la visite de Fred au supermarché (*son frigo était vide*) ne doit pas être sous la portée de *Narration*. La structure de ce discours est $Explication(F_1, F_2) \wedge Narration(F_1, F_3)$. Cette structure, qui correspond à un graphe de dépendances non arborescent (Section 2.2) ne peut pas être obtenue en D-LTAG. De même, la structure $R_a(\pi_1, \pi_2) \wedge R_b(\pi_2, \pi_3)$, qui correspond aussi à un graphe de dépendances non arborescent, ne peut pas être obtenu en D-LTAG.

A ce propos, signalons que (Lee *et al.*, 2008) présentent des statistiques fort étonnantes pour les quatre interprétations des discours de forme $C_1 Conj_a C_2 Adv_b C_3$, où $Conj_a$ est une des douze conjonctions de subordination les plus usitées en anglais. Ils indiquent, à partir d'observations faites sur le PDTB (PDTB Group, 2008), que la première interprétation $R_a(\pi_1, \pi_2) \wedge R_b(\pi_1, \pi_3)$ s'observe dans 12,4% des cas, la seconde $R_b(R_a(\pi_1, \pi_2), \pi_3)$ dans 83,4% des cas, la troisième $R_a(\pi_1, \pi_2) \wedge R_b(\pi_1, \pi_3)$ et la quatrième $R_a(\pi_1, R_b(\pi_2, \pi_3))$ dans 4,2% des cas⁹. L'écrasante majorité de la seconde interprétation nous étonne énormément. En effet, une conjonction de subordination exprimant en général une relation subordonnante¹⁰, cette seconde interprétation, illustrée par (2b), revient à dire qu'une relation subordonnante R_a forme un constituant complexe qui est le premier argument de R_b . Comme expliqué dans (Danlos, 2006), la formation d'un constituant complexe autour d'une relation subordonnante est exclue en RST et elle n'est prise en compte que de façon marginale en SDRT. Il semble peu probable que RST et SDRT, les deux théories de discours les plus élaborées, soient passées à côté d'une interprétation qui représenterait 83,4% des cas (ou un peu moins si on soustrait les cas où la conjonction $Conj_a$ exprime une relation coordonnante). Nous n'avons pas encore été vérifié dans le PDTB, mais nous spéculons que des exemples comme (12) relevant de la première interprétation ont été massivement annotés comme relevant de la seconde interprétation, ce qui n'est peut-être pas sans rapport avec la structure arborescente de la seconde qui contraste avec la structure non arborescente de la première, et avec le fait que le PDTB a été annoté principalement par l'équipe qui a développé D-LTAG, qui donne à (12) la seconde interprétation.

Une autre différence entre D-STAG et D-LTAG s'observe avec le traitement de la modification des connecteurs de discours (Section 3.2). D-LTAG ignore la modification des connecteurs et traite *par exemple* dans (11a) comme un connecteur, ce qui conduit à de calculs laborieux (Forbes-Riley *et al.*, 2006, pp 31-35) pour obtenir la représentation sémantique de (11a). 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.

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

10. (Matthiessen et Thompson, 1988) ont posé qu'il y avait 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)*.

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 (basés sur TAG ou d'autres formalismes) mais aucun analyseur sémantique (obtenu par une S-TAG ou autrement). De plus, les analyseurs syntaxiques ne donnent pas de bons résultats sur les frontières de clauses, alors que c'est un point crucial pour la seconde étape¹¹. Nous nous attaquons quand même à l'implémentation d'un analyseur D-STAG du français sans attendre de meilleurs résultats au niveau phrastique. Ce travail est effectué en collaboration avec Pascal Denis, Grégoire Détrez, Charlotte Roze et Benoît Sagot, membres ou étudiants d'ALPAGE.

Pour cela, nous travaillons à la réalisation d'un système d'identification de clauses, appelé DNFEUR, qui s'inspire du système SIGLÉ (Système d'Identification de propositions avec une Grammaire Légère (Nakamura-Delloye, 2006)) reposant sur une analyse syntaxique superficielle. Signalons que la réalisation de DNFEUR est une fin en soi. En effet, les FND ouvrent une avenue de recherche pour les linguistes travaillant sur les connecteurs de discours et leurs arguments, 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. Nous voulons aussi étudier les cas où deux connecteurs se partagent la même phrase hôte. Ces cas ne sont pas encore traités en D-STAG alors qu'ils le sont en D-LTAG (Webber *et al.*, 2001). Avant d'élaborer une stratégie pour les gérer adéquatement en D-STAG, nous voulons les étudier en corpus afin de pouvoir répondre à une question fondamentale : est-ce que deux connecteurs se partageant la même phrase hôte expriment deux relations de discours ?

11. Nous avons constaté cette triste réalité sur FRMG, métagrammaire TAG (Villemonde de La Clergerie, 2005), les chercheurs toulousains travaillant sur le discours 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 phrase 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.

Par ailleurs, nous réalisons une base lexicale de connecteurs du français, appelée LEXCONN. 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 examinons si c’est effectivement un connecteur, et si c’est le cas, nous déterminons quelle(s) relation(s) de discours exprime le connecteur et la nature coordonnante et subordonnante des relations. Ce travail est effectué en étroite collaboration avec les chercheurs toulousains qui ont aussi réalisé une base lexicale de connecteurs dans le cadre d’ANNODIS¹². Leur base compte 64 entrées, la nôtre 370. Ces bases sont destinées à être fusionnées.

Une fois DNFEUR et LEXCONN terminés, nous pourrions nous attaquer à la troisième étape, la partie discursive de D-STAG. Sa grammaire 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 2.4). Pour l’analyseur, nous devons 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 et subordonnante des relations de discours des arbres sémantiques vers les arbres syntaxiques, et donc prendre en compte la RFC au niveau syntaxique ce qui limite le nombre d’analyses possibles (voir la remarque à la fin de la Section 3.1.1.3).

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 demande 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 reposant sur le formalisme D-STAG présenté dans cet article. Ce formalisme repose sur une théorie du discours SDRT, sur des bases linguistiques sérieuses, et sur un formalisme STAG qui a une grande puissance d’expressivité et dont la complexité informatique est maîtrisée.

12. 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>.

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, 2009.
- Charolles M., « Framing adverbials and their role in discourse cohesion, from connection to forward labelling », *Proceedings of the First Symposium on the Exploration and Modelling of Meaning, 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 SIG-DIAL'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.
- Harris Z., *Mathematical Structures of Language*, Krieger Pub co, New York, 1986.
- Joshi A., « Tree-adjointing 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 n° IRCS-08-01, Institute for Research in Cognitive Science, University of Philadelphia, 2008.
- Prasad R., Dinesh N., Lee A., Joshi A., Webber B., « Attribution and its annotation in the Penn Discourse TreeBank », *Revue TAL*, 2006.
- Schabes Y., Waters R., « Tree insertion Grammar: A Cubic-Time, Parsable Formalism that Lexicalizes Context-Free Grammar without Changing the Trees Produced », *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.
- Stede M., « RST Revisited: Disentangling Nuclearity" », in C. Fabricius-Hansen, W. Ramm (eds), *Subordination' vs. 'coordination in Sentences and Texts*, John Bejamins, p. 33-58, 2007.
- 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., « DTAG: extending lexicalized TAG to discourse », *Cognitive Science*, vol. 28, n° 5, p. 751-779, 2004.
- Webber B., Knott A., Joshi A., « Multiple Discourse Connectives in a Lexicalized Grammar for Discourse », in H. Bunt, R. Muskens, E. Thijsse (eds), *Computing Meaning, volume 2*, Kluwer Academic Press, p. 229-246, 2001.
- Webber B. L., Joshi A., Stone M., Knott A., « Anaphora and Discourse Structure », *Computational Linguistics*, vol. 29, n° 4, p. 545-587, 2003.