Actes de CORIA-TALN-RJCRI-RECITAL 2025
Marseille, France, 30 juin au 4 juillet 2025

Syntaxe en dépendance avec les grammaires catégorielles abstraites : une application à la théorie sens-texte

Marie Cousin
Université de Lorraine, CNRS, Inria, LORIA, F-54000 Nancy, France
marie.cousin@loria.fr

RÉSUMÉ

L'implémentation de Cousin (2025) de la théorie sens-texte dans les grammaires catégorielles abstraites, un formalisme grammatical basé sur le λ-calcul, présente différentes limitations, en particulier l'articulation des dépendances au sein des structures, et le comportement des adjectifs et adverbes (rôle prédicatif des adjectifs et adverbes au niveau sémantique, nombre de modifieurs, etc.). Tout en utilisant la composition de grammaires catégorielles abstraites de Cousin (2025), nous proposons une représentation des structures syntaxiques en dépendances inspirée de de Groote (2023b) qui lève ces limitations.

ABSTRACT

Dependency Syntax with Abstract Categorial Grammars : an Application to Meaning-Text Theory

Cousin (2025)'s encoding of the meaning-text theory into abstract categorial grammars, a grammatical formalism based on λ-calculus, shows some limitations, in particular the articulation of dependencies within structures and the behaviour of adjectives and adverbs (predicative role of adjectives and adverbs at semantic level, number of modifiers, etc.). While using the composition of abstract categorial grammars from Cousin (2025), we offer a representation of syntactic dependency structures inspired by de Groote (2023b) that overcomes these limitations.

MOTS-CLÉS : Grammaires Catégorielles Abstraites, Théorie Sens-Texte, Syntaxe de Dépendance.

KEYWORDS: Abstract Catégrial Grammars, Meaning-Text Theory, Dependency Syntax.

1 Introduction

Cousin (2023b,a, 2025) présente une implémentation de la théorie sens-texte (TST) dans les grammaires catégorielles abstraites (ACG). Tout en reprenant la même composition d'ACG, nous proposons une représentation des structures en dépendance inspirée de de Groote (2023a,b). Ces modifications concernent les représentations syntaxiques de la TST. Ces travaux s'inscrivent dans un contexte de génération de texte avec des méthodes formelles. Des motivations similaires aux nôtres peuvent être retrouvées dans Grammatical Framework (Ranta, 2004). Nous souhaitons en effet avoir un fort contrôle sur le texte généré afin de s'assurer qu'il traduit effectivement le message que l'on voulait transmettre. Un tel système pourrait être utile dans certains domaines de traduction spécialisée, où une grande précision est demandée. Nous souhaitons également mettre en œuvre les structures linguistiques de la TST au travers desquelles s'exerce ce contrôle. Nous nous appuyons pour ce faire sur un modèle formel, celui des ACG. C'est ce modèle que nous étudions ici. De nombreuses propriétés formelles

Actes des 32ème Conférence sur le Traitement Automatique des Langues Naturelles (TALN), volume 1 : articles scientifiques originaux, pages 715–728.

des ACG sont déjà connues ; nous nous intéressons à la manière dont les structures linguistiques peuvent être exprimées dans les ACG, en particulier la TST.

La TST (Mel'čuk *et al.* (2012, 2013, 2015), cf. section 2) est une théorie linguistique qui décrit le lien entre le sens et le texte d'un énoncé. La TST peut être utilisée en génération comme en analyse de texte. Les systèmes présentés par Lareau *et al.* (2018) et Lareau *et al.* (2018) sont deux exemples d'application de la TST en génération de texte. Elle est composée de 7 niveaux de représentations (niveaux sémantique, syntaxiques, morphologiques, phonologiques) et de modules de transitions entre chacun de ces niveaux. À chaque niveau correspond une représentation de l'énoncé, composée d'au moins une structure prédicative et d'une structure communicative. Les ACG (de Groote (2001), cf. section 3) sont un formalisme grammatical basé sur le λ-calcul. Elles permettent la représentation d'autres formalismes grammaticaux comme le montre Pogodalla (2017) avec son implémentation des TAG, et peuvent être également utilisées en génération comme en analyse (Kanazawa, 2007). Dans cet article, nous nous intéressons à une implémentation de la TST dans les ACG. L'utilisation de ce formalisme grammatical a deux principaux avantages. Le premier est que l'encodage réalisé est réversible : le système peut être utilisé en génération comme en analyse de texte. Le second est qu'on retrouve des similarités avec d'autres théories ou grammaires plus traditionnelles, ce qui permet une comparaison avec d'autres formalismes, et d'utiliser ces mêmes formalismes pour certaines parties de notre système. C'est par exemple le cas de la modification (adjective ou adverbiale) présentée en section 5.3 et inspirée de l'encodage des TAG de (Pogodalla, 2017).

Nos précédents travaux (Cousin, 2023a,b) décrivent une première implémentation de la TST dans les ACG. Cependant, seules les structures prédicatives de la TST y sont prises en compte et la structure communicative y est ignorée. Le traitement de phénomènes lexicaux, tels que les collocations, ou encore la génération de paraphrases sont aussi permis, en particulier grâce à l'encodage des fonctions lexicales. (Cousin, 2025) reprend cette implémentation pour prendre en compte la structure communicative. L'opposition thème-rhème (Polguère, 1990) comprise dans la structure communicative joue un rôle crucial dans la détermination de l'arbre de syntaxe profonde associé au graphe sémantique d'un énoncé. À partir d'une même structure sémantique prédicative, selon la structure communicative, l'expression obtenue ne sera pas la même. C'est cette opposition, qui, par exemple, fait apparaître ou non une copule en syntaxe profonde lorsqu'un adjectif s'applique à un nom, et décide si l'adjectif s'exprime sous forme d'épithète (« *[the* ***purple*** *dragon]*$_{Rheme}$ ») ou en tant qu'attribut du sujet (« *[the dragon]*$_{Theme}$ *[is* ***purple****]*$_{Rheme}$ »). Deux expressions obtenues à partir d'une même structure sémantique prédicative, mais avec deux structures communicatives différentes ne sont donc pas paraphrases l'une de l'autre. Nous ne détaillerons pas plus cette partie, mais souhaitons ainsi garder l'ajout de (Cousin, 2025) de la structure communicative. Nous nous inspirons ici de ces travaux en ce qui concerne l'architecture et la composition des différentes ACG utilisées.

(de Groote, 2023a,b) présente l'encodage en ACG d'une interface syntaxe-sémantique entre une théorie sémantique inspirée de la sémantique de Montague et des structures en dépendance. Nous utilisons un fonctionnement similaire afin de pouvoir ajouter des dépendants non obligatoires à un gouverneur, comme des adjectifs ou des adverbes. Cependant, le système détaillé dans cet article présente plusieurs différences avec celui décrit par de Groote (2023b) :

— contrairement à de Groote (2023b) qui utilise des ACG complexes (dont l'analyse n'est pas polynomiale, et pas implantée en général (de Groote, 2015)), nous souhaitons utiliser des ACG d'ordre au plus deux,
— l'implémentation de de Groote (2023b) a pour but l'analyse (de la syntaxe vers la sémantique), or nous avons pour but le sens inverse, la génération (de la sémantique vers la syntaxe),
— de Groote définit le principe de cohérence : quel que soit l'ordre (de calcul) dans lequel

les dépendants sont reliés au gouverneur, les structures calculés par de Groote (2023b) sont équivalentes et génèrent la même interprétation sémantique. Comme nous nous intéressons à la génération (soit au sens inverse de calcul de de Groote (2023b)), et que respecter le principe de cohérence impliquerait l'usage d'ACG dont l'analyse n'est pas décidable, le sytème décrit dans cet article ne respectera pas le principe de cohérence de de Groote.

Nos contributions par rapport à l'existant sont les suivantes :

— des structures syntaxiques en dépendance (profondes et de surface, on ne parlera ici que de surface), moins générales que celles de de Groote, mais dont l'analyse est décidable et polynomiale,
— des ACG décidables, d'une classe particulière dite d'ordre deux (alors que Cousin (2025) avait de l'ordre supérieur),
— une implémentation qui s'inscrit naturellement dans l'architecture décrite dans (Cousin, 2023a, 2025) et avec une meilleure prise en compte des structures prédicatives multiples exprimées par des modifieurs (adjectifs, adverbes).

Les sections 2 et 3 présentent la TST et les ACG. Les sections 4 et 5 sont le cœur de cet article et détaillent cette nouvelle implémentation. La section 6 présente les résultats obtenus avant de conclure.

2 Théorie Sens-Texte (TST)

La TST (Mel'čuk *et al.*, 2012; Milićević, 2006) est une théorie linguistique visant à faire le lien entre le sens et le texte d'un énoncé. Le sens est le contenu linguistique à communiquer, alors que le texte est un fragment de discours (Milićević, 2006). La TST utilise pour ce faire un modèle sens-texte (MST, cf. figure 1) composé de 7 niveaux de représentation et de 6 modules de transition. Elle utilise également les concepts clés de fonctions lexicales (FL, Mel'Čuk & Polguère (2021)) et de paraphrase (Milićević, 2007; Iordanskaja *et al.*, 1991) que nous ne détaillerons pas ici.

Niveaux de représentation et modules de transition : Le MST (cf. figure 1) est composé de 7 niveaux de représentation, correspondant chacun à une représentation éponyme : les représentations sémantique (SemR), syntaxique profonde (DSyntR), syntaxique de surface (SSyntR), morphologiques profonde et de surface (resp. DMorphR et SMorphR), phonologiques profonde et de surface (resp. DPhonR et SPhonR). Entre deux niveaux adjacents se trouve un module de transition, portant le nom du niveau le plus proche de la sémantique. Ainsi, le module syntaxique profond fait la transition entre DSyntR et SSyntR.

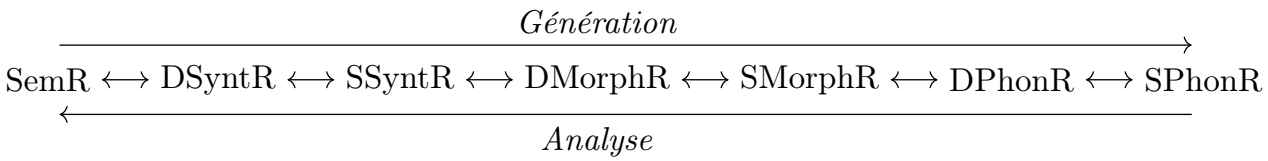


FIGURE 1 – Schéma du modèle sens-texte (Mel'čuk *et al.*, 2012)

Chaque niveau de représentation est constitué d'une structure prédicative (des graphes pour la SemR, des arbres pour les DSyntR et SSyntR, et des chaînes de caractères pour les 4 autres), mais également d'une structure communicative. Cette dernière comporte entre autres l'opposition thème-rhème qui va venir annoter la structure prédicative du niveau de représentation (Polguère, 1990). Un

module de transition effectue toutes les opérations nécessaires au changement de représentation entre deux niveaux. Il s'occupe entre autres du changement de structures prédicatives, tel que le module sémantique qui transforme les graphes de la SemR en arbres pour la DSyntR.

Représentation syntaxique de surface : La SSyntR (Milićević, 2006) est composée de quatre sous-structures (cf. figure 2), la structure syntaxique de surface (SSyntS), la structure syntaxique communicative de surface (SSynt-CommS), la structure syntaxique prosodique de surface (SSynt-ProsS), et la structure syntaxique anaphorique de surface (SSynt-AnaphS). Nous ne considérons que la SSyntS dans cet article.

SSyntR = ⟨ SSyntS, SSynt-CommS, SSynt-ProsS, SSynt-AnaphS ⟩

FIGURE 2 – Composition de la représentation syntaxique de surface

Les SSyntS sont des arbres dont les nœuds sont des lexèmes de surface et dont les branches sont étiquetées par des relations de surface. Plusieurs exemples de SSyntS sont donnés en section 5. Là où la syntaxe profonde avait des relations assez abstraites ($actant_1$, $actant_2$, etc.), la syntaxe de surface a des relations plus explicites (**Sujet**, **ObjetDirect**, etc.). De plus, les lexèmes sont plus détaillés dans les SSyntS. On y retrouve les prépositions (qui étaient absentes des DSyntS), les expressions idiomatiques (qui étaient représentées par un unique lexème dans les DSyntS) sont ici représentées par autant de lexèmes que de mots qui les composent. On peut également remarquer l'absence de FL.

Module syntaxique profond : Le module syntaxique profond effectue la transition entre la DSyntR et la SSyntR. Il effectue certaines opérations structurelles, liées à l'ajout des lexèmes qui n'apparaissaient pas en syntaxe profonde. En effet, ajouter un lexème signifie ajouter au moins une relation syntaxique de surface, et modifie la structure de l'arbre. Il réalise également les FL. Si, par exemple, on avait `anti`(CALM) dans une DSyntR, on aurait UPSET dans la SSyntR correspondante. En plus des transformations structurelles et des modifications de lexèmes, le module syntaxique profond effectue également une opération de paraphrase (Iordanskaja *et al.*, 1991) dans le choix des SSyntR (plusieurs SSyntR peuvent correspondre à une même DSyntR).

3 Grammaires Catégorielles Abstraites (ACG)

Les ACG (de Groote (2001), dont nous reprenons les définitions ici) sont un formalisme grammatical basé sur le λ-calcul permettant de représenter d'autres formalismes grammaticaux. Une ACG est composée de deux langages (les langages abstrait et objet), tous deux reliés par un lexique. Le lexique permet l'interprétation des termes du langage abstrait par des termes du langage objet. Le langage abstrait correspond à l'ensemble des structures grammaticales abstraites, telles que les arbres d'analyse par exemple. Le langage objet correspond à l'ensemble des réalisations concrètes des structures du langage abstrait, comme les chaînes de caractères par exemple.

La définition 3 du lexique repose sur la définition 2 des signatures, qui elle-même repose sur la définition 1 des types implicatifs linéaires.

Définition 1 *Soit A un ensemble de types atomiques. $\mathcal{T}(A)$ est appelé l'ensemble des* ***types implicatifs linéaires*** *et est obtenu par induction sur A :*

— *si* $a \in A$ *alors* $a \in \mathcal{T}(A)$
— *si* $\alpha, \beta \in \mathcal{T}(A)$ *alors* $\alpha \rightarrow \beta \in \mathcal{T}(A)$

Définition 2 *Une* ***signature d'ordre supérieur*** Σ *est un triplet* $\Sigma = \langle A, C, \tau \rangle$*, où :*
— A *est l'ensemble des types atomiques,*
— C *est un ensemble de constantes,*
— $\tau : C \longrightarrow \mathcal{T}(A)$ *est une fonction qui associe un type aux constantes.*
La notation $\vdash_\Sigma t : s$ *indique que le type du* λ*-terme* t *dans* Σ *est* s *(ou* $t : s$ *s'il n'y a pas d'ambiguité sur la signature considérée). On note* $\Lambda(\Sigma)$ *l'ensemble des* λ*-termes bien typés obtenus en utilisant l'ensemble C, les variables, les abstractions et les applications.*

Par exemple, $dragon^{ss} : NP$ signifie que la constante $dragon^{ss}$ est de type NP et $invite^{ss} : AdvP \rightarrow NP \rightarrow NP \rightarrow S$ signifie que la constante $invite^{ss}$ est de type $AdvP \rightarrow NP \rightarrow NP \rightarrow S$, soit qu'elle attend un groupe adverbial ($AdvP$) et deux groupes nominaux (NP) pour former une phrase (S). $\Sigma_{\textit{surface-syntactic}}$ et $\Sigma_{\textit{ssynt-tree}}$, illustrées en figure 3, sont deux signatures d'ordre supérieur dont un extrait est donné en annexe, figure 7.

Définition 3 *Soient* Σ_1 *et* Σ_2 *deux signatures. Un* ***lexique*** $\mathcal{L}_{12}$ *de* Σ_1 *vers* Σ_2 *est une paire de morphismes* $\langle F, G \rangle$ *tels que* $F : \tau(A_1) \longrightarrow \tau(A_2)$ *et* $G : \Lambda(\Sigma_1) \longrightarrow \Lambda(\Sigma_2)$*.*
On note $\mathcal{L}_{12}(t) = \gamma$ *pour signifier que l'interprétation de* t *par* $\mathcal{L}_{12}$ *est* γ*, que ce soit par* F *si* t *et* γ *sont des types ou par* G *si ce sont des termes (ou* $t := \gamma$ *s'il n'y a pas d'ambiguité sur le lexique utilisé).*

Ainsi, $\mathcal{L}_{ssyntRel}(dragon^{ss}) = \lambda A.\ A\ dragon^{st}$ signifie que la constante $dragon^{ss}$ de $\Sigma_{\textit{surface-syntactic}}$ sera interprétée par le terme $\lambda A.\ A\ dragon^{st}$ de $\Lambda(\Sigma_{ssynt-tree})$. Dans la figure 3, $\mathcal{L}_{\textit{ssyntRel}}$ est le lexique allant de $\Sigma_{\textit{surface-syntactic}}$ dans $\Sigma_{\textit{ssynt-tree}}$. Un extrait de $\mathcal{L}_{\textit{ssyntRel}}$ est donné en annexe, figure 8.

Définition 4 *Une* ***grammaire catégorielle abstraite*** *est un quadruplet* $\mathcal{G} = \langle \Sigma_1, \Sigma_2, \mathcal{L}_{12}, s \rangle$ *où :*
— $\Sigma_1 = \langle A_1, C_1, \tau_1 \rangle$ *et* $\Sigma_2 = \langle A_2, C_2, \tau_2 \rangle$ *sont deux signatures d'ordre supérieur,*
— $\mathcal{L}_{12} = \Sigma_1 \longrightarrow \Sigma_2$ *est un lexique,*
— $s \in \mathcal{T}(A_1)$ *est le type distingué de la grammaire.*

Définition 5 *Le* ***langage abstrait*** $\mathcal{A}$ *et le* ***langage objet*** $\mathcal{O}$ *d'une ACG* $\mathcal{G}_{12} = \langle \Sigma_1, \Sigma_2, \mathcal{L}_{12}, s \rangle$ *sont :*
— $\mathcal{A}(\mathcal{G}_{12}) = \{t \in \Lambda(\Sigma_1) \mid \vdash_{\Sigma_1} t : s \textit{ est dérivable}\}$
— $\mathcal{O}(\mathcal{G}_{12}) = \{t \in \Lambda(\Sigma_2) \mid \exists u \in \mathcal{A}(\mathcal{G}_{12}) \textit{ tel que } t = \mathcal{L}_{12}(u)\}$
On appelle ***analyse*** *(ou parsing) l'opération d'inversion de morphismes, notée* $\mathcal{L}_{12}^{-1}$*, telle que* $t = \mathcal{L}_{12}^{-1}(\gamma)$ *ssi* $\mathcal{L}_{12}(t) = \gamma$*, avec* $t \in \Lambda(\Sigma_1)$ *et* $\gamma \in \Lambda(\Sigma_2)$*.*

Les langages abstrait et objet d'une ACG sont des ensembles de λ-termes obtenus par induction sur les signatures abstraite et objet de cette ACG. Par exemple, l'ensemble des arbres de syntaxe de surface (SSyntS) constitue le langage objet de l'ACG composée de $\Sigma_{\textit{surface-syntactic}}$ (sa signature abstraite), $\Sigma_{\textit{ssynt-tree}}$ (sa signature objet) et $\mathcal{L}_{\textit{ssyntRel}}$ et illustrée en figure 3. Les ACG ont la propriété de pouvoir se composer (cf. figure 3). Leurs lexiques sont inversibles. On peut appliquer un lexique de la signature abstraite à la signature objet ($\mathcal{L}_{ssyntRel}(\gamma^{ss}) = \gamma^{st}$) et inverser un lexique ($\mathcal{L}_{ss2ds}^{-1}(\gamma^{d0f}) = \{\gamma^{ss}\}$). On parle dans le premier cas d'opération d'application, et dans le second d'opération de parsing ou d'analyse. Ces deux opérations peuvent être composées pour effectuer une opération de transduction : $\mathcal{L}_{ssyntRel}(\mathcal{L}_{ss2ds}^{-1}(\gamma^{d0f})) = \{\gamma^{st}\}$.

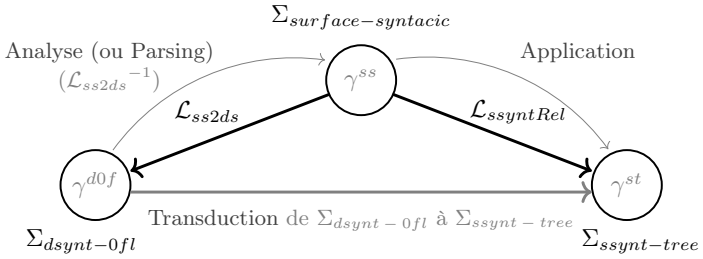


FIGURE 3 – Illustration des opérations d'application, d'analyse (parsing) et de transduction

Une ACG est caractérisée par son ordre et sa complexité. La complexité d'une ACG concerne la complexité de l'analyse du lexique de l'ACG en question.

Définition 6 *L'ordre d'un type $\tau \in \mathcal{T}(A)$ est défini par induction :*
— $ordre(\tau) = 1$ *si* $\tau \in A$,
— $ordre(\alpha \to \beta) = max(1 + ordre(\alpha), ordre(\beta))$ *sinon.*
*L'ordre d'une constante abstraite est l'ordre de son type τ, et l'**ordre d'une ACG** est le maximum des ordres de ses constantes abstraites.*
*La **complexité** d'une ACG est le maximum des ordres des réalisations de ses types atomiques. Une ACG d'ordre γ et de complexité η est notée $ACG_{(\gamma,\eta)}$.*

Nous souhaitons utiliser une classe particulière des ACG, les ACG dites d'ordre deux, car la complexité de leur analyse est polynomiale (Kanazawa, 2007; Salvati, 2005).

4 Architecture et composition des ACG

Nous reprenons l'architecture de (Cousin, 2025) (qui était celle de (Cousin, 2023b,a) en ajoutant l'opposition communicative thème-rhème à la transduction entre la sémantique et la syntaxe profonde). Plusieurs ACG sont remplacées ou modifiées afin de modéliser les arbres de dépendances des DSyntR et SSyntR en s'inspirant des ACG de (de Groote, 2023a,b). Nous utilisons le logiciel ACGtk (Guillaume *et al.*, 2024) pour mettre en œuvre notre modèle.

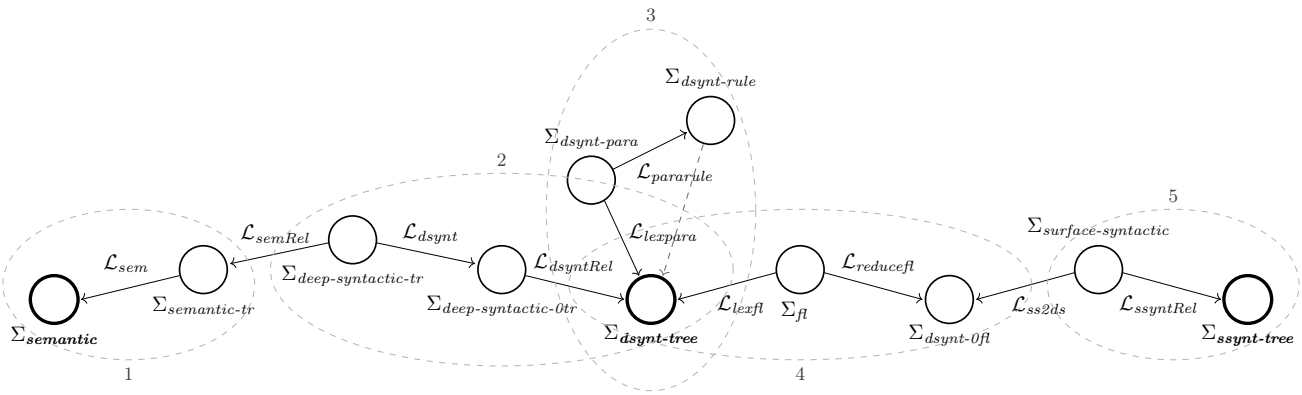


FIGURE 4 – Composition des ACG de l'implémentation de MTT

La figure 4 illustre l'architecture et la composition des ACG de (Cousin, 2023a, 2025) que nous

reprenons. Elle est divisée en cinq zones :

— zones 1 et 2 : la transduction entre $\Sigma_{semantic\text{-}tr}$ et $\Sigma_{dsynt\text{-}tree}$ permet la transduction entre les représentations des SemR et celles des DSyntS. La zone 1 correspond à la sémantique, la zone 2 à la syntaxe profonde.
— zone 3 : la transduction entre $\Sigma_{dsynt\text{-}tree}$ et $\Sigma_{dsynt\text{-}rule}$ permet d'effectuer la paraphrase syntaxique profonde.
— zone 4 : la transduction entre $\Sigma_{dsynt\text{-}tree}$ et $\Sigma_{dsynt\text{-}0fl}$ permet de réaliser les éventuelles FL présentes dans les DSyntS, et d'effectuer la première étape (dans cette implémentation) du module syntaxique profond (cf. section 2).
— zone 5 : la transduction entre $\Sigma_{dsynt\text{-}0fl}$ et $\Sigma_{ssynt\text{-}tree}$ permet la deuxième étape (dans cette implémentation) du module syntaxique profond, en effectuant le changement de structures entre la DSyntS et la SSyntS. La zone 5 correspond à la syntaxe de surface.

Nos modifications portent sur les zones 2, 4 et 5. Nous présenterons dans cet article uniquement celles portant sur la zone 5, soit sur la syntaxe de surface [1].

Notations : Dans ce qui suit, $\Sigma_{surface\text{-}syntactic}$ sera notée Σ_{ss}, et $\Sigma_{ssynt\text{-}tree}$ sera notée Σ_{st}. Une constante représentant le lexème LEX dans la signature Σ_S sera notée lex^S. Ainsi, $dragon^{ss}$ est la constante représentant le lexème DRAGON dans Σ_{ss}. Un arbre γ représentant une expression E dans le langage obtenu par induction sur une signature Σ_S sera noté γ^S_E. Ainsi, le terme encodant la SSyntR associée à l'expression (cpd) : « *the calm purple dragon* » sera noté γ^{st}_{cpd}.

5 Implémentation de la syntaxe de surface

Nous adaptons l'implémentation de de Groote (2023b), qui permet de dériver des structures sémantiques en dépendance, aux ACG encodant les représentations de dépendances syntaxiques de la TST. Les figures 7 et 8 en annexe donnent des extraits de Σ_{ss}, Σ_{st} et $\mathcal{L}_{ssyntRel}$ utilisés dans cette section.

5.1 Encodage des SSyntS (Σ_{st})

Σ_{st} permet de représenter les SSyntS. Ce sont des structures de dépendances dont les relations sont des relations syntaxiques de surface, comme **Subj**, **DirO**, ou **Modif**, pour représenter les relations sujet, objet direct, ou modificateur (adjectif) respectivement. Une relation est une branche, qui part du lexème gouverneur et pointe sur le lexème dépendant.

Nous encodons chaque lexème par une constante de type T. Les catégories grammaticales ne sont pas distinguées dans Σ_{st}.

(1) $$dragon^{st}, invite^{st} : T$$

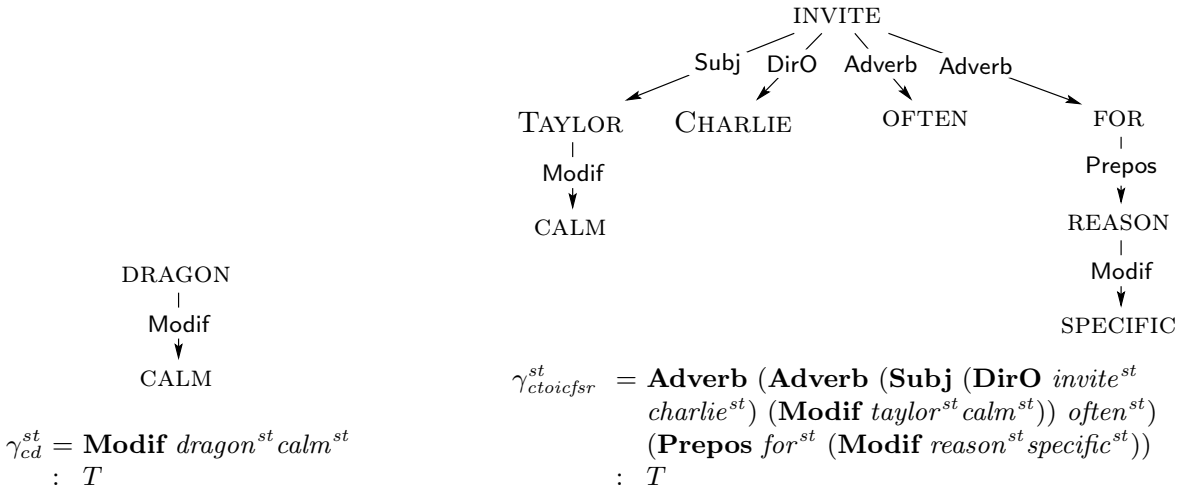
Alors, une relation syntaxique de surface sera encodée par une constante prenant deux autres constantes (représentant respectivement le gouverneur et le dépendant) en argument, et aura

1. La structure communicative n'est pour l'instant prise en compte qu'aux niveaux sémantiques et syntaxiques profonds. La section suivante s'intéressant à la syntaxe de surface, la structure communicative n'y sera pas détaillée. Pour plus de détails quant à l'encodage de l'opposition thème-rhème, le lecteur est invité à lire (Cousin, 2025).

comme type final T. Toute constante représentant une relation syntaxique de surface sera de type $T \to T \to T$.

(2) $$\textbf{Subj}, \textbf{Modif} : T \to T \to T$$

Comme le type final d'une relation est T, le sous-arbre représentant la dépendance (constituée du gouverneur, de la relation et du dépendant) pourra être utilisé comme s'il s'agissait d'un lexème. Le type T est le type associé aux arbres de la SSyntS, qu'il s'agisse d'arbres, de sous-arbres (comme une simple dépendance) ou de noeuds (comme un lexème). La figure 5 donne deux exemples de SSyntS, les termes les encodant dans $\Lambda(\Sigma_{st})$ et les types associés. La figure 5a illustre une SSyntR simple, n'ayant qu'une dépendance, et la figure 5b illustre une SSyntR plus complexe, ayant plusieurs dépendances.



(a) « *the calm dragon* » (γ^{st}_{cd}) (b) « *The calm Taylor often invites Charlie for a specific reason* » ($\gamma^{st}_{ctoicfsr}$)

FIGURE 5 – Représentation de SSyntS associées aux expressions « *the calm dragon* » et « *The calm Taylor often invites Charlie for a specific reason* » et des termes objets (de $\Lambda(\Sigma_{st})$) associés.

5.2 Encodage des contraintes (Σ_{ss} et $\mathcal{L}_{ssyntRel}$)

Comme tous les lexèmes de Σ_{st} sont encodés par des constantes de type T, il est tout à fait possible de construire une structure grammaticalement incorrecte telle qu'illustrée en (3), où un lexème représentant un nom ($dragon^{st}$) possède un verbe ($invite^{st}$) en tant que dépendant sujet, et un adjectif ($purple^{st}$) en tant que dépendant objet.

(3) $$\gamma^{st}_{illicite} = \textbf{Subj}\ (\textbf{DirO}\ dragon^{st}\ purple^{st})\ invite^{st} \quad : T$$

Afin d'éviter l'obtention de telles structures, *nous utilisons la signature abstraite Σ_{ss} pour contrôler les structures* obtenues. Ce contrôle est effectué par *les types des constantes abstraites.*

Chaque prédicat possède un nombre minimal d'actants obligatoires, donné par sa structure prédicative, dont on connaît les catégories grammaticales. Par exemple, dans le cas général, un verbe transitif à l'actif, comme INVITE, possède un groupe nominal sujet et un groupe nominal objet (« X_{sujet} INVITE Y_{objet}. »). De plus, certaines catégories de prédicats peuvent être modifiées : un nom peut

être modifié par un adjectif (« *the purple dragon* »), et un verbe et un adjectif peuvent être modifiés par un adverbe [2] (« *Taylor often invites Charlie* », « *the often calm dragon* »). Ces contraintes sont encodées dans les types des constantes représentant les lexèmes dans Σ_{ss}. Pour chaque constante de Σ_{ss} représentant un prédicat, son type sera composé du type de son éventuel modifieur, puis des types respectifs de ses arguments obligatoires, et de son type final. Les modifieurs eux-mêmes représentent un cas particulier et sont traités dans la section suivante. Les types généraux des verbes transitifs à l'actif (comme $invite^{ss}_{12,act}$) et des noms (comme $dragon^{ss}$) sont donnés en (4) ci-dessous.

$$\begin{aligned} invite^{ss}_{12,act} &: AdvP \rightarrow NP \rightarrow NP \rightarrow S \\ dragon^{ss} &: AdjP \rightarrow NP \end{aligned} \tag{4}$$

Enfin, le lexique $\mathcal{L}_{ssyntRel}$ encode les interprétations des constantes de Σ_{ss} par des termes de $\Lambda(\Sigma_{st})$ leur correspondant comme illustré en (5) ci-dessous [3].

$$\begin{aligned} \mathcal{L}_{ssyntRel}(invite^{ss}_{12,act}) &:= \lambda\, A\; x\; y.\; A\; (\textbf{Subj}\; (\textbf{DirO}\; invite^{st}\; y)\; x) \\ \mathcal{L}_{ssyntRel}(dragon^{ss}) &:= \lambda\, A.\; A\; dragon^{st} \end{aligned} \tag{5}$$

On aura donc les égalités suivantes [4] :

$$\begin{aligned} \mathcal{L}_{ssyntRel}(dragon^{ss}(calm^{ss}\, I^{ss}_{adv} I^{ss}_{adj})) &= \gamma^{st}_{cd} \\ \mathcal{L}_{ssyntRel}(invite^{ss}_{12,act} I^{ss}_{adv}(taylor^{ss}\, I^{ss}_{adj}))(charlie^{ss}\, I^{ss}_{adj}) &= \textbf{Subj}(\textbf{DirO}\; invite^{st}\, charlie^{st})\; taylor^{st} \end{aligned} \tag{6}$$

Comme Σ_{ss} encode et contraint la nature grammaticale des dépendants de chaque prédicat et que $\mathcal{L}_{ssyntRel}$ interprète chaque prédicat par sa structure de dépendance, les termes grammaticalement incorrects tels que $\gamma^{st}_{illicite}$ ne pourront pas être obtenus lors de la génération. Réciproquement, en analyse, ils n'auront aucun antécédent par parsing de $\mathcal{L}_{ssyntRel}$ dans $\Lambda(\Sigma_{ss})$.

Les interprétations par $\mathcal{L}_{ssyntRel}$ de certaines constantes de cette implémentation sont assez similaires à celles de de Groote (2023b). C'est le cas des noms par exemple. Cependant, il encode les relations de dépendance dans la signature abstraite, alors que nous les encodons dans la signature objet. Dans (de Groote, 2023b) ce sont les relations qui contraignent les natures grammaticales de leurs dépendants. Nous nous différencions de (de Groote, 2023b) en encodant cette contrainte dans les prédicats de la signature abstraite et non dans les relations de dépendance.

De plus, (de Groote, 2023b) respecte le principe de cohérence : quel que soit l'ordre dans lequel les dépendances sont calculées, les interprétations des termes objets par le lexique seront équivalentes. Pour respecter ce principe, il utilise des constantes affines. Nous nous imposons comme contraintes d'utiliser des constantes linéaires (donc non affines) et d'avoir des ACG d'ordre 2 (pour des soucis de complexités de calcul de l'analyse des lexiques et pour pouvoir implémenter notre modèle dans ACGtk et automatiser les calculs). De plus l'ordre dans lequel sont calculées les relations de dépendance (syntaxiques de surface) importe ici. En effet, lors de la transition vers la DMorphR, la SSyntR sera linéarisée et les règles de linéarisation prennent en compte l'ordre de lecture des relations. Cet ordre de lecture est représenté par l'ordre de calcul et est traduit par l'interprétation des constantes dans $\mathcal{L}_{ssyntRel}$. Deux représentations, possédant les mêmes dépendances mais dans des ordres différents, ne doivent par conséquent pas être équivalentes : **DirO** (**Subj** $invite^{st}\; taylor^{st}$) $charlie^{st} \neq$ **Subj** (**DirO** $invite^{st}\; charlie^{st}$) $taylor^{st}$. Les ACG présentées ici ne respectent donc pas le principe de cohérence.

2. Un adverbe peut également modifier un autre adverbe. Pour des raisons de lisibilité et complexité calculatoire, nous ne considèrerons pas ce cas ici. Cependant, cette modification (d'un groupe adverbial par un groupe adverbial) peut être traitée de la même manière que la modification d'un adjectif par un groupe adverbial (voir section 5.3).

3. Les variables A abstraites de (5) encodent les modifieurs. La section 5.3 explique leur fonctionnement.

4. Les constantes I^{ss}_{adv} et I^{ss}_{adj} sont de types respectifs $AdvP$ et $AdjP$ et sont interprétées par $\mathcal{L}_{ssyntRel}$ comme l'identité.

5.3 Comportement des modifieurs : adverbes et adjectifs

Les modifieurs (adverbes et adjectifs) ont un comportement particulier. Ils ne font pas partie des dépendants obligatoires d'un prédicat, mais un prédicat peut avoir plusieurs modifieurs parmi ses dépendants, comme l'illustrent les figures 6a, 5a et 6b par exemple.

$\gamma^{st}_{d} = dragon^{st} : T$

(a) « *the dragon* » (γ^{st}_{d})

$\gamma^{st}_{cpd} = \mathbf{Modif}\ (\mathbf{Modif}\ dragon^{st}\ calm^{st})\ purple^{st} : T$

(b) « *the calm purple dragon* » (γ^{st}_{cpd})

FIGURE 6 – Représentation de SSyntS associées aux expressions nominales « *the dragon* », « *the calm purple dragon* » et des termes objets (de $\Lambda(\Sigma_{st})$) associés.

Un adverbe a traditionnellement en sémantique de Montague ou en grammaires catégorielles un type d'ordre supérieur, comme $(NP \to S) \to (NP \to S)$ (Carpenter, 1998, p.120). Nous utilisons la modélisation des modifieurs en TAG de l'approche de (Pogodalla, 2017) afin d'encoder les adjectifs et adverbes tout en gardant de l'ordre 2. Par exemple, une constante de Σ_{ss} représentant un adverbe aura pour type $AdvP \to AdvP$ [5], avec $\mathcal{L}_{ssyntRel}(AdvP) = T \to T$.

Les constantes de Σ_{st} représentant les adverbes sont de type T, comme les autres lexèmes. (7) illustre les types des constantes encodant les adverbes (tels que $often^{ss}$) dans Σ_{ss} ainsi que leur interprétation par $\mathcal{L}_{ssyntRel}$. Ces constantes sont bien d'ordre 2.

$$often^{ss} : AdvP \to AdvP$$
$$\mathcal{L}_{ssyntRel}(often^{ss}) := \lambda A\ v.\ A\ (\mathbf{Adverb}\ v\ often^{st}) \tag{7}$$

De plus, on a $\mathcal{L}_{ssyntRel}(AdvP) = T \to T$. Ainsi, $\mathcal{L}_{ssyntRel}(AdvP \to AdvP) = (T \to T) \to (T \to T)$, soit l'interprétation qu'aurait eu un type plus traditionnel (tel que $(NP \to S) \to (NP \to S)$) par $\mathcal{L}_{ssyntRel}$.

Le cas des adjectifs est similaire à celui des adverbes, à cela près qu'un adjectif peut-être modifié par un adverbe. Le type de la constante abstraite encodant un adjectif sera donc composé du type d'un potentiel modifieur adverbial $AdvP$, puis du reste de son type $AdjP \to AdjP$, comme illustré en (8) :

$$purple^{ss} : AdvP \to AdjP \to AdjP$$
$$\mathcal{L}_{ssyntRel}(purple^{ss}) := \lambda\ M\ A\ n.\ A\ (\mathbf{Modif}\ n\ (M\ purple^{st})) \tag{8}$$

Comme chaque lexème modifiable (les noms par des adjectifs, les verbes et adjectifs par des adverbes) a son premier argument du type du modifieur (cf. (4)), cela permet d'utiliser, par exemple, un adjectif comme en (6) et figure 5a, plusieurs comme en (9)[4] et figure 6b, ou aucun adjectif comme en (9) et figure 6a.

$$\mathcal{L}_{ssyntRel}(dragon^{ss}\, I_{adj}) = \gamma^{st}_{d}$$
$$\mathcal{L}_{ssyntRel}(dragon^{ss}(calm^{ss}\, I^{ss}_{adv}(purple^{ss}\, I^{ss}_{adv}\, I^{ss}_{adj}))) = \gamma^{st}_{cpd} \tag{9}$$

5. Un adverbe pouvant également modifier un autre adverbe, son type devrait être $AdvP \to AdvP \to AdvP$, avec le premier $AdvP$ resprésentant le modifieur de l'adverbe. Pour des raisons de lisibilité et complexité calculatoire (comme pour la note 2), nous utiliserons le type $AdvP \to AdvP$ à la place.

Il en est de même pour les adverbes. Par exemple, (10)[4] donne le terme abstrait qui sera interprété par $\mathcal{L}_{ssyntRel}$ comme $\gamma^{st}_{ctoicfsr}$ (cf. figure 5b), où deux groupes adverbiaux modifient le verbe.

$$
\begin{aligned}
\gamma^{ss}_{ctoicfsr} &= invite^{ss}_{12,act}(often^{ss}(for-reason^{ss}(specific^{ss}I^{ss}_{adv}I^{ss}_{adj})I^{ss}_{adv})) \\
&\quad (taylor^{ss}(calm^{ss}I^{ss}_{adv}I^{ss}_{adj}))(charlie^{ss}I^{ss}_{adj}) \\
\mathcal{L}_{ssyntRel}(\gamma^{ss}_{ctoicfsr}) &= \gamma^{st}_{ctoicfsr}
\end{aligned} \tag{10}
$$

6 Résultats et Conclusion

Nous manipulons ici des lexiques ayant entre 30 et 50 lexèmes environ, qui permettent de modéliser les concepts de FL, paraphrase sémantique et syntaxique de surface, la structure communicative et plusieurs phénomènes lexicaux comme la synonymie, les expressions idiomatiques, ou le traitement des modifieurs tels que les adjectifs et les groupes adverbiaux. Il s'agit d'une grammaire-jouet, mais il serait possible d'extraire des données de SUD (Gerdes *et al.*, 2018) ou du réseau lexical du français (ATILF, 2025) afin d'étendre notre couverture. Les structures syntaxiques de la TST sont des structures en dépendance, dont les structures sont en accord avec celles de SUD. Les étiquettes des relations sont différentes, que ce soit en syntaxe profonde ou en syntaxe de surface, mais la structure de l'arbre syntaxique est similaire. Les représentations de l'implémentation présentée ici sont proches de ce qu'on retrouve dans les corpus annotés, comme ceux disponibles dans Grew (Guillaume, 2021). Nos ACG sont testées par un jeu de tests sur 61 termes issus de $\Lambda(\Sigma_{deep-syntactic-tr})$. Les scripts effectuent les transductions jusqu'à $\Sigma_{semantic}$ et Σ_{st}. Pour les 61 structures en entrée de ces tests, 372 structures prédicatives syntaxiques de surface sont générées et vérifiées manuellement.

(Cousin, 2023b,a, 2025) présentent une architecture permettant la paraphrase syntaxique profonde (zone 3 de la figure 4). Nous n'avons pas encore pu nous pencher sur cette question et cette étape n'est donc pas encore complètement permise par notre implémentation.

Inspirée de (de Groote, 2023b), l'implémentation présentée dans cet article est plus proche d'implémentations plus classiques telles que celle des TAG de (Pogodalla, 2017). Contrairement aux implémentations de (Cousin, 2023b,a, 2025), cette implémentation est d'ordre 2, et non plus d'ordre supérieur. Cela permet, outre une complexité polynomiale des ACG, le parsing de tous les lexiques utilisés. De plus, ce ne sont plus les lexèmes qui portent et définissent leur nombre de dépendants. Les constantes représentant les lexèmes contraignent leur nombre minimal de dépendants (deux pour INVITE) et la nature grammaticale de ceux-ci en syntaxe de surface (NP pour chacun des dépendants obligatoires de INVITE) dans le type des constantes abstraites associées. L'ajout de dépendants en plus, comme des adjectifs ou des adverbes (cf. section 5) est permis.

Remerciements

Je remercie Sylvain Pogodalla (Université de Lorraine, CNRS, Inria, LORIA) pour ses commentaires et remarques constructives.

Références

ATILF (2025). Réseau lexical du français (rl-fr). ORTOLANG (Open Resources and TOols for LANGuage) –www.ortolang.fr.

CARPENTER B. (1998). Chapter 4 : Applicative categorial grammar. In *Type-Logical Semantics*. MIT Press. DOI : 10.7551/mitpress/6945.003.0006.

COUSIN M. (2023a). Meaning-Text Theory within Abstract Categorial Grammars : Towards Paraphrase and Lexical Function Modeling for Text Generation. In M. AMBLARD & H. BREITHOLTZ, Éds., *IWCS 2023 - 15th International Conference on Computational Semantics*, Nancy, France : Association for Computational Linguistics. HAL : hal-04104453.

COUSIN M. (2023b). Vers une implémentation de la théorie sens-texte avec les grammaires catégorielles abstraites. In M. CANDITO, T. GERALD & J. G. MORENO, Éds., *18e Conférence en Recherche d'Information et Applications – 16e Rencontres Jeunes Chercheurs en RI – 30e Conférence sur le Traitement Automatique des Langues Naturelles – 25e Rencontre des Étudiants Chercheurs en Informatique pour le Traitement Automatique des Langues*, p. 72–86, Paris, France : ATALA. HAL : hal-04100197.

COUSIN M. (2025). Adding communicative structure to the MTT into ACG encoding. In *Congreso Internacional sobre Estudios Teóricos y Aplicados de Léxico, 2025*, Madrid, Spain. Une version longue est en cours de rédaction., HAL : hal-04889333.

DE GROOTE P. (2001). Towards abstract categorial grammars. In *Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics*, p. 252–259, Toulouse, France : Association for Computational Linguistics. DOI : 10.3115/1073012.1073045.

DE GROOTE P. (2015). Abstract categorial parsing as linear logic programming. In M. KUHLMANN, M. KANAZAWA & G. M. KOBELE, Éds., *Proceedings of the 14th Meeting on the Mathematics of Language (MoL 2015)*, p. 15–25, Chicago, USA : Association for Computational Linguistics. DOI : 10.3115/v1/W15-2302.

DE GROOTE P. (2023a). Deriving formal semantic representations from dependency structures. In D. BEKKI, K. MINESHIMA & E. MCCREADY, Éds., *Logic and Engineering of Natural Language Semantics*, p. 157–172, Cham : Springer Nature Switzerland.

DE GROOTE P. (2023b). On the semantics of dependencies : Relative clauses and open clausal complements. In D. BEKKI, K. MINESHIMA & E. MCCREADY, Éds., *Logic and Engineering of Natural Language Semantics - 20th International Conference, LENLS20, Osaka, Japan, November 18-20, 2023, Revised Selected Papers*, volume 14569 de *Lecture Notes in Computer Science*, p. 244–259 : Springer. DOI : 10.1007/978-3-031-60878-0_14.

GERDES K., GUILLAUME B., KAHANE S. & PERRIER G. (2018). SUD or surface-syntactic Universal Dependencies : An annotation scheme near-isomorphic to UD. In M.-C. DE MARNEFFE, T. LYNN & S. SCHUSTER, Éds., *Proceedings of the Second Workshop on Universal Dependencies (UDW 2018)*, p. 66–74, Brussels, Belgium : Association for Computational Linguistics. DOI : 10.18653/v1/W18-6008.

GUILLAUME B. (2021). Graph matching and graph rewriting : GREW tools for corpus exploration, maintenance and conversion. In D. GKATZIA & D. SEDDAH, Éds., *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics : System Demonstrations*, p. 168–175, Online : Association for Computational Linguistics. DOI : 10.18653/v1/2021.eacl-demos.21.

GUILLAUME M., POGODALLA S. & TOURNEUR V. (2024). ACGtk : A Toolkit for Developing and Running Abstract Categorial Grammars. In J. GIBBONS & D. MILLER, Éds., *Functional and Logic Programming. 17th International Symposium, FLOPS 2024*, volume Lecture Notes in Computer Science de *Functional and Logic Programming. 17th International Symposium, FLOPS 2024*, p. 13–30, Kumamoto, Japan : Springer. DOI : 10.1007/978-981-97-2300-3_2, HAL : hal-04479621.

IORDANSKAJA L., KITTREDGE R. & POLGUÈRE A. (1991). *Lexical Selection and Paraphrase in a Meaning-Text Generation Model*, In C. L. PARIS, W. R. SWARTOUT & W. C. MANN, Éds., *Natural Language Generation in Artificial Intelligence and Computational Linguistics*, p. 293–312. Springer US : Boston, MA. DOI : 10.1007/978-1-4757-5945-7_11.

KANAZAWA M. (2007). Parsing and generation as datalog queries. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, p. 176–183, Prague, Czech Republic : Association for Computational Linguistics.

LAREAU F., LAMBREY F., DUBINSKAITE I., GALARRETA-PIQUETTE D. & NEJAT M. (2018). GenDR : A generic deep realizer with complex lexicalization. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan : European Language Resources Association (ELRA).

MEL'ČUK I. & POLGUÈRE A. (2021). Les fonctions lexicales dernier cri. In S. MARENGO, Éd., *La Théorie Sens-Texte. Concepts-clés et applications*, Dixit Grammatica, p. 75–155. L'Harmattan. HAL : hal-03311348.

MEL'ČUK I., MEL'ČUK I., BECK D. & POLGUÈRE A. (2012). *Semantics : From Meaning to Text*, volume 1 de *Semantics : From Meaning to Text*. John Benjamins Publishing Company.

MEL'ČUK I., MEL'ČUK I., BECK D. & POLGUÈRE A. (2013). *Semantics : From Meaning to Text*, volume 2 de *Semantics : From Meaning to Text*. John Benjamins Publishing Company.

MEL'ČUK I., MEL'ČUK I., BECK D. & POLGUÈRE A. (2015). *Semantics : From Meaning to Text*, volume 3 de *Semantics : From Meaning to Text*. John Benjamins Publishing Company.

MILIĆEVIĆ J. (2007). *La paraphrase : modélisation de la paraphrase langagière*. Sciences pour la communication. Lang. DOI : 10.3726/978-3-0352-0096-6.

MILIĆEVIĆ J. (2006). A short guide to the meaning-text linguistic theory. *Journal of Koralex*, **8**, 187–233.

POGODALLA S. (2017). A syntax-semantics interface for Tree-Adjoining Grammars through Abstract Categorial Grammars. *Journal of Language Modelling*, **5**(3), 527–605. DOI : 10.15398/jlm.v5i3.193, HAL : hal-01242154.

POLGUÈRE A. (1990). *Structuration et mise en jeu procédurale d'un modèle linguistique déclaratif dans un cadre de génération de texte*. Thèse de doctorat.

RANTA A. (2004). Grammatical framework. *J. Funct. Program.*, **14**, 145–189. DOI : 10.1017/S0956796803004738.

SALVATI S. (2005). *Problèmes de filtrage et problème d'analyse pour les grammaires catégorielles abstraites*. Thèse de doctorat, Institut National Polytechnique de Lorraine. Thèse de doctorat dirigée par Philippe de Groote.

A Signatures $\Sigma_{\textit{surface-syntacic}}$ et $\Sigma_{\textit{ssynt-tree}}$ et lexique $\mathcal{L}_{\textit{ssyntRel}}$

$\Sigma_{\textit{surface-syntacic}}$

$NP,\ AdjP,\ AdvP,\ S$	$: type$
I^{ss}_{adj}	$: AdjP \to AdjP$
I^{ss}_{adv}	$: AdvP \to AdvP$
$calm^{ss}$	$: AdvP \to AdjP \to AdjP$
$charlie^{ss}$	$: AdjP \to NP$
$invite^{ss}_{12,act}$	$: AdvP \to NP \to NP \to S$
$dragon^{ss}$	$: AdjP \to NP$
$often^{ss}$	$: AdvP \to AdvP$
$purple^{ss}$	$: AdvP \to AdjP \to AdjP$
$for-reason^{ss}$	$: AdjP \to AdvP \to AdvP$
$specific^{ss}$	$: AdvP \to AdjP \to AdjP$
$taylor^{ss}$	$: AdjP \to NP$

$\Sigma_{\textit{ssynt-tree}}$

T	$: type$
Adverb	$: T \to T \to T$
DirO	$: T \to T \to T$
Modif	$: T \to T \to T$
Prepos	$: T \to T \to T$
Subj	$: T \to T \to T$
$calm^{st}$	$: T$
$charlie^{st}$	$: T$
$invite^{st}_{12,act}$	$: T$
$dragon^{st}$	$: T$
for^{st}	$: T$
$often^{st}$	$: T$
$purple^{st}$	$: T$
$reason^{st}$	$: T$
$specific^{st}$	$: T$
$taylor^{st}$	$: T$

FIGURE 7 – Extrait de $\Sigma_{\textit{surface-syntactic}}$ et $\Sigma_{\textit{ssynt-tree}}$

$\mathcal{L}_{\textit{ssyntRel}}$

$NP,\ S$	$:= T$
$AdjP, AdvP$	$:= T \to T$
I^{ss}_{adj}	$:= \lambda n.\ n$
I^{ss}_{adv}	$:= \lambda v.\ v$
$calm^{ss}$	$:= \lambda M\ A\ n.\ A\ (\textbf{Modif}\ n\ (M\ calm^{st}))$
$charlie^{ss}$	$:= \lambda A.\ A\ charlie^{st}$
$invite^{ss}_{12,act}$	$:= \lambda A\ x\ y.\ A\ (\textbf{Subj}\ (\textbf{DirO}\ invite^{st}\ y)\ x)$
$dragon^{ss}$	$:= \lambda A.\ A\ dragon^{st}$
$often^{ss}$	$:= \lambda A\ v.\ A\ (\textbf{Adverb}\ v\ often^{st})$
$purple^{ss}$	$:= \lambda\ M\ A\ n.\ A\ (\textbf{Modif}\ n\ (M\ purple^{st}))$
$for-reason^{ss}$	$:= \lambda x\ A\ v.\ A\ (\textbf{Adverb}\ v\ (\textbf{Prepos}\ for^{st}\ (x\ reason^{st})))$
$specific^{ss}$	$:= \lambda M\ A\ n.\ A\ (\textbf{Modif}\ n\ (M\ specific^{st}))$
$taylor^{ss}$	$:= \lambda A.\ A\ taylor^{st}$

FIGURE 8 – Extrait de $\mathcal{L}_{\textit{ssyntRel}}$, lexique de $\Sigma_{\textit{surface-syntactic}}$ vers $\Sigma_{\textit{ssynt-tree}}$