# Une approche paresseuse de l'analyse sémantique ou comment construire une interface syntaxe-sémantique à partir d'exemples

François-Régis Chaumartin[1,2] Sylvain Kahane[3,2]

(1) Proxem, 7 impasse Dumur, 92110 Clichy
(2) Alpage, Université Paris 7 & INRIA
(3) Modyco, Université Paris Ouest Nanterre & CNRS
frc@proxem.com, sylvain@kahane.fr

**Résumé** Cet article montre comment calculer une interface syntaxe-sémantique à partir d'un analyseur en dépendance quelconque et interchangeable, de ressources lexicales variées et d'une base d'exemples associés à leur représentation sémantique. Chaque exemple permet de construire une règle d'interface. Nos représentations sémantiques sont des graphes hiérarchisés de relations prédicat-argument entre des acceptions lexicales et notre interface syntaxe-sémantique est une grammaire de correspondance polarisée. Nous montrons comment obtenir un système très modulaire en calculant certaines règles par « soustraction » de règles moins modulaires.

**Abstract** This article shows how to extract a syntax-semantics interface starting from an interchangeable dependency parser, various lexical resources and from samples associated with their semantic representations. Each example allows us to build an interface rule. Our semantic representations are hierarchical graphs of predicate-argument relations between lexical meanings and our syntax-semantics interface is a polarized unification grammar. We show how to obtain a very modular system by computing some rules by "subtraction" of less modular rules.

**Mots-clés :** Interface syntaxe-sémantique, graphe sémantique, grammaires de dépendance, GUP (Grammaire d'unification polarisée), GUST (Grammaire d'unification Sens-Texte)

**Keywords:** Syntax-semantics Interface, Semantic Graph, Dependency Grammar, PUG (Polarized Unification Grammar), MTUG (Meaning-Text Unification Grammar)

## Introduction

Une interface syntaxe-sémantique est un module qui prend en entrée les sorties d'un analyseur syntaxique et produit des représentations sémantiques correspondantes qui sont des graphes de relations prédicat-argument entre des acceptions lexicales. Le développement manuel d'un tel module peut être coûteux et il est périlleux de construire une interface syntaxe-sémantique qui s'appuie sur les sorties d'un analyseur syntaxique particulier qui peut rapidement devenir obsolète. Notre objectif est donc de pouvoir extraire une interface-sémantique automatiquement pour n'importe quel analyseur syntaxique. Notre idée est de partir d'une base de phrases d'exemples associées à leur représentation sémantique, de traiter ces exemples avec l'analyseur de notre choix et de calculer ainsi une grammaire permettant de faire la correspondance

entre les arbres de dépendance en sortie de l'analyseur et des graphes sémantiques. Contrairement aux stratégies d'extraction de grammaires sur corpus annotés par des méthodes statistiques, nous calculons une grammaire formelle purement algébrique. Chaque exemple de notre base est conçu pour obtenir une règle et peut être vu comme une demi-règle contenant la partie sémantique et dont la partie syntaxique est calculée par l'analyseur syntaxique.

Notre représentation sémantique est un graphe de relations prédicat-argument entre les signifiés des unités lexicales et grammaticales d'une phrase (où chaque unité lexicale a été désambiguïsée par rapport à un lexique de référence). Elle est directement inspirée des représentations sémantique et syntaxique profonde de la Théorie Sens-Texte (Mel'čuk 1988a ; Candito & Kahane 1998 ; Kahane 2002). Il s'agit d'une représentation sémantique du contenu linguistique et pas d'une sémantique dénotationnelle comme les représentations sémantiques basées sur la logique. Il n'y a donc pas à proprement parler de calcul de valeurs de vérité associées ; par contre, ce type de représentation permet des calculs de paraphrases (Mel'čuk 1988b ; Milićević 2007) et a été implémenté avec succès pour la génération de textes (Iordanskaja *et al.* 1988 ; Bohnet & Wanner 2001) ou la traduction automatique (Apresjan *et al.* 2003). Des représentations similaires ont été proposées par d'autres auteurs sans référence explicite à la Théorie Sens-Texte. Voir par exemple (Copestake 2009) ou (Bédaride & Gardent 2009).

Nous commencerons par présenter le formalisme utilisé pour l'écriture d'une interface syntaxe-sémantique (section 1), puis les principes généraux du calcul de règles grammaticales ne nécessitant pas de connaissances lexicales (section 2). L'exploitation de ressources lexicales pour la production de nouvelles règles sera esquissée (section 3). Nous terminerons en montrant comment réaliser l'articulation lexique-grammaire par la « soustraction » de règles lexicales à nos règles grammaticales (section 4). Cet article porte essentiellement sur les questions théoriques liées au calcul de l'interface syntaxe-sémantique. Une implémentation non encore évaluée est en cours.

# 1. Écrire une interface syntaxe-sémantique

Pour écrire notre interface syntaxe-sémantique, nous utilisons un formalisme générique, la Grammaire d'Unification Polarisée (GUP) (Kahane, 2004). Ce formalisme permet d'écrire des grammaires de correspondances entre graphes et a déjà été proposé pour l'interface syntaxe-sémantique (Kahane & Lareau, 2005). Ce formalisme permet, à l'image de TAG, de combiner des structures élémentaires afin d'obtenir une structure complète. Les structures que nous souhaitons obtenir sont des couples formés d'un arbre de dépendance syntaxique et d'un graphe sémantique ; nos structures élémentaires sont donc des fragments d'arbre syntaxique associés à des fragments de graphe sémantique. La particularité de ce formalisme est un contrôle rigoureux de ce que chaque règle consomme, à l'aide de polarités associées aux objets manipulés par les règles. Le jeu de polarités le plus simple est constitué de deux polarités, que nous appelons noir (■) et blanc (□). Chaque objet de la structure reçoit une polarité. Sont considérés comme des objets les nœuds (identifiés avec l'élément lexical qu'ils portent), les dépendances et les éléments flexionnels ayant une contribution sémantique (temps verbal, nombre nominal, etc.). Les règles sont combinées par identification des objets dont les étiquettes peuvent s'unifier et les polarités se combiner.

La Figure 1 présente un exemple d'interface syntaxe-sémantique en GUP. En haut à gauche se trouve la phrase *Mary seems to sleep* avec l'analyse syntaxique en dépendance qu'en propose le Stanford Parser. Les dépendances syntaxiques sont orientées vers la gauche (<nsubj) ou vers la droite (xcomp>). Nous ajoutons des polarités blanches sur les dépendances (□) et les mots (□) indiquant que ces objets doivent être consommés par des règles d'interface. Pour les verbes, une deuxième polarité blanche (○) indique que la flexion verbale doit aussi être consommée. En haut à droite se trouve le résultat attendu, c'est-à-dire un graphe sémantique associé à la phrase et polarisé en noir puisque produit par l'interface. Pour assurer cette correspondance, nous utilisons les trois règles qui figurent en dessous. Ce sont des règles lexicales

associées aux lemmes SEEM, SLEEP et MARY. Comme on peut le voir la règle associée à SEEM consomme la totalité des dépendances syntaxiques, mais ne produit qu'une dépendance sémantique. La deuxième dépendance sémantique est produite par la règle de SLEEP. Mais pour que cette règle puisse s'appliquer il est nécessaire que la règle de SEEM restitue une dépendance syntaxique. Notons pour terminer que la règle de SEEM impose à son xcomp> d'être un verbe infinitif (Vinf) et consomme ainsi sa polarité flexionnelle (●).

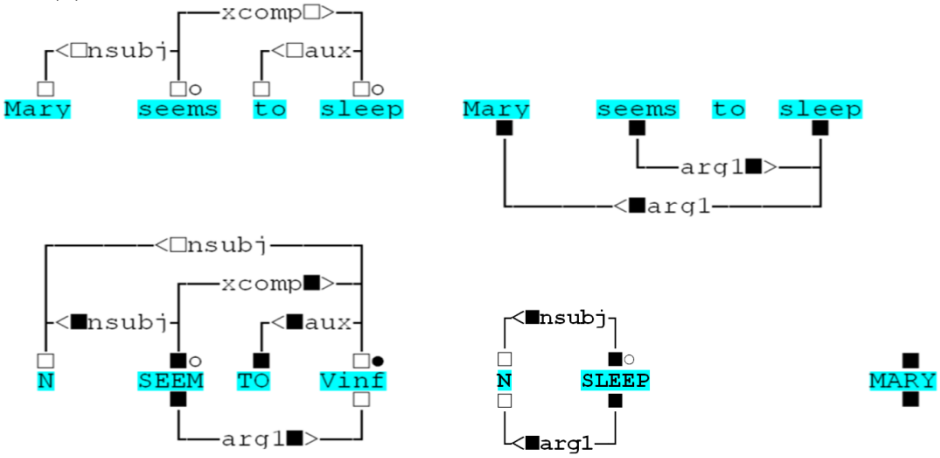


Figure 1. Un exemple d'interface syntaxe-sémantique en GUP

## 2. Calcul d'une règle grammaticale d'interface syntaxe-sémantique

Nous allons calculer la règle pour l'aspect progressif. Celui-ci est exprimé par BE + Ving et nous voulons récupérer au niveau sémantique un attribut [aspect= progressive] sur le verbe. Pour apprendre la règle, nous construirons un exemple de phrase avec un progressif (en l'occurrence *Mary is sleeping*) en indiquant que pour *is* le lemme seul sera consommé (■○) et que pour *sleeping* la flexion seule sera consommée (□●) (voir Figure 3). Autrement dit, nous connaissons déjà la règle et nous pourrions l'écrire à la main. Notre objectif est de l'adapter automatiquement et sans effort aux sorties de n'importe quel analyseur, d'autant que les analyseurs varient non seulement dans les étiquettes qu'ils utilisent, mais aussi dans les structures qu'ils manipulent. Dans le cas du progressif, nous ne savons pas si le sujet sera relié à l'auxiliaire ou au verbe lexical. Nos deux analyseurs de référence, le Stanford Parser et le Link Grammar, font des choix différents. C'est pourquoi nous intégrons le sujet dans la règle et considérons donc une relation sémantique <arg1 correspondante. Nous verrons dans la section 4 comment « retirer » cette information.

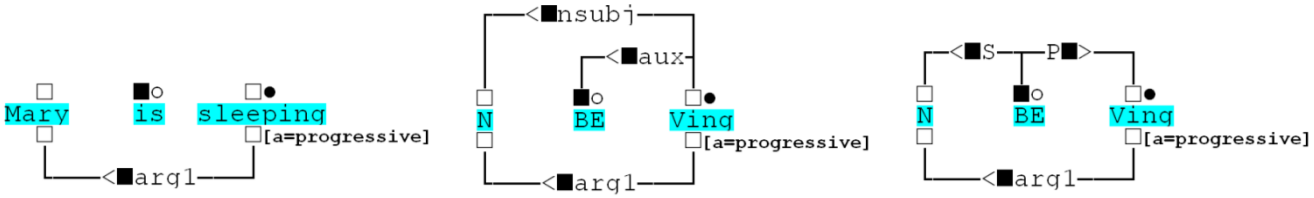


Figure 3. Calcul de la règle du progressif : exemple de départ (à gauche) et règles obtenues pour le Stanford Parser (au centre) et pour Link Grammar Parser (à droite)

## 3. Des règles lexicales pour l'interface syntaxe-sémantique

L'utilisation d'un lexique électronique permet le calcul automatique de règles. Par exemple, une ressource telle que VerbNet (pour l'anglais) ou Dicovalence (pour le français), décrivant des cadres de sous-

catégorisation, peut être mise à profit. L'idée est alors d'analyser les exemples fournis pour en déduire les règles. Par exemple, le cadre give-13.1 de VerbNet est décrit de la façon suivante :

```
<DESCRIPTION descriptionNumber="0.2" primary="NP V NP PP.recipient"/>
<EXAMPLES><EXAMPLE>They lent a bicycle to me.</EXAMPLE></EXAMPLES>
<SYNTAX><NP value="Agent" /> <VERB /> <NP value="Theme" /> <PREP value="to" />
<NP value="Recipient" /></SYNTAX>
```

Cette description peut être utilisée pour créer automatiquement la règle lexicale de la Figure 4, avec le processus décrit dans (Chaumartin, 2005). Première étape : à partir de l'exemple donné par VerbNet et sa description dans VerbNet la demi-règle sémantique est construite. Deuxième étape : l'exemple est analysé par l'analyseur de notre choix (ici le Stanford Parser), ce qui nous fournit une règle lexicale pour l'interface avec les résultats de cet analyseur.

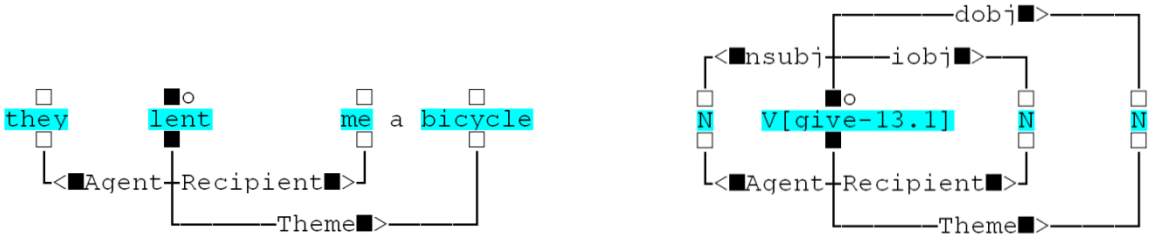


Figure 4. Extraction d'une règle lexicale à partir du cadre give-13.1 de VerbNet : exemple de départ construit à partir de VerbNet à gauche, règle obtenue pour le Stanford Parser à droite

Cette règle est par ailleurs utile pour lever des ambiguïtés lexicales et syntaxiques : en effet, d'une part, la recherche de la règle la plus couvrante dans la forêt d'analyses syntaxiques produite pour une phrase donnée permet d'augmenter le score des analyses où la règle est applicable ; d'autre part, VerbNet précise les sens du verbe compatibles avec le cadre de sous-catégorisation, et impose éventuellement des contraintes de sélection sur ses arguments.

# 4. Articulation lexique-grammaire et soustraction de règles

Considérons une phrase telle que *They were lending me a bicycle*. Nous pouvons y appliquer la règle grammaticale du progressif (calculée en Section 2). Mais cette règle consomme le lien sujet et nous ne pourrons pas ensuite appliquer la règle lexicale du verbe LEND que nous avons créée à partir de VerbNet (Section 3). La solution habituelle à ce problème est celle adoptée par exemple par les grammaires TAG consistant à produire à partir de la diathèse de base toutes les réalisations possibles (Candito 1999) ou (Bédaride & Gardent, 2009) dans un formalisme similaire au notre. Il en résulte un lexique-grammaire assez volumineux en raison de la croissance rapide du nombre de règles en fonction du nombre de phénomènes pris en compte (le lexique inclut en fait la grammaire). Plutôt que d'additionner divers phénomènes au sein d'une même règle, nous proposons au contraire de soustraire aux règles grammaticales la partie lexicale pour permettre à la règle lexicale de se combiner avec les règles grammaticales. Dans le cas du progressif, réalisé par une construction avec un auxiliaire BE + Ving nous nous ramenons au cas d'une forme verbale simple. Pour ramener la construction A au cas d'une construction B, nous proposons simplement de calculer comme précédemment des règles pour A et B, puis de soustraire la règle de B à celle de A. La soustraction est contrôlée par les polarités selon le calcul suivant :

- ■ - ■ = suppression (autrement dit, tout objet manipulé dans A et B est supprimé)
- (□ -) ■ = □ (un objet uniquement manipulé dans B doit être absolument introduit dans la règle A–B pour être consommé ensuite par l'application de B)

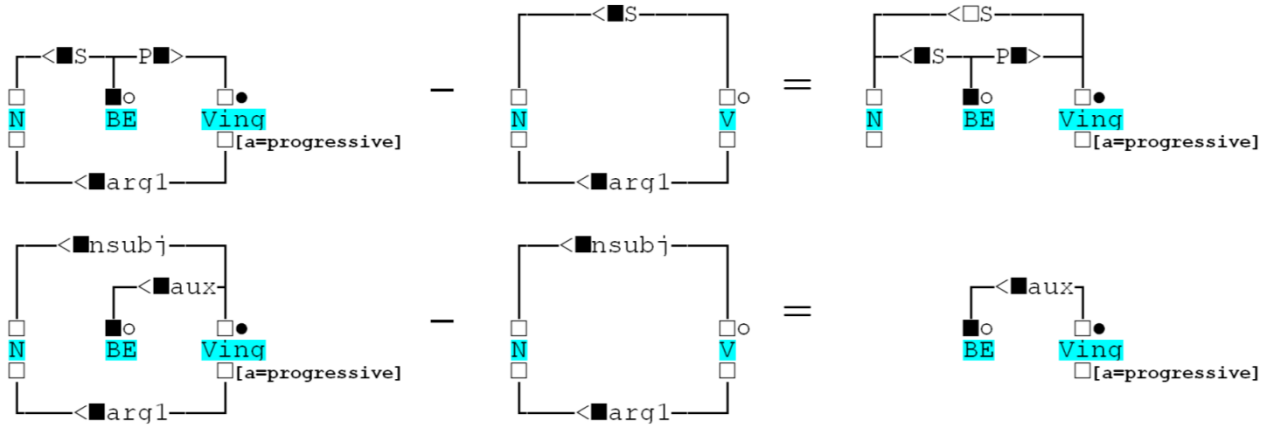


Figure 6. Calcul de la règle pour le progressif

Dans la Figure 6, la première ligne montre les règles obtenues pour le Link Grammar et la deuxième montre celles obtenues pour le Stanford Parser. Rappelons que la base d'exemple contient uniquement la partie inférieure des règles (le graphe sémantique) pour les deux exemples, A = *Mary is sleeping* and B = *Mary sleeps*). La partie supérieure est calculée par l'analyseur, puis la règle B est soustraite de A. Dans le cas du Ling Grammar, le lien <■S de B ne correspond pas au lien <■S de A (ils n'ont pas le même gouverneur) et donne donc un lien <□S dans A–B. Dans le cas du Standford Parser, les liens <■nsubj de A et B se correspondent et s'annulent donc l'un l'autre. Notons que le lien <■S dans la règle du progressif obtenue avec le Ling Grammar n'est pas un problème même pour l'analyse d'une phrase comme *Mary seems to be sleeping* : l'application de la règle pour SEEM (Figure 1) permettra de se ramener à une structure similaire à celle de la phrase *Mary is sleeping*, où la règle du progressif pourra s'appliquer.

Terminons en présentant la règle pour le passif. Comme précédemment, le principe consiste à se ramener à une forme plus simple, c'est-à-dire, dans ce cas, l'actif. La polarisation des nœuds est donnée dans la base d'exemples (les parties inférieures qui s'annulent, puisque le passif n'a pas de contribution sémantique, ne sont pas montrées), la partie supérieure est calculée par l'analyseur (ici le Link Grammar), puis la règle définitive est obtenue par soustraction.

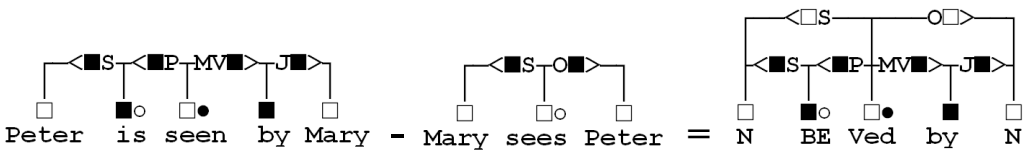


Figure 7. Calcul de la règle pour le passif

## Conclusion

L'idée de développer une interface syntaxe-sémantique entre arbre de dépendance et graphe de relation prédicat-argument est ancienne et remonte au début de la Théorie Sens-Texte dans les années 60 (Mel'čuk 1988, Kahane 2002). L'originalité du présent travail est de proposer une stratégie simple pour construire une telle interface à partir de ressources existantes, analyseurs syntaxiques et lexiques sémantiques. D'une part, notre grammaire s'adapte automatiquement aux sorties de n'importe quel analyseur syntaxique en dépendance. D'autre part, notre grammaire est très modulaire avec une seule règle pour chaque régime d'une unité lexicale et des règles grammaticales séparées pour les différentes constructions et redistributions dont cette unité lexicale peut faire l'objet. Nous pensons que cette approche est pragmatique dans la mesure où elle met en œuvre des ressources de large couverture dans leur état actuel.

Les avantages escomptés de cette approche sont une grande modularité et une facilité de maintenance de l'interface syntaxe-sémantique obtenue, l'indépendance vis-à-vis de tout analyseur syntaxique particulier, et une facilité de prise en compte de nouvelles ressources. Notons que notre grammaire est complètement réversible et peut servir aussi bien pour l'analyse que pour la génération de texte. La mise en œuvre d'une telle grammaire pose évidemment des difficultés qui ne sont pas abordées ici. Notons simplement que le formalisme a déjà fait l'objet d'une implémentation (Lison, 2006) ; nous en développons actuellement une nouvelle implémentation, en utilisant l'outil de réécriture de graphes GrGen et différentes heuristiques afin d'éviter toute explosion combinatoire.

# Références

APRESJAN J. ET AL. (2003). ETAP-3 Linguistic Processor: a Full-Fledged NLP Implementation of the MTT. Actes de *MTT*, Paris, 279-288.

BEDARIDE P., GARDENT C. (2009). Semantic Normalisation: a Framework and an Experiment. Actes d'*IWCS'09: 8th International Conference on Computational Semantics*, Tilburg, Netherland.

BOHNET B., WANNER L. (2001). On using a parallel graph rewriting formalism in generation. Actes du *Workshop on Natural Language Generation, ACL 2001*, Toulouse.

CANDITO M.-H., KAHANE S. (1998). Can the derivation tree represent a semantic graph? An answer in the light of Meaning-Text Theory". Actes de *TAG+4*, Philadelphie, 21-24.

CANDITO, M.-H. (1999). *Organisation modulaire et paramétrable de grammaires électroniques lexicalisées. Application au français et à l'italien*. Thèse de doctorat, Université Paris 7.

CHAUMARTIN F.-R. (2005). Conception et réalisation d'une interface syntaxe / sémantique utilisant des ressources de large couverture en langue anglaise. Actes de *RECITAL 2007*.

CHAUMARTIN F.-R. (2008). ANTELOPE, une plateforme industrielle de traitement linguistique. *TAL* 49.2.

COPESTAKE A. (2009). Slacker semantics : Why superficiality, dependency and avoidance of commitment can be the right way to go. Actes d'*EACL 2009*, Invited Talk, 1–9, Athènes.

IORDANSKAJA L., KITTREDGE R., POLGUÈRE A. (1988). Implementing a Meaning-Text Model for Language Generation. Actes de *COLING 1998*.

KAHANE S. (2002). *Grammaire d'Unification Sens-Texte : Vers un modèle mathématique articulé de la langue naturelle*, Document de synthèse de l'Habilitation à diriger des recherches, Université Paris 7.

KAHANE S. (2004). Grammaires d'unification polarisées. Actes de *TALN 2004*, Fèz.

KAHANE S., LAREAU F. (2005). Meaning-Text Unification Grammar: modularity and polarization. Actes de *MTT 2005*, Moscou.

LISON P. (2006). *Implémentation d'une interface sémantique-syntaxe basée sur des grammaires d'unification polarisées*. Master's thesis, Université Catholique de Louvain, Louvain-la-Neuve, Belgium.

MEL'CUK I. (1988a). *Dependency Syntax: Theory and Practice*, SUNY Press, Albany.

MEL'ČUK I. (1988b). Paraphrase et lexique dans la théorie linguistique Sens-Texte : vingt ans après, *Revue internationale de lexicologie et lexicographie*, Vol. 52/53, pp. 5-50/5-53.

MILIĆEVIĆ J. (2007). *La paraphrase - Modélisation de la paraphrase langagière*. Bern : Peter Lang.

# Ressources citées

Dicouebe (MEL'ČUK, POLGUÈRE) : http://olst.ling.umontreal.ca/dicouebe/

Dicovalence (MERTENS, VAN DEN EYNDE) : http://bach.arts.kuleuven.be/dicovalence/

GrGen : http://www.info.uni-karlsruhe.de/software/grgen/

Link Grammar (SLEATOR, TEMPERLEY, LAFFERTY) : http://www.link.cs.cmu.edu/link/:

Stanford Parser (MANNING, KLEIN) : http://nlp.stanford.edu/software/lex-parser.shtml

VerbNet (KIPPER, SCHULER) : http://verbs.colorado.edu/~mpalmer/projects/verbnet.html