

Grammaire d'Unification Sens-Texte : modularité et polarisation

Sylvain Kahane (1), François Lareau (2)

(1) Modyco, U. Paris 10 / Lattice, U. Paris 7
sk@ccr.jussieu.fr

(2) OLST, U. de Montréal / Lattice, U. Paris 7
francois.lareau@umontreal.ca

Résumé – Abstract

L'objectif de cet article est de présenter l'état actuel du modèle de la Grammaire d'Unification Sens-Texte, notamment depuis que les bases formelles du modèle ont été éclaircies grâce au développement des Grammaires d'Unification Polarisées. L'accent est mis sur l'architecture du modèle et le rôle de la polarisation dans l'articulation des différents modules — l'interface sémantique-syntaxe, l'interface syntaxe-morphotopologie et les grammaires décrivant les différents niveaux de représentation. Nous étudions comment les procédures d'analyse et de génération sont contrôlables par différentes stratégies de neutralisation des différentes polarités.

This article presents the Meaning-Text Unification Grammar's current state, now that its formal foundations have been clarified with the development of Polarized Unification Grammars. Emphasis is put on the model's architecture and the role of polarization in linking the various modules — semantic-syntax interface, syntax-morphotopology interface and the well-formedness grammars of each representation level. We discuss how various polarity neutralization strategies can control different analysis and generation procedures.

Mots Clés – Keywords

Théorie Sens-Texte, interface syntaxe-sémantique, synchronisation, grammaire d'unification polarisée, grammaire de dépendance, grammaire topologique, génération de textes.

Meaning-Text Theory, syntax-semantics interface, synchronization, polarized unification grammar, dependency grammar, topology grammar, text generation.

1 Introduction

L'objectif de cet article est de présenter l'état actuel du modèle de la Grammaire d'Unification Sens-Texte [GUST], notamment depuis que ses bases formelles ont été éclaircies grâce au développement des Grammaires d'Unification Polarisées [GUP]. Le

formalisme des GUP permet de simuler élégamment la plupart des formalismes basés sur la combinaison de structures, notamment les grammaires de réécriture, TAG, LFG ou HPSG (Kahane 2004). Toutefois, GUP a été initialement développé pour donner une assise formelle solide à GUST, mais la formalisation de GUST en GUP n'a encore jamais été présentée.

L'architecture de GUST est basée sur la théorie Sens-Texte (McL'cuk 1997 ; Kahane 2001). Plusieurs niveaux de représentation de la phrase sont considérés : sémantique, syntaxique, morphotopologique et phonologique. Ces niveaux sont bien séparés et chacun possède sa grammaire propre. La structure sémantique est un graphe, la structure syntaxique un arbre de dépendance, la structure morphotopologique un arbre ordonné, et la structure phonologique une chaîne linéaire. Sur ces structures peuvent s'ajouter d'autres structures comme une hiérarchie logique ou une structuration informationnelle au niveau sémantique ou encore une structure prosodique au niveau phonologique, mais nous n'en parlerons pas ici. De plus, ces niveaux sont ordonnés, du plus profond au plus proche de la surface, ce qui nous donne trois modules d'interface pour passer du sens au « texte » : sémantique-syntaxe, syntaxe-morphotopologie et morphotopologie-phonologie.

L'implémentation de GUST nécessite donc un formalisme capable de manipuler différents types de structures (graphe, arbre, arbre ordonné, chaîne) et de pouvoir les apparier. Comme on le verra, le formalisme des GUP est un formalisme mathématique spécialement conçu pour manipuler aussi simplement que possible de telles structures. Par ailleurs, GUP contrôle la saturation des structures qu'il combine par une polarisation de leurs objets : on peut comparer la construction d'une phrase en GUP à la formation d'une molécule en chimie par la neutralisation de la valence des atomes.

Notre implémentation de GUST fait un grand usage de la polarisation en contrôlant la construction des objets (et donc la saturation des structures) par une polarité propre à chaque module, que ce soit une grammaire de bonne formation des structures d'un niveau donné ou une grammaire d'interface entre deux niveaux. De plus, chaque module utilise la polarité des modules adjacents pour son articulation avec eux. L'ordre dans lequel nous neutraliserons ces différentes polarités va décider d'une procédure particulière en analyse ou en génération. Les procédures en largeur résultent de la neutralisation successive de toutes les polarités propres à un module (neutralisation de toute la structure sémantique, puis syntaxique, etc), tandis que les procédures en profondeur résultent d'une neutralisation en cascade de toutes les polarités introduites par un objet, c'est-à-dire que dès qu'un objet est construit à un niveau donné, on cherche à neutraliser les polarités des autres niveaux associées à cet objet plutôt que de construire d'autres objets du même niveau. Malgré une architecture stratifiée (séparation des niveaux et des interfaces), notre modèle peut donc tout à fait gérer une interaction complexe entre les différents modules et simuler une analyse ou une génération incrémentale (tentative de neutralisation de la structure du premier mot de la phrase du niveau phonologique jusqu'au niveau sémantique, puis du deuxième mot, etc).

La définition de GUP sera brièvement rappelée dans la section 2. Dans la section 3, nous proposerons des grammaires de bonne formation pour les différents types de structures que nous considérons dans GUST. Dans la section 4, nous introduirons la notion de grammaire de correspondance et nous montrerons comment GUP permet d'écrire des interfaces.

2 Grammaires d'unification polarisées [GUP]

Les grammaires d'unification polarisées sont des grammaires permettant de générer des ensembles de structures finies. Une structure repose sur des *objets*. Par exemple, pour un graphe (orienté), les objets sont des nœuds et des arcs. Chaque arc est lié à deux nœuds par les fonctions *source* et *cible*. Ce sont ces *fonctions* qui fournissent la structure proprement dite.

Une *structure polarisée* est une structure dont les objets sont polarisés, c'est-à-dire associés par une fonction à une valeur appartenant à un ensemble fini P de *polarités*. L'ensemble P est muni d'une opération commutative et associative notée « \cdot », appelée *produit*. Un sous-ensemble N de P contient les polarités dites *neutres*. Une structure polarisée est dite *neutre* si tous les objets de cette structure sont neutres. Nous utiliserons ici un système de polarités $P = \{\bullet, \circ, \ominus\}$ (que nous appellerons ainsi : \bullet = noire = saturée, \circ = blanche = contexte obligatoire et \ominus = grise = neutre absolu), avec $N = \{\bullet, \ominus\}$, et un produit défini par le tableau suivant (où \perp représente l'impossibilité de se combiner) :

\cdot	\bullet	\circ	\bullet
\bullet	\bullet	\circ	\bullet
\circ	\circ	\circ	\bullet
\bullet	\bullet	\bullet	\perp

Tableau 1 — Le produit des polarités

Les structures peuvent être combinées par *unification*. L'unification de deux structures A et B donne une nouvelle structure $A \oplus B$ obtenue en « collant » ensemble ces structures par l'identification d'une partie des objets de A avec une partie de ceux de B . Lorsque A et B sont unifiées, la polarité d'un objet de $A \oplus B$ obtenu par identification d'un objet de A et d'un objet de B est le produit de leurs polarités. Toutes les fonctions associées aux objets unifiés sont nécessairement identifiées (comme le sont les traits quand on unifie deux structures de traits).

Une *grammaire d'unification polarisée* (GUP) est définie par une famille finie T de types d'objets (avec des fonctions associées aux différents types d'objets), un système (P, \cdot) de polarités, un sous-ensemble N de P de polarités neutres, et un ensemble fini de structures élémentaires polarisées, dont les objets sont décrits par T et dont une peut être marquée comme la structure initiale. Les structures *générées* par la grammaire sont les structures neutres obtenues par combinaison de la structure initiale et/ou d'un nombre fini de structures élémentaires. Rappelons que le formalisme est monotone (avec l'ordre $\bullet < \circ < \ominus$ sur les polarités) et que les structures peuvent être combinées dans n'importe quel ordre.

Nous verrons dans la section suivante deux exemples de GUP qui génèrent respectivement les graphes sémantiques et les arbres syntaxiques. Tous nos exemples seront illustrés par (1) :

(1) *Pierre mange deux pommes.*

3 Grammaires de bonne formation

3.1 Grammaire sémantique

Nos représentations sémantiques sont essentiellement basées sur un graphe de relations prédicat-argument entre sémantèmes (les signifiés des unités lexicales et grammaticales de la phrase). On peut superposer sur ce graphe d'autres informations pour encoder la structure informationnelle ou les relations de portée (Mel'cuk 2001, Kahane 2005). Les nœuds d'une représentation sémantique représentent les sémantèmes et les arcs représentent les relations prédicat-argument. Notre grammaire $G_{\text{sém}}$ est donc une grammaire de graphe dont les objets portent chacun une polarité, notée $p_{\text{sém}}$ dans la suite¹, qui indique quel est le nœud construit par la règle (polarisé en noir) et quels sont les nœuds constituant la valence sémantique du prédicat (polarisés en blanc) (Figure 1).

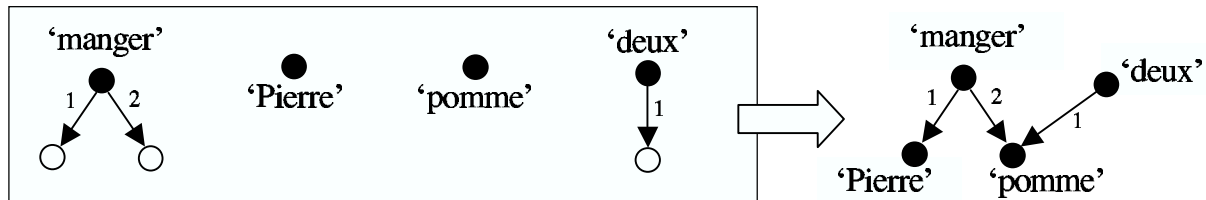


Figure 1 — Un extrait de $G_{\text{sém}}$ générant le graphe sémantique de (1)

3.2 Grammaire syntaxique

Nos représentations syntaxiques sont des arbres de dépendance dont les nœuds sont étiquetés par des représentations lemmatisées de mots. Plus exactement le nœud syntaxique est étiqueté par le nom d'un lexème et des fonctions dites *grammaticales* lient ce nœud à d'autres objets représentant les grammèmes². Les arcs de l'arbre sont étiquetés par des fonctions syntaxiques, (qui varient d'une langue à l'autre). Voir Figure 2 à droite la représentation syntaxique de (1).

Nous présentons notre grammaire syntaxique G_{synt} avec une seule polarité p_{synt} , qui comme précédemment indique quels sont les objets construits par la règle et permet de vérifier que les fonctions associées à chaque objet sont bieninstanciées : la source et la cible pour les arcs, l'étiquette, la partie du discours (*pdd*) pour les nœuds et les fonctions grammaticales éventuelles (*nombre* pour les noms, *mode*, *temps*, *nombre* et *personne* pour les verbes, etc.)³. Une deuxième polarité, $p_{\text{synt-gouv}}$, masquée ici, sert à vérifier que la structure est bien un arbre, c'est-à-dire que chaque nœud est gouverné une et une seule fois, à l'exception du sommet.

¹ Par convention, nous faisons référence aux polarités par le nom de la fonction qui les associe aux objets qui les portent. Ainsi, à tous les objets des règles de $G_{\text{sém}}$ est associée une fonction $p_{\text{sém}}$ qui retourne comme valeur une des polarités de l'ensemble P décrit plus haut.

² Nous indiquons ces fonctions en italiques dans nos figures, suivies du grammème en question. Les grammèmes sont eux-mêmes liés aux lexèmes par des fonctions inverses que nous masquons. Ce double lien rend les deux objets indissociables : l'unification de l'un force l'unification de l'autre.

³ Nous présentons un système grammatical simplifié en masquant par exemple la détermination pour le nom ou la finitude et la voix pour le verbe.

La Figure 2 présente un extrait de G_{synt} permettant de générer l'arbre syntaxique de (1). La toute première règle est la structure initiale, qui doit être utilisée une et une seule fois et qui correspond au sommet de l'arbre (et qui devra recevoir une polarité $p_{\text{synt-gouv}}$ noire indiquant qu'il ne peut être gouverné). Les quatre règles suivantes sont des règles lexicales ; elles indiquent la partie du discours des lexèmes et les grammèmes qui leur sont nécessaires. On notera que ces règles ne contrôlent pas la valence syntaxique, qui sera contrôlée par l'interface sémantique-syntaxe. Les trois règles qui suivent sont des règles sagittales ; elles décrivent différentes relations syntaxiques possibles (et contrôle la structure d'arbre par une polarisation $p_{\text{synt-gouv}}$ noire du nœud dépendant, non indiquée ici). Les autres règles sont des règles grammaticales. Ces règles peuvent dépendre du contexte : ainsi l'indicatif nécessite une relation sujet et vice-versa, la présence d'un numéral impose le pluriel ou celle d'un sujet non pronominal la 3^e personne du verbe. Notons également que l'introduction d'un grammème peut dépendre d'un autre grammème : l'indicatif par exemple exigera un temps, mais pas l'infinitif⁴.

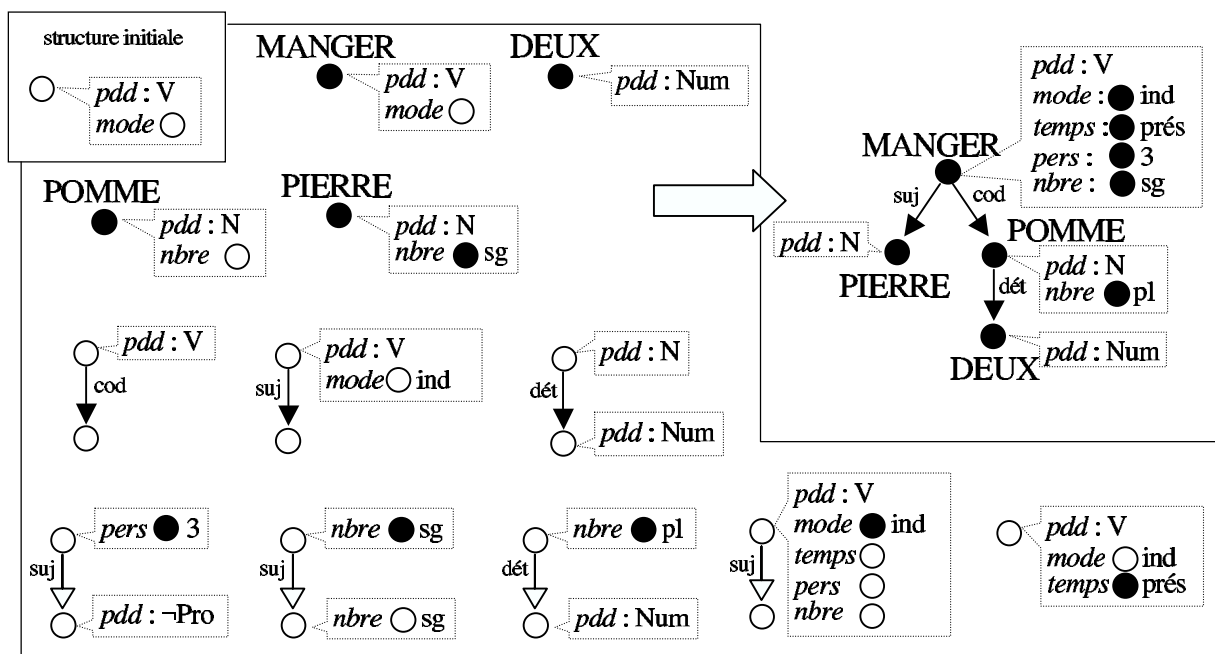


Figure 2 — Un extrait de G_{synt} générant la représentation syntaxique de (1)

Nous présenterons la grammaire morphotopologique en même temps que l'interface syntaxe-morphotopologie, dans la section 4.3. Nous laisserons de côté la phonologie.

4 Interfaces

Nous introduisons d'abord la notion de grammaire de correspondance avant d'introduire les deux interfaces considérées ici : sémantique-syntaxe et syntaxe-morphotopologie.

⁴ Nous considérons l'opposition entre l'infinitif et l'infinitif dit « passé » d'ordre aspectuel.

4.1 Grammaires de correspondance

Une grammaire de correspondance \mathcal{G} est une grammaire qui met en correspondance des structures appartenant à deux ensembles, que nous notons \mathcal{A} et \mathcal{B} . Les règles de \mathcal{G} mettent en correspondance des structures élémentaires composant les éléments de \mathcal{A} et de \mathcal{B} . On peut considérer trois fonctionnements pour \mathcal{G} : équatif, transductif et génératif, selon qu'on fournit en entrée deux, une ou zéro structures (Kahane 2000). Une grammaire équative vérifie que deux structures fournies se correspondent, une grammaire transductive traduit une structure fournie en une autre, tandis qu'une grammaire générative génère un couple de structures en correspondance.

Considérons le fonctionnement équatif de \mathcal{G} , qui est le plus élémentaire : \mathcal{G} partitionne les deux structures en un même nombre de fragments qui se correspondent deux à deux par les règles de la grammaire. En termes de polarités, cela signifie que les deux structures en entrée doivent être déclarées comme des ressources que \mathcal{G} devra toutes « consommer » (assurant ainsi que les deux structures ont été totalement mises en correspondance). Les objets des deux structures recevront donc une polarité $p_{\mathcal{G}}$ blanche que les règles de \mathcal{G} devront neutraliser. Les règles de \mathcal{G} contiendront quant à elles des objets (des niveaux de représentation de \mathcal{A} et de \mathcal{B}) de polarité $p_{\mathcal{G}}$ noire mis en correspondance. Ainsi, en neutralisant les deux structures fournies, \mathcal{G} les met en correspondance.

Voyons maintenant comment la même grammaire \mathcal{G} fonctionne de manière transductive. C'est ce mode de fonctionnement qui modélise les processus linguistiques de synthèse et d'analyse. On fournit à \mathcal{G} une structure A de \mathcal{A} dont tous les objets portent une polarité $p_{\mathcal{G}}$ blanche. On déclenche alors un jeu de règles afin de neutraliser A , ce qui nous construit une structure B synchronisée avec A . Une grammaire de bonne formation des structures de \mathcal{B} doit maintenant vérifier que B appartient bien à \mathcal{B} . La structure B , tout en étant neutre pour \mathcal{G} (tous ses objets portent une polarité $p_{\mathcal{G}}$ noire), doit donc déclencher la grammaire de \mathcal{B} . Elle doit pour cela être entièrement blanche en polarité $p_{\mathcal{B}}$. En somme, chaque module, qu'il soit une grammaire de bonne formation ou une grammaire d'interface, possède une polarité propre contrôlant la construction des objets par cette grammaire, mais pour assurer l'appel des modules adjacents, les objets des règles de \mathcal{A} et de \mathcal{B} reçoivent, en plus de leurs polarités respectives $p_{\mathcal{A}}$ et $p_{\mathcal{B}}$, une polarité $p_{\mathcal{G}}$ blanche, tandis qu'il faut ajouter aux règles de \mathcal{G} une polarité blanche $p_{\mathcal{A}}$ pour les éléments du niveau de représentation de \mathcal{A} et $p_{\mathcal{B}}$ pour les éléments du niveau de \mathcal{B} . Un objet construit par \mathcal{A} aura donc une double polarisation $p_{\mathcal{A}}-p_{\mathcal{G}}$ (\bullet, \circ) , tandis que l'élément correspondant construit par \mathcal{G} aura une double polarisation $p_{\mathcal{A}}-p_{\mathcal{G}}$ (\circ, \bullet) . Ceci nous donne un système à quatre polarités $\{(\circ, \circ), (\circ, \bullet), (\bullet, \bullet), (\bullet, \circ)\}$ équivalent au système $\{\circ, -, +, \bullet\}$ de Bonfante *et al.* 2004 et Kahane 2004. Ces couples de polarités sont les *polarités d'articulation*.

4.2 Interface sémantique-syntaxe

L'interface sémantique-syntaxe $I_{\text{sém-synt}}$ est une grammaire de correspondance entre l'ensemble des graphes sémantiques décrit par $\mathcal{G}_{\text{sém}}$ et l'ensemble des arbres syntaxiques décrit par $\mathcal{G}_{\text{synt}}$.

Nous allons illustrer la présentation de $I_{\text{sém-synt}}$ par son fonctionnement transductif en synthèse. On fournit à la grammaire un graphe sémantique comme celui de la Figure 1 à droite, à la différence que chacun des objets reçoit maintenant une double polarisation $p_{\text{sém}}-p_{\text{sém-synt}}$ (\bullet, \circ). Les objets des règles de $I_{\text{sém-synt}}$ portent tous une polarité $p_{\text{sém-synt}}$ (blanche ou noire, selon qu'ils sont ou non construits par la règle en question), mais ceux du niveau sémantique (resp. syntaxique) auront en plus une polarité $p_{\text{sém}}$ (resp. p_{synt}) blanche. La Figure 3 présente un extrait de la grammaire d'interface qui traite le graphe sémantique de (1). Seule la polarité $p_{\text{sém-synt}}$ est représentée.

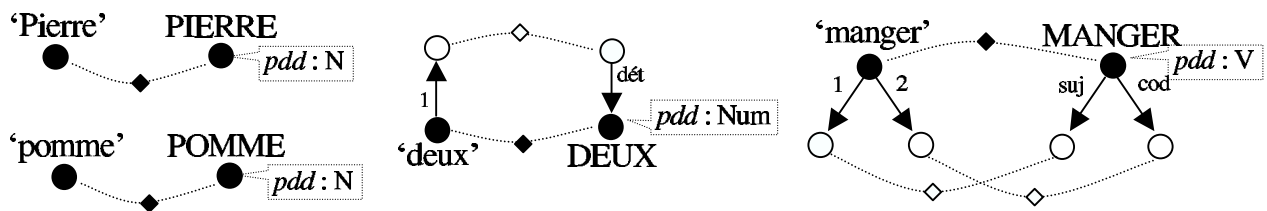


Figure 3 — Un extrait de la grammaire d'interface $I_{\text{sém-synt}}$

Dans la Figure 4, nous donnons trois structures : le graphe sémantique de (1) avec la double polarisation $p_{\text{sém}}-p_{\text{sém-synt}}$, le résultat après neutralisation par $I_{\text{sém-synt}}$ de toutes les polarités qui pouvaient l'être (c'est-à-dire $p_{\text{sém-synt}}$ et $p_{\text{sém}}$, mais pas p_{synt}), puis le résultat après application de G_{synt} sur l'arbre syntaxique construit par $I_{\text{sém-synt}}$. Les polarités qui servent à l'articulation avec le module adjacent sont indiquées en décalé. A noter que les grammèmes d'accord (comme le *nbre* et *pers* pour le verbe) n'ont pas de contrepartie sémantique directe et ne doivent donc pas porter de polarité $p_{\text{sém-synt}}$.

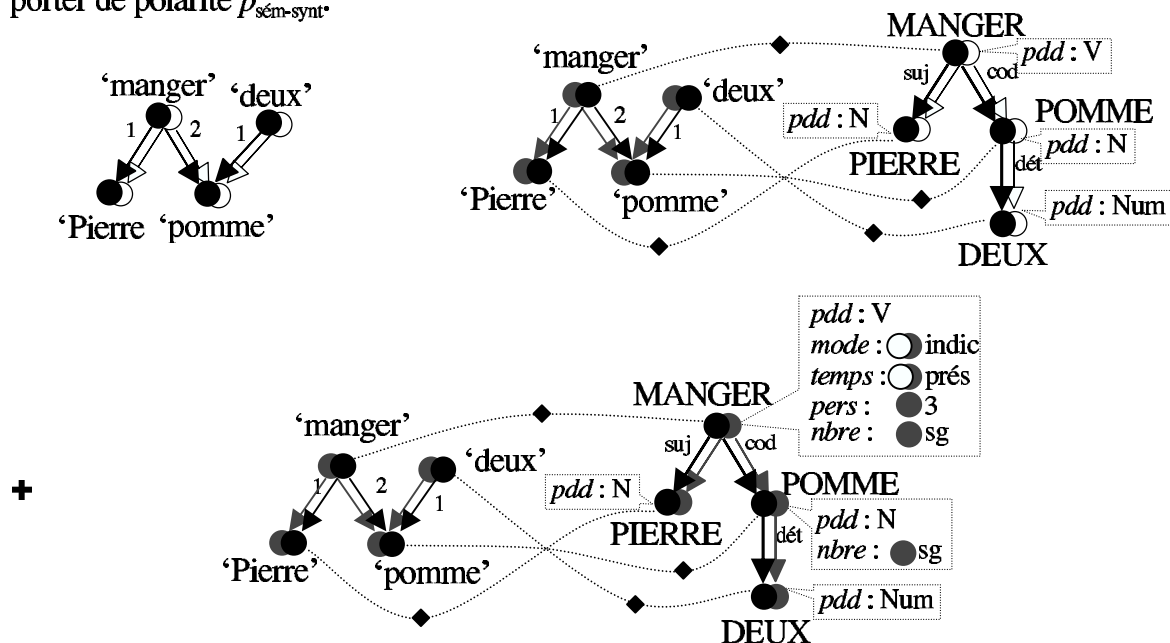


Figure 4 — Le graphe sémantique de (1) avant et après l'application de $I_{\text{sém-synt}}$ et G_{synt}

De même que $G_{\text{sém}}$, la grammaire syntaxique G_{synt} doit être enrichie pour assurer l'articulation avec à la fois l'interface sémantique-syntaxe et l'interface syntaxe-morphotopologie. Comme on peut le voir dans de bas de la Figure 4, l'application de G_{synt} sur la sortie de $I_{\text{sém-synt}}$ permet de neutraliser toutes les polarités p_{synt} des objet syntaxiques construits par $I_{\text{sém-synt}}$. Cependant, il n'est pas possible d'introduire le lexème MANGER sans lui attribuer un mode (cf. Figure 2)

et en lui attribuant le mode indicatif, par exemple, on force l'utilisation d'un grammème de temps et de grammèmes d'accord en personne et en nombre. La grammaire G_{synt} , en neutralisant les polarités p_{synt} de la sortie de $J_{\text{sém-synt}}$, introduit donc des grammèmes qui ne figuraient pas dans cette sortie et qui pour certains (pas les grammèmes d'accord) auront une polarité $p_{\text{sém-synt}}$ blanche (Figure 4 en bas). Cette polarité blanche sur les grammèmes de mode et de temps ainsi introduits devra donc être neutralisée par des règles de $J_{\text{sém-synt}}$. Or, rien dans la structure sémantique de départ ne peut correspondre à ces grammèmes. Qu'à cela ne tienne ! Le jeu des polarités de GUST permet de modifier en cours de route la représentation donnée en entrée afin de satisfaire des contraintes provenant d'autres niveaux de représentation. Supposons que notre modèle comprenne les deux règles données en Figure 5. La première est une règle de $G_{\text{sém}}$ qui donne une représentation d'un des sens du temps présent. La seconde est une règle de $J_{\text{sém-synt}}$ qui fait correspondre à ce sens le grammème de temps présent. Ces deux règles peuvent s'appliquer dans le sens inverse de la synthèse qui est en cours pour venir modifier la structure de départ en lui ajoutant le sens du présent (nous laissons de côté la question du mode pour des raisons de clarté). On obtient alors la structure sémantique à droite.

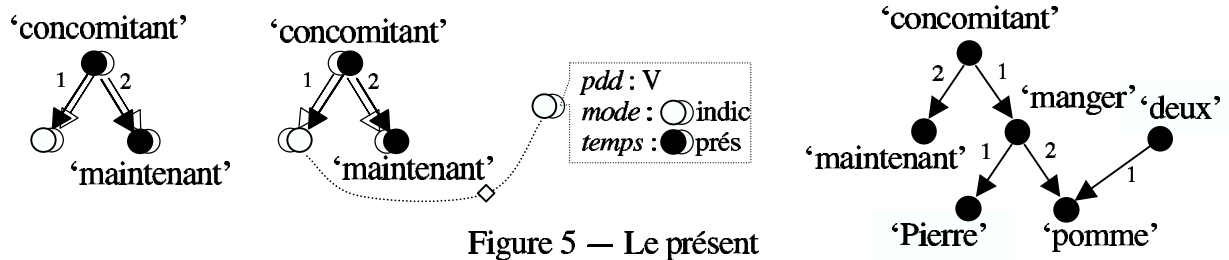


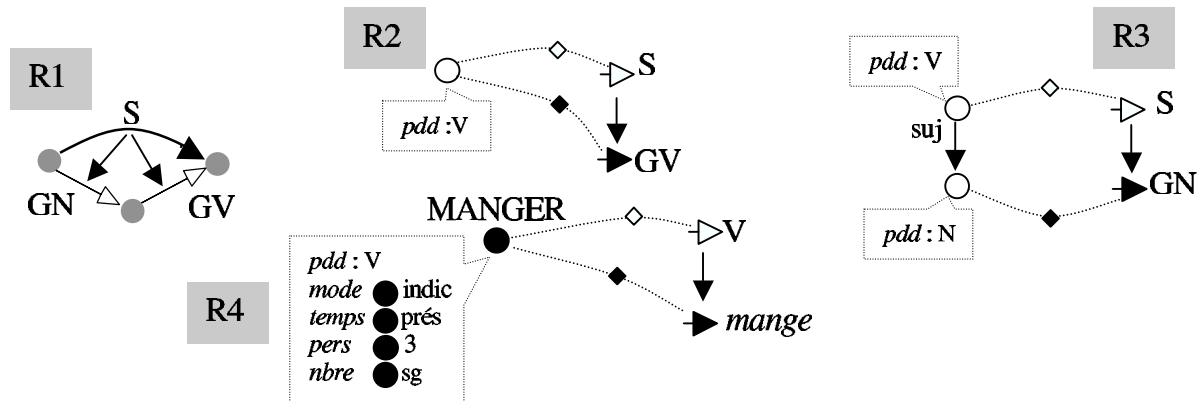
Figure 5 — Le présent

Bien entendu, il existe d'autres règles qui peuvent saturer les besoins en mode et en temps, et on obtiendra autant de structures sémantiques modifiées qu'il y a de façons de satisfaire les contraintes syntaxiques. Dans le cadre d'un système de génération de texte, il faut alors que le programme choisisse, parmi les graphes sémantiques proposés, celui qui convient le mieux selon les connaissances du monde auxquelles il a accès. Dans le cas qui nous préoccupe, le système aura ainsi à répondre à la question « Quand Pierre mange-t-il deux pommes : est-ce dans le passé, dans le présent ou dans le futur ? ». Ce mécanisme modélise bien le fait que les sens grammaticaux ne sont pas nécessairement exprimés parce qu'ils répondent à un besoin communicatif de la part du locuteur, mais plutôt parce qu'ils sont imposés par la langue (voir à ce sujet Polguère 1998). Ce type de mécanisme, en plus d'être plausible d'un point de vue cognitif, pourrait s'avérer particulièrement utile dans le cadre de la traduction automatique, où la représentation sémantique obtenue par analyse dans la langue source n'est pas forcément exprimable dans la langue cible. Il peut également opérer en situation d'analyse. C'est alors la chaîne sonore qui est donnée en entrée et qui peut être modifiée si elle n'est pas tout à fait bien formée, ce qui permettrait de contourner certains problèmes liés au bruit ou aux erreurs de performance. Ces hypothèses n'ont toutefois pas encore été validées par expérimentation.

4.3 Interface syntaxe-morphotopologie

Nous allons simplifier la présentation de l'interface syntaxe-morphotopologie au maximum en ne traitant pas réellement la morphologie (nous associons directement un lexème et les grammèmes qui lui correspondent à une forme fléchie) et en adoptant une structure topologique conventionnelle du type de celle des grammaires syntagmatiques (*cf.* Gerdes &

Kahane 2004 pour une étude de la topologie du français et Gerdes & Kahane 2001 pour une modélisation formelle). La grammaire de bonne formation topologique est donc équivalente à une grammaire de réécriture hors-contexte (cf. Kahane 2004 pour la simulation en GUP des grammaires de réécriture) : elle construit un arbre de constituants topologiques et son principal objectif est d'indiquer dans quel ordre se trouvent les constituants frères (cf. règle R1 équivalente à $S \rightarrow GN\ GV$)⁵. L'interface syntaxe-morphotopologie contient trois types de règles : des règles pour le sous-constituant tête (cf. règle R2 disant que GV est la tête de S et que donc GV et S sont synchronisés avec le même nœud syntaxique), des règles pour le placement d'un dépendant (cf. règle R3 disant que le sujet correspond au GN sous S) et des règles de « morphologie » (cf. règle R4 disant que *mange* \Leftrightarrow MANGER_{ind, prés, 3, sg}). Cette interface est assez proche d'une grammaire LFG (Bresnan 1999), assurant comme elle la synchronisation d'une structure de constituants avec une structure de dépendance, mais la polarisation permet d'explicitier ce que chaque règle construit réellement et de découper davantage l'information (par exemple, nos trois premières règles correspondent à une unique règle LFG : $S \rightarrow GN[\downarrow=\uparrow\text{Subj}]\ GV[\downarrow=\uparrow]$). Dans la Figure 6, nous n'indiquons que la polarité propre à chaque module (p_{topo} dans R1 et $p_{\text{synt-topo}}$ dans les autres).



5 Conclusion

Notre article porte essentiellement sur l'architecture d'un modèle linguistique. Nous avons proposé un modèle modulaire (3 grammaires de bonne formation et 2 grammaires d'interface dans la version simplifiée présentée ici) permettant de gérer différents niveaux d'organisation et différents types de structures (graphe, arbre, arbre ordonné) et pourtant nous utilisons un unique formalisme. Cela nous permet de combiner les règles dans n'importe quel ordre et permet aux différents modules d'interagir constamment. De plus, la saturation des structures ainsi que l'interaction des modules sont soigneusement contrôlées grâce à la polarisation (généralement multiple) de tous les objets. Chaque polarité ayant un rôle bien déterminé, nous pouvons diriger l'application du modèle. On peut ainsi choisir entre une procédure en largeur et une procédure en profondeur : en privilégiant la neutralisation de tous les objets portant une

⁵ Les nœuds gris dans la règle R1 servent à indiquer le début et la fin des constituants. Ils reçoivent une polarité p_{topo} grise, c'est-à-dire qu'ils sont absolument neutres et peuvent se combiner librement à chaque fois qu'une unification d'arc l'oblige. Ils n'auront pas de polarité $p_{\text{synt-topo}}$, puisqu'ils n'ont pas de correspondant au niveau syntaxique. A noter que, comme pour le niveau syntaxique, une deuxième polarité, non considérée ici, est nécessaire pour assurer la structure d'arbre et vérifier que chaque « arc » est gouverné.

polarité propre à un module, les modules s'appliqueront les uns à la suite des autres, tandis qu'en privilégiant la neutralisation de toutes les polarités associées à un objet, chaque élément de la structure d'entrée sera traité du sens au texte (ou vice-versa). Il est important de noter que la procédure en profondeur n'est possible que parce qu'un même formalisme a été utilisé pour l'ensemble des modules. Nous avons également vu que, grâce à l'interaction étroite des différents modules, une GUST peut accepter en entrée une représentation sémantique sous-spécifiée où tous les sens flexionnels sont absents, et générer quand même des phrases bien formées en ajoutant à la représentation initiale les sens flexionnels qu'impose la langue. Enfin, comme l'ont montré Bonfante *et al.* (2004), la polarisation permet un filtrage efficace des règles réduisant les calculs sans issue.

Le développement d'un modèle du français est en cours. Le noyau central de la topologie du français est décrit par Gerdes & Kahane (2004), la syntaxe des temps du français est décrite par Lareau (2004), d'autres points de l'interface sémantique-syntaxe (les distorsions entre syntaxe et sémantique et les questions de portée des quantificateurs) sont étudiés par Kahane (2003, 2005). L'implémentation de GUST dans le formalisme GUP est également en cours. Mais davantage que la couverture, l'accent est mis sur la propreté du traitement théorique et l'adéquation de la formalisation.

Références

- BRESNAN J. (1999), *Lexical-Functional Syntax*, Blackwell.
- BONFANTE G., GUILLAUME B. & PERRIER G. (2004), Polarization and abstraction of grammatical formalisms as methods for lexical disambiguation, *Actes CoLing*, Genève, 303-309.
- GERDES K. & KAHANE S. (2001), Word order in German: A formal dependency grammar using a topological hierarchy, *Actes ACL*, Toulouse, 220-227.
- GERDES K. & KAHANE S. (2004), L'amas verbal au cœur d'une modélisation topologique du français, *Actes Journées de la syntaxe – Ordre des mots dans la phrase française, positions et topologie*, Bordeaux, 8 p.
- KAHANE S. (2000), Des grammaires formelles pour définir une correspondance, *Actes TALN*, Lausanne, 197-206.
- KAHANE S. (2001), Grammaires de dépendance formelles et théorie Sens-Texte, Tutoriel, *Actes TALN*, vol. 2, 17-76.
- KAHANE S. (2004), Grammaires d'unification polarisées, *Actes TALN*, Fès, 233-242.
- KAHANE S. (2005), Structure des représentations logiques, polarisation et sous-spécification, *Actes TALN*, Dourdan, 10 p.
- LAREAU F. (2004), *Vers un modèle formel de la conjugaison française dans le cadre des grammaires d'unification Sens-Texte polarisées*, Document pour l'examen général de synthèse, Montréal, Université de Montréal.
- MEL'CUK I. (1997), *Vers une linguistique sens-texte : leçon inaugurale*, Paris, Collège de France.
- MEL'CUK I. (2001), *Communicative Organisation of Natural Language*, Benjamins.
- NASR A. (1995), A formalism and a parser for lexicalised dependency grammars, *4th Int. Workshop on Parsing Technologies*, State University of New York Press.
- POLGUERE A. (1998), Pour un modèle stratifié de la lexicalisation en génération de texte, *TAL*, 39:2, 57-76.
- SHIEBER S. M. & SCHABES Y. (1990), Synchronous tree-adjointing grammars, *Proceedings of the 13th Int. Conference on Computational Linguistics*, vol. 3, 253-258, Helsinki, Finland.