



**HAL**  
open science

## La sémantique dans les grammaires d'interaction

Guy Perrier

► **To cite this version:**

Guy Perrier. La sémantique dans les grammaires d'interaction. 11ième Conférence annuelle sur le Traitement Automatique des Langues Naturelles - TALN'2004, 2004, Fès, Maroc, France. 10 p. inria-00099887

**HAL Id: inria-00099887**

**<https://inria.hal.science/inria-00099887>**

Submitted on 26 Sep 2006

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

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

## La sémantique dans les grammaires d'interaction

Guy Perrier  
LORIA - Université Nancy 2  
BP 239 - 54506 Vandœuvre-lès-Nancy cedex  
perrier@loria.fr

### Résumé - Abstract

Nous proposons d'intégrer la sémantique dans les grammaires d'interaction, formalisme qui a été conçu pour représenter la syntaxe des langues. Pour cela, nous ajoutons au formalisme un niveau supplémentaire qui s'appuie sur les mêmes principes fondamentaux que le niveau syntaxique : contrôle de la composition par un système de polarités et utilisation de la notion de description de structure pour exprimer la sous-spécification. A la différence du niveau syntaxique, les structures sont des graphes acycliques orientés et non des arbres localement ordonnés. L'interface entre les deux niveaux est assurée de façon souple par une fonction de liage qui associe à tout nœud syntaxique au plus un nœud sémantique.

We propose an integration of semantics into Interaction Grammars, a formalism that was designed for representing the syntax of natural languages. It consists in the addition of a new level to the formalism and this level is based on the same fundamental principles as the syntactical level: the control of composition with a system of polarities and the use of the notion of structure description for expressing underspecification. Unlike the syntactical level, structures are directed acyclic graphs and not locally ordered trees. The interface between the two levels is performed in a flexible way by a linking function which maps every syntactical node to at most one semantical node.

### Mots-clefs – Keywords

formalisme grammatical, interface syntaxe-sémantique, sous-spécification, polarités.  
grammatical formalism, syntax-semantics interface, underspecification, polarities.

### Introduction

Initialement, les grammaires d'interaction (IG en abrégé pour Interaction Grammars) ont été conçues pour modéliser la syntaxe des langues (Perrier, 2002) à partir de deux traditions différentes des grammaires formelles : les grammaires catégorielles (CG en abrégé pour Categorical Grammars) et les grammaires d'arbres adjoints (TAG).

Des CG, les IG retiennent l'idée que les syntagmes sont vus comme des ressources consommables et qu'il y a une dualité entre celles-ci qui s'exprime dans le mécanisme de composition syntaxique : certaines ressources, munies de polarités négatives, sont attendues alors que d'autres, munies de polarités positives, sont disponibles si bien que les premières vont chercher à rencontrer les secondes et c'est ce principe de *neutralisation des polarités opposées* qui va guider la composition syntaxique.

La plupart des formalismes linguistiques tiennent compte de la sensibilité aux ressources des langues mais en général ils le font de façon externe à l'aide de principes de bonne formation qui sont vérifiés après coup sur les structures engendrées. Seules les CG en font un principe incorporé dans le formalisme lui-même, qui s'applique dans le processus de composition lui-même et permet de le contrôler. Les IG peuvent être vues comme un raffinement des CG en ce sens que la notion de polarité est descendue du niveau des syntagmes au niveau des traits grammaticaux utilisés pour les décrire (parties du discours, fonctions syntaxiques ...). Le principe de composition syntaxique reste fondamentalement le même, prenant seulement la forme de la *neutralisation de traits duaux*.

En même temps qu'un raffinement des CG, les IG introduisent un assouplissement considérable dans le formalisme en ayant recours à la notion de *description d'arbre*. Utilisée par Vijay-Shanker (1992) pour exprimer l'adjonction des TAG sous forme d'une opération monotone, elle s'inscrit dans une nouvelle approche de la formalisation des langues qui considère les grammaires comme des systèmes de contraintes plutôt que comme des systèmes dérivationnels. Les énoncés des langues sont alors vus comme des instanciations de ces systèmes de contraintes dont la résolution produit les analyses. Une bonne illustration de cette approche nous est donnée par les grammaires de propriétés de Blache (2001). L'intérêt d'une telle approche est qu'elle permet une grande souplesse de composition et une expression économique de la sous-spécification. La notion de description d'arbre n'est que l'application de cette approche à la représentation des structures syntaxiques à l'aide de contraintes de domination et de précedence portant sur des nœuds représentant des syntagmes. L'originalité des IG est d'étiqueter ces nœuds par des structures de traits polarisés et d'ajouter aux contraintes précédentes des contraintes de neutralité sur les polarités.

Dans les langues, la syntaxe n'est qu'un moyen pour accéder à la sémantique. Un formalisme ne peut donc se préoccuper de la première en ignorant la seconde. Il ne peut considérer non plus que la sémantique est une simple projection de la syntaxe et il est plus conforme à la réalité de voir leur rapports en termes d'interactions qu'en termes de dépendance passive de la première relativement à la seconde.

Malheureusement, parmi les formalismes distinguant le niveau sémantique du niveau syntaxique, beaucoup ne présentent le premier que comme une déduction passive du second. Cela a comme conséquence soit de ne pas pouvoir rendre compte de phénomènes sémantiques lorsqu'ils ne sont pas le résultat mécanique de phénomènes syntaxiques, soit de compliquer artificiellement la syntaxe pour en rendre compte et d'une certaine façon d'inclure la sémantique de manière détournée dans la syntaxe.

Une bonne illustration de notre propos nous est fournie par les CG dont sont justement issues les IG. Classiquement, la sémantique y est intégrée sous forme de la sémantique de Montague (Montague, 1970). Le cadre utilisé est celui d'une logique d'ordre supérieur exprimée à l'aide d'un lambda-calcul typé (Carpenter, 1998). Chaque phrase est représentée par une formule logique qui est une fonction de la représentation sémantique de chacun des mots de la phrase. Cette fonction est représentée sous forme d'un lambda-terme qui est une stricte projection de la structure syntaxique de la phrase. Cette projection consiste à oublier l'ordre des mots et à ne conserver que les dépendances syntaxiques.

Prenons par exemple la phrase *Jean aime Marie*. Sa structure syntaxique dans les CG s'exprime sous forme de l'application de la fonction  $\lambda x_{GN} \lambda y_{GN} . (y * aime * x)_P$  à deux arguments de type *groupe nominal (GN)* *Marie* et *Jean* qui vont instancier respectivement  $x$  et  $y$ . L'opérateur "\*" est l'opérateur de concaténation et la valeur retournée par la fonction est de type *phrase (P)* : c'est la phrase *Jean aime Marie*. La représentation sémantique s'obtient par projection de chaque type syntaxique en un type sémantique et par oubli de l'ordre des mots dans l'expression du résultat de la fonction précédente. Si on considère que les types *GN* et *P* se projettent sémantiquement selon les types *entité (e)* et *booléen (b)*, la représentation sémantique de la phrase s'obtient comme application de la fonction  $\lambda u_e \lambda v_e . aime(v, u)_b$  aux entités correspondant à

*Marie et Jean* qui vont instancier respectivement  $u$  et  $v$ .

Maintenant, voyons comment analyser la phrase *tous aiment quelqu'un*. Syntactiquement, elle se présente comme *Jean aime Marie* mais si nous reprenons la même analyse syntaxique que celle qui vient d'être effectuée, par projection, on obtiendra la même représentation sémantique et on aura complètement ignoré la présence et le rôle des deux quantificateurs dans la phrase. Compte tenu des relations de portée entre les deux quantificateurs présents, il y a deux représentations sémantiques possibles de la phrase d'où nécessité de deux structures syntaxiques correspondantes : dans l'une, *tous* est représenté par une fonction qui s'applique à *aiment quelqu'un* et dans l'autre, c'est *quelqu'un* qui est représenté par une fonction s'appliquant à *tous aiment*. Dès que l'on va augmenter le nombre quantificateurs, l'analyse syntaxique va vite devenir très compliquée. On peut percevoir ainsi les limites d'une telle conception de l'interface syntaxe-sémantique.

L'objet du propos qui va suivre est d'ajouter un niveau sémantique aux IG de façon à avoir un formalisme linguistique qui couvre à la fois la syntaxe et la sémantique des langues. On vient d'entrevoir les limites d'une interface syntaxe-sémantique trop rigide comme celle qui existe pour les CG. Les IG évitent cet écueil en gardant un niveau syntaxique autonome. Le mécanisme de liage entre les deux niveaux est aussi peu contraignant que possible puisqu'il consiste en une simple fonction partielle qui associe à un syntagme au plus un objet sémantique. On ne demande aucune propriété particulière à cette fonction. La représentation sémantique, quant à elle, est fondée sur les mêmes notions de *polarité* et de *description sous-spécifiée* que la représentation syntaxique avec le même mécanisme de composition guidé par le principe de *neutralisation des polarités*. Il y a cependant une différence : au niveau syntaxique, nous manipulons des arbres localement ordonnés et donc des descriptions d'arbres localement ordonnés mais, au niveau sémantique, nous cherchons à exprimer des dépendances sémantiques entre entités sous forme de relations prédicat-arguments et nous utilisons pour cela des graphes acycliques orientés (DAG en abrégé pour Directed Acyclic Graph) donc des descriptions de DAG. Dans les sections 1 et 2, nous présenterons séparément les niveaux syntaxique et sémantique des IG puis dans la section 3, nous montrerons comment ils interagissent dans le processus d'analyse. Nous terminerons dans la section 4 par une comparaison avec d'autres formalismes.

## 1 Le niveau syntaxique et les descriptions d'arbres polarisées

Une description d'arbre syntaxique est un ensemble de nœuds et de relations de domination immédiate, de domination large et de précédence entre ces nœuds. Les nœuds représentent des syntagmes et les relations les dépendances entre ces syntagmes. Les propriétés morpho-syntaxiques de ces syntagmes sont exprimées par des structures de traits attachées aux nœuds. La figure 1, dans sa partie inférieure, montre un exemple simplifié de description d'arbre,  $D_{syn}$ , qui est associée par un lexique à la phrase *tous aiment quelqu'un*. Les nœuds y sont représentés par des rectangles avec pour entête leur nom et pour corps la structure de traits attachée. Les relations de domination immédiate sont représentées par des traits continus, celles de précédence par des flèches et les relations de domination large par des traits discontinus. Ces dernières, qui vont permettre d'exprimer aussi bien les possibilités d'appliquer des modificateurs à certains syntagmes et les dépendances syntaxiques non bornées, peuvent être contraintes par des structures de traits. Ainsi, la relation de domination  $v-max \overset{*}{>} v-min$  est contrainte par le trait  $cat = v$ . Le nœud  $v-min$  représente l'ancre<sup>1</sup> de la description, c'est-à-dire la position du verbe associé *aiment* dans la description. A ce verbe peuvent éventuellement s'adjoindre une négation, des clitiques ou des adverbes pour former sa projection maximale  $v-max$ . La contrainte  $cat = v$  sur la relation de domination  $v-max \overset{*}{>} v-min$  exprime exactement cela puisque elle signifie que tout nœud qui domine strictement  $v-min$  et qui est dominé au sens large par  $v-max$  doit être por-

<sup>1</sup>Les ancres sont représentées par des rectangles colorés.

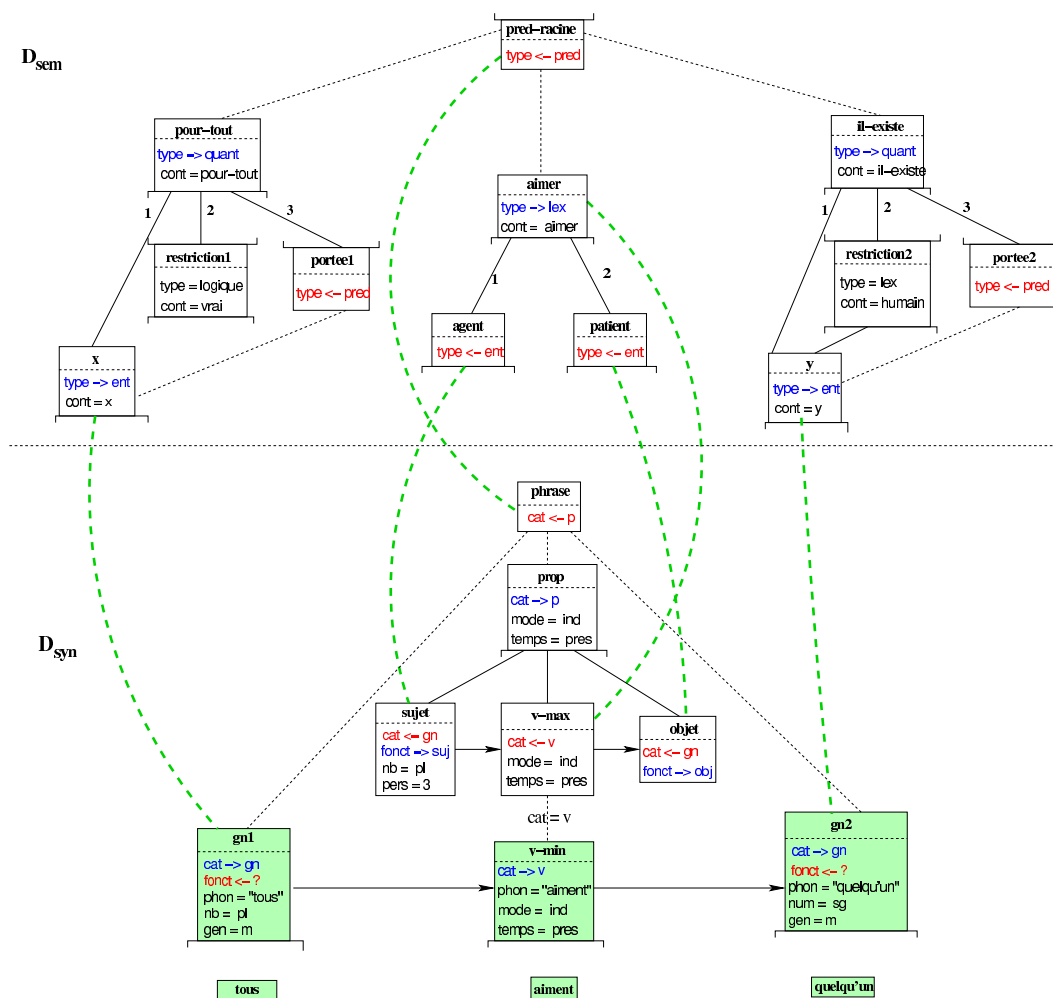


Figure 1: Description syntaxico-sémantique associée à la phrase *tous aiment quelqu'un* par le lexique d'une IG.

teur d'un trait pouvant s'unifier avec le trait  $cat=v$ . Lorsque le rectangle représentant un nœud est muni à son pied d'un crochet tourné vers le bas, cela signifie que l'ensemble des fils du nœud a sa cardinalité fixée. Par exemple, le crochet associé au nœud *prop* représente la relation  $prop \rightarrow \{sujet, v-max, objet\}$  qui signifie que *prop* a exactement trois fils distincts : *sujet*, *v-max*, *objet*.

L'originalité des IG est que les traits morpho-syntaxiques associés aux syntagmes sont polarisés. Habituellement, un trait est associé à une valeur dans un couple (*trait, valeur*). Dans les IG, un trait est associé à une polarité et à une valeur dans un triplet (*trait, polarité, valeur*). Une polarité peut être  $\leftarrow$ ,  $\rightarrow$ ,  $=$  et  $\leftrightarrow$  pour dire qu'un trait est *négatif*, *positif*, *neutre ordinaire* ou *saturé*. Un trait négatif  $f \leftarrow v$  représente une ressource attendue, un trait positif  $f \rightarrow v$  une ressource disponible et un trait neutre ordinaire  $f = v$  une propriété qui ne se comporte pas comme une ressource consommable. Un trait saturé  $f \leftrightarrow v$  est un trait neutre particulier qui provient de la neutralisation d'un trait négatif par un trait positif. Contrairement à un trait neutre ordinaire, il ne peut pas s'unifier avec un trait positif ou négatif.

Une description d'arbre peut être considérée comme un ensemble de contraintes définissant un ensemble d'arbres syntaxiques et chacun de ces arbres peut être vu comme un modèle de la description correspondante. Les polarités définissent des contraintes spécifiques de neutralité sur les modèles : dans tout modèle valide, tout trait positif doit avoir été neutralisé exactement par un trait négatif correspondant et réciproquement. Pour une définition formelle de cette

notion de modèle, le lecteur pourra se reporter à (Perrier, 2002).

S'il est nécessaire de définir rigoureusement cette notion de modèle, il est important ensuite de pouvoir calculer les modèles d'une description donnée. C'est l'opération de *neutralisation de traits* qui nous en donne les moyens. Cette opération consiste dans une description à identifier deux nœuds porteurs de traits duaux, c'est-à-dire qui ont le même nom mais des polarités opposées. Par exemple, sur la figure 1, les nœuds syntaxiques *gn1* et *sujet* de  $D_{syn}$  portent les traits duaux  $cat \rightarrow gn$  et  $cat \leftarrow gn$ . La fusion des deux nœuds va permettre de neutraliser les deux traits qui vont s'unifier en un trait  $cat \leftrightarrow gn$ . Il se trouve qu'ici, incidemment, on obtient une deuxième neutralisation : celle des traits  $fonct \leftarrow ?^2$  et  $fonct \rightarrow sujet$  qui vont s'unifier dans  $fonct \leftrightarrow sujet$ . Le nœud qui résulte de la fusion de *gn1* et *sujet* est le nœud *gn1.sujet* que l'on peut voir sur la figure 2.

En itérant l'opération de neutralisation de traits, nous allons pouvoir spécifier pas à pas une description initiale et construire progressivement un modèle neutre de celle-ci. Dans notre exemple, après 4 neutralisations de traits effectués sur  $D_{syn}$ , on obtient l'arbre syntaxique complètement spécifié de la figure 2. Cette façon procédurale d'obtenir un modèle est équivalente

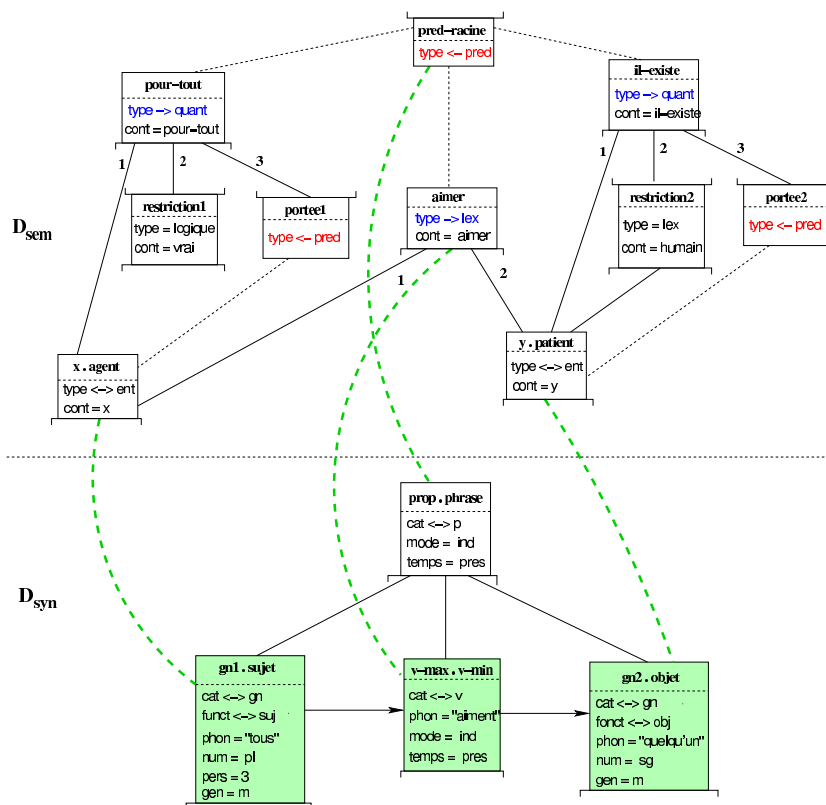


Figure 2: Description syntactico-sémantique à la fin de l'analyse syntaxique de la phrase *tous aiment quelque'un*.

à la façon déclarative de le définir (Perrier, 2003). Il est à noter que l'opération de neutralisation de traits qui permet de composer les descriptions entre elles est beaucoup plus riche que les opérations de composition d'arbres utilisées par les formalismes fondés sur les arbres tels que les TAG. Elle permet de superposer partiellement des arbres et s'apparente beaucoup plus à l'unification de structures de traits d'HPSG mais avec le contrôle par les polarités en plus.

<sup>2</sup>Le symbole “?” représente ici la *valeur indéterminée*, c'est-à-dire, comme les valeurs sont des disjonctions d'atomes, la disjonction de tous les atomes du domaine correspondant au trait, *fonct* en l'occurrence.

## 2 Le niveau sémantique et les descriptions de DAG polarisées

Nous ne proposons pas de formaliser une théorie sémantique particulière mais plutôt de fournir un cadre pour représenter différentes sémantiques objets. Pour illustrer notre propos, nous choisirons comme sémantique objet le calcul des prédicats mais nous aurions très bien pu prendre les graphes sémantiques de la théorie Sens-Texte (Mel'cuk, 1988). Certes, le choix n'est pas totalement arbitraire car le calcul des prédicats permet d'exprimer la notion de portée, notion qui pose des problèmes d'explosion combinatoire quand on cherche à calculer une représentation sémantique. Ce n'est pas un hasard si c'est en sémantique informatique, dans le domaine de la traduction, qu'ont vu le jour depuis une dizaine d'années une série de langages de représentation sémantique qui visent à résoudre ce problème en utilisant l'idée de sous-spécification (Reyle, 1993; Bos, 1995; Egg *et al.*, 1998; Copestake *et al.*, 1999).

Nous proposons de reprendre au niveau sémantique les deux idées-forces qui sont au cœur de notre représentation de la syntaxe : l'utilisation de la notion de description de structure pour exprimer la sous-spécification et celle de polarité pour contrôler la composition d'une structure complètement spécifiée. La différence avec le niveau syntaxique est que nous allons chercher à représenter des dépendances sémantiques entre objets sous forme de relations prédicat-arguments donc les structures sémantiques complètement spécifiées seront des DAG et non des arbres localement ordonnés : en effet, la précédence entre nœuds n'a plus de sens et un objet sémantique peut être argument de plusieurs prédicats à la fois. De façon analogue au niveau syntaxique, une description de DAG sémantique permettra de représenter un ensemble de DAG qui seront ses modèles. Si nous choisissons comme sémantique objet le calcul des prédicats, un DAG sera la représentation géométrique d'une formule logique et ses nœuds seront soit des prédicats, soit des individus. Les seules relations qui ont du sens entre les nœuds d'une description de DAG sont les relations prédicat-arguments qui sont représentées par des relations de domination immédiate et les relations de portée qui sont représentées par des relations de domination large. Comme au niveau syntaxique, les nœuds sont étiquetés par des structures de traits polarisés qui expriment cette fois des propriétés sémantiques. La notion de modèle neutre d'une description de DAG sémantique se définit de la même façon qu'au niveau syntaxique et la même opération de neutralisation de traits nous fournit une méthode de construction pas à pas de ces modèles. Pour une définition formelle complète, le lecteur peut se référer à (Perrier, 2003).

La figure 1, dans sa partie supérieure, nous montre un exemple de description de DAG sémantique,  $D_{sem}$ . Dans cet exemple, nous nous contentons d'utiliser deux traits *type* et *cont*. Le premier indique le type sémantique du nœud, prédicat lexical (*lex*), opérateur logique (*logique*) ou quantificateur (*quant*)<sup>3</sup>, et le second son contenu. Les deux quantificateurs associés à *tous* et *quelqu'un* sont représentés à l'aide de prédicats à 3 arguments : la *variable quantifiée*, sa *restriction* qui permet de caractériser logiquement l'entité représentée par cette variable et la *portée* du quantificateur. Dans notre exemple, la restriction correspondant à *tous* est vide d'où la valeur *vrai* de son trait *cont* alors que celle correspondant à *quelqu'un* est le prédicat *humain*. Comme au niveau syntaxique, des contraintes permettent de fixer la cardinalité de l'ensemble des fils d'un nœud et elles sont aussi représentées graphiquement par des crochets tournés vers le bas. Par exemple, la contrainte *pour-tout*  $> [x, restriction1, portee1]$  exprime le fait que *pour-tout* a exactement trois fils  $x, restriction1, portee1$  mais à la différence de la syntaxe, pour distinguer leurs rôles dans le prédicat, ils sont ordonnés dans une liste et leur rang est noté sur l'arc représentant leur rôle dans le prédicat. Comme un même nœud peut avoir plusieurs pères, nous avons aussi des contraintes fixant la cardinalité de l'ensemble des pères d'un nœud. Par exemple, dans  $D_{sem}$ , la contrainte  $\{ \} > pred-racine$  exprime que le nœud *pred-racine* a son ensemble de pères qui est vide donc que c'est une racine. Enfin, les relations de portée peuvent

<sup>3</sup>La valeur *pred* est une abréviation de la disjonction  $lex \vee logique \vee quant$ , étant donné que les valeurs de traits sont des disjonctions finies d'atomes.

être contraintes de la même façon que les relations de domination large au niveau syntaxique avec des structures de traits neutres les étiquetant.

La description  $D_{sem}$  a exactement deux modèles neutres au sens où nous le définissons dans (Perrier, 2003) sous forme de deux DAG qui peuvent être obtenus par 3 neutralisations de polarités (le lecteur pourra lui-même effectuer le calcul). Pour ensuite interpréter ces deux DAG obtenus sous forme de formules logiques, il est nécessaire de disposer d'une traduction logique de chacun de leurs nœuds. On peut le faire à l'aide du lambda-calcul en associant un lambda-terme à chaque nœud et en interprétant chaque relation père-fils comme une application du lambda-terme associé au nœud père à ceux qui sont associés à ses fils. Si on associe aux nœuds *pour-tout* et *il-existe* les lambda-termes respectifs  $\lambda x R P. \forall x ((Rx) \Rightarrow (Px))$  et  $\lambda x R P. \exists x ((Rx) \wedge (Px))$ , où  $x$ ,  $R$  et  $P$  correspondent à la variable quantifiée, la restriction et la portée, le lecteur en déduira facilement l'interprétation des deux DAG par les deux formules logiques  $\forall x (\exists y (humain(y) \wedge aimer(x, y)))$  et  $\exists y (humain(y) \wedge \forall x aimer(x, y))$  correspondant aux deux lectures possibles de la phrase *tous aiment quelqu'un*.

La proposition que nous faisons ici se distingue de la plupart des langages donnés en référence au début de la section par le fait qu'elle fournit un moyen de contrôler la saturation des structures sémantiques sous-spécifiées grâce au mécanisme des polarités. Il est un langage qui s'en rapproche, c'est la *Hole Semantics* (Bos, 1995) : les nœuds des descriptions  $y$  sont soit des constantes, soit des trous et c'est le mécanisme d'identification un à un des trous par des constantes qui fournit le moyen de réduire la sous-spécification et de produire les modèles d'une description donnée. Dans l'exemple du calcul des prédicats choisi comme sémantique objet, nous nous distinguons aussi de ce qui se fait dans la plupart des langages précédemment cités qui n'identifient les nœuds qu'avec des prédicats. Les nœuds de nos descriptions peuvent être soit des prédicats, soit des individus. Cette uniformisation de la représentation présente deux avantages : elle permet d'aller plus loin dans la sous-spécification en laissant indéterminé le type de l'argument d'un prédicat, individu ou prédicat lui-même. La deuxième raison est que cela simplifie beaucoup l'interface avec la syntaxe, les syntagmes pouvant correspondre aussi bien à des individus qu'à des prédicats.

### 3 L'interaction entre syntaxe et sémantique dans le processus d'analyse

Nous venons de décrire séparément les niveaux syntaxique et sémantique. Leur liage s'effectue par une simple fonction qui projette tout nœud syntaxique sur au plus un nœud sémantique. La figure 1 nous en fournit un exemple; la fonction de liage  $y$  est représentée à l'aide de traits pointillés. On y constate que certains nœuds syntaxiques n'ont aucune image sémantique et, dans l'autre sens, certains nœuds sémantiques, tels que les prédicats représentant les quantificateurs, n'ont aucun antécédent syntaxique.

Les IG sont lexicalisées. Une entrée lexicale associe un mot à une description syntaxique et une description sémantique couplées par la fonction de liage. Dans la description syntaxique, un nœud  $y$  joue le rôle privilégié d'*ancree*, c'est-à-dire que c'est lui qui représente la position du mot dans l'arbre syntaxique correspondant à la description.

Analyser une phrase avec les IG consiste tout d'abord à sélectionner une entrée dans le lexique pour chacun des mots de la phrase. En juxtaposant toutes les descriptions d'arbres syntaxiques sélectionnées, on obtient une unique description qui doit être complétée par des relations de précedence entre les ancrs exprimant l'ordre des mots dans la phrase et par un nœud racine, ici le nœud *phrase*, représentant la phrase attendue comme but de l'analyse. Il en résulte une description  $D_{syn}$ , telle celle que l'on peut voir à la figure 1, qui constitue le point de départ de l'analyse. De façon analogue, on réunit les descriptions de DAG sémantiques sélectionnées du lexique en une unique description  $D_{sem}$  couplée avec  $D_{syn}$ , comme on l'observe sur la figure



1. On ajoute une racine *pred-racine* à  $D_{sem}$  liée à la racine syntaxique *phrase* pour exprimer le but de l'analyse au niveau sémantique.

Le processus d'analyse est dirigé par la syntaxe, c'est-à-dire qu'il s'effectue en itérant les neutralisations de traits dans  $D_{syn}$ , jusqu'à que l'on obtienne un arbre syntaxique complètement spécifié où tous les traits sont neutres. Dans ce processus, la fonction de liage joue deux rôles :

- Elle contribue à spécifier progressivement  $D_{sem}$  car chaque fusion de deux nœuds syntaxiques entraîne la fusion des nœuds sémantiques qui sont leurs images, si elles existent.
- Le même mécanisme peut entraîner une réaction du niveau sémantique sur le niveau syntaxique. Certaines neutralisations au niveau syntaxique vont échouer parce que les neutralisations qu'elles entraînent au niveau sémantique échouent aussi.

A la fin d'une analyse syntaxique qui réussit, la description sémantique  $D_{sem}$  peut rester sous-spécifiée comme c'est le cas dans notre exemple et comme le montre la figure 2. Si l'on souhaite obtenir tous les modèles de  $D_{sem}$ , il faut continuer le processus de neutralisation de traits mais au niveau sémantique cette fois. Si nous le faisons dans notre exemple, nous allons, après 3 neutralisations, obtenir les deux DAG sémantiques attendus correspondant aux deux relations de portée possibles entre les deux quantificateurs. Cet exemple est très simple mais le lecteur trouvera des exemples de modélisation de phénomènes linguistiques plus complexes dans (Perrier, 2003) (propositions relatives appositives et restrictives, verbes à montée et à contrôle du sujet, etc.).

D'un point de vue théorique, un tel processus d'analyse est extrêmement coûteux mais, comme nous le montrons dans (Bonfante *et al.*, 2003), les polarités nous fournissent des méthodes originales d'analyse qui permettent en pratique d'éviter l'explosion combinatoire. Ces méthodes ont été implémentées dans l'analyseur syntaxique LEOPAR qui intègre la sémantique et qui est téléchargeable librement avec une grammaire et un lexique jouets du français<sup>4</sup>.

## 4 Autres approches de l'interface syntaxe-sémantique

Les IG présentent certaines similarités avec les TAG synchrones (Shieber & Schabes, 1990; Shieber, 1994). Les deux formalismes visent à lier deux niveaux de représentation qui utilisent le même principe de composition : neutralisation de polarités pour les IG et adjonction pour les TAG. Néanmoins, les deux formalismes ont de profondes différences. Déjà les TAG sont limitées par l'adjonction qui ne permet pas de faire de la superposition de structures mais en outre l'interface syntaxe-sémantique y est très rigide : toute adjonction au niveau sémantique doit être couplée avec une adjonction au niveau syntaxique, si bien que les arbres de dérivation syntaxique et sémantique sont isomorphes. Cet isomorphisme est un facteur majeur de rigidité même s'il peut être un peu assoupli (Rambow & Satta, 1996).

L'autre approche de la sémantique dans les TAG qui s'appuie sur les arbres de dérivation syntaxique est aussi très rigide. La représentation sémantique issue de l'arbre de dérivation est parfois en contradiction avec la représentation souhaitée (Candito & Kahane, 1998), ce qui a d'ailleurs amené à l'introduction de nouveaux formalismes (Rambow *et al.*, 1995). D'autre part, elle se prête mal à la représentation de phénomènes comme la quantification qui ne découlent pas mécaniquement de la syntaxe.

Gardent et Kallmeyer (2003) proposent une nouvelle approche qui s'appuie sur les arbres syntaxiques dérivés. Chaque arbre dérivé est couplé avec une formule logique sous-spécifiée dans le cadre de la *Hole Semantics* (Bos, 1995). L'interface syntaxe-sémantique utilise en plus des traits associés aux nœuds des arbres syntaxiques dérivés et qui indiquent avec quels individus

<sup>4</sup>L'adresse où le logiciel peut être téléchargé est : [www.loria.fr/equipes/calligramme/leopar](http://www.loria.fr/equipes/calligramme/leopar).

ou prédicats sémantiques ils sont en correspondance. Compte tenu du fait que la *Hole Semantics* est très proche de la représentation sémantique dans les IG, on a une approche semblable, la principale différence restant la représentation du niveau syntaxique.

Du côté des CG, de Groot (2001) et Muskens (2003) ont des propositions voisines l'une de l'autre qui visent à assouplir et à généraliser les CG. Ils proposent un formalisme à plusieurs niveaux paramétrables. L'un de ces niveaux peut être instancié par la syntaxe et un autre par la sémantique. A chaque niveau, c'est le fragment implicatif de la logique linéaire intuitionniste qui est utilisé pour gérer la composition des structures, ce qui présente certaines limites dans le pouvoir d'expression. En outre, tous les niveaux sont isomorphes en un certain sens : toute opération à un niveau qui est une opération de déduction logique élémentaire (application ou abstraction) doit être couplée avec une opération du même type à l'autre niveau. Cet isomorphisme ne conduit pas à la même rigidité que les TAG synchrones du fait la latitude qui est offerte quant au langage d'instanciation de chaque niveau.

On trouve aussi une grande souplesse dans la représentation de l'interface syntaxe-sémantique du côté des grammaires de dépendance dans le formalisme proposé par Kahane (2002), les *Grammaires d'Unification Sens-Texte (GUST)*. Suivant la théorie Sens-Texte (Mel'cuk, 1988), celui-ci comporte 3 niveaux : sémantique, syntaxique et phonologique. Les structures représentées au niveau sémantique et syntaxique sont respectivement des DAG et des arbres et le liage entre les deux se fait par ensembles de nœuds car, au niveau syntaxique, les nœuds représentent des mots et non des syntagmes. Le fait que la quantification et les relations de portée ne soient pas traitées au niveau sémantique et que les nœuds représentent des mots ne rend pas nécessaire la représentation de la sous-spécification comme dans une approche syntagmatique car les dépendances non bornées sont traitées de la même façon que les dépendances locales. Dans sa présentation originelle, les GUST utilisaient l'unification comme mécanisme de composition des structures sans système de contrôle de la saturation de ces structures. Cette faiblesse est maintenant corrigée de la même façon que dans les IG avec un système de polarités (Kahane, 2004).

L'intégration de la *Minimal Recursion Semantics* dans HPSG (Copestake *et al.*, 1999) fournit un autre exemple de souplesse de l'interface syntaxe-sémantique qui est assurée par la co-indexation dans des structures de traits codant en même temps le niveau syntaxique et le niveau sémantique. L'unification sert de principe de composition de structures et elle peut être vue comme une généralisation de la superposition partielle de DAG, telle qu'elle se présente dans les IG. Lui manque un mécanisme de contrôle de la saturation des structures qui est garantie a posteriori par des principes de bonne formation.

## Conclusion

L'intégration de la sémantique dans les IG que nous proposons se veut avant tout un cadre formel pour exprimer la représentation sémantique d'un énoncé et l'interface avec sa représentation syntaxique. Elle ne préjuge en rien des choix linguistiques qui en permettent la réalisation. Ce cadre formel se caractérise par trois propriétés importantes : la notion de description permet d'exprimer la sous-spécification aux deux niveaux, le mécanisme de composition des structures complètement spécifiées tant au niveau sémantique que syntaxique est guidé par un système de polarités et l'interface syntaxe-sémantique est souple et simple.

## Remerciements

Merci à Benoit Crabbé et Sylvain Kahane pour la relecture de cet article et leurs commentaires pertinents.

## Références

- BLACHE P. (2001), *Les Grammaires de Propriétés : des contraintes pour le traitement automatique des langues naturelles*, Hermès Sciences.
- BONFANTE G., GUILLAUME B., PERRIER G. (2003), Analyse syntaxique électrostatique, *Traitement Automatique des Langues*, Vol. , A paraître.
- BOS J. (1995), Predicate logic unplugged, dans P. DEKKER & M. STOKHOF, Eds., *10th Amsterdam Colloquium*, p. 133–142.
- CANDITO M.-H., KAHANE S. (1998), Can the derivation tree represent a semantic graph? an answer in the light of Meaning-Text Theory, *TAG+4, Philadelphia*, p. 21–24.
- CARPENTER B. (1998), *Type-logical Semantics*, Cambridge, Massachusetts, MIT Press.
- COPESTAKE A., FLICKINGER D., SAG I. (1999), Minimal Recursion Semantics - an Introduction, Draft.
- DE GROOTE P. (2001), Towards Abstract Categorical Grammars, *Association for Computational Linguistics, 39th Annual Meeting and 10th Conference of the European Chapter, Toulouse, France*, p. 148–155.
- EGG M., NIEHREN J., RUHRBERG P., XU F. (1998), Constraints over lambda structures in semantic underspecification., *17th International Conference on Computational Linguistics and 36th Annual Meeting of the Association for Computational Linguistics (COLING/ACL'98), Montreal, Quebec, Canada*.
- GARDENT C., KALLMEYER L. (2003), Semantic construction in FTAG, *EACL'2003, Budapest, Hungary*.
- KAHANE S. (2002), *Grammaire d'Unification Sens-Texte - Vers un modèle mathématique articulé de la langue*, Habilitation à diriger des recherches, Université Paris 7.
- KAHANE S. (2004), Grammaires d'unification polarisées, *11ième Conférence annuelle sur le Traitement Automatique des Langues Naturelles (TALN'04), Fès, Maroc, France*, Soumis.
- MEL'CUK I. (1988), *Dependency Syntax: Theory and Practice*, Albany, N.Y.: The SUNY Press.
- MONTAGUE R. (1970), Universal grammar, *Theoria*, Vol. 36, 373–398, Reprinted in R. Thomason, editor, *Formal Philosophy*, 188-221, New Haven: Yale University Press.
- MUSKENS R. (2003), Lambda Grammars, *Prospects and Advances in the Syntax/Semantics Interface, Lorraine-Saarland Workshop Series, Nancy*, p. 29–32.
- PERRIER G. (2002), Descriptions d'arbres avec polarités : les grammaires d'interaction, *9ième Conférence annuelle sur le Traitement Automatique des Langues Naturelles (TALN'02), Nancy, France, 2002*.
- PERRIER G. (2003), *Les grammaires d'interaction*, Habilitation à diriger des recherches, Université Nancy2, Film de la soutenance visible à l'URL: <http://www.inria.fr/multimedia/Didactheque-fra.html>.
- RAMBOW O., SATTI G. (1996), Synchronous Models of Language, *ACL'96, Santa Cruz*.
- RAMBOW O., VIJAY-SHANKER K., WEIR D. (1995), D-tree grammars, *ACL'95, Cambridge, USA*, p. 151–158.
- REYLE U. (1993), Dealing with ambiguities by underspecification: Constrcution, Representation and Deduction, *Journal of Semantics*, Vol. 10, 123–179.
- SHIEBER S. (1994), Restricting the Weak-Generative Capacity of Synchronous Tree-Adjoining Grammars, *Computational Intelligence*, Vol. 10(4), 371–385.
- SHIEBER S., SCHABES Y. (1990), Synchronous Tree-Adjoining Grammars, *CoLing'90, Helsinki*, volume 3, p. 1–6.
- VIJAY-SHANKAR K. (1992), Using description of trees in a tree adjoining grammar, *Computational Linguistics*, Vol. 18(4), 481–517.