



**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. Revue TAL : traitement automatique des langues, 2005, 45 (3), pp.123–144. inria-00000275

**HAL Id: inria-00000275**

**<https://inria.hal.science/inria-00000275>**

Submitted on 21 Sep 2005

**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.* 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. À 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.

*ABSTRACT.* 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-CLÉS :* formalisme grammatical, interface syntaxe-sémantique, sous-spécification, polarités.

*KEYWORDS:* 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 [PER 02] 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. On retrouve la même idée dans les grammaires de dépendances que Tesnière présentait déjà en utilisant la métaphore de la réaction chimique [TES 34] : une phrase peut être comparée à une molécule et les mots qui la composent à ses atomes qui se combinent pour saturer leurs valences.

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. La logique linéaire et son concept de réseau de démonstration en fournit l'arrière-plan théorique et ce n'est pas un hasard si le formalisme des IG est issu d'une étude des réseaux de démonstration de logique linéaire intuitionniste [PER 03]. 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*. Cette neutralisation de traits duaux est aussi présente dans les grammaires minimalistes [STA 97] et il serait intéressant, mais cela dépasse l'objectif de cet article, d'étudier plus à fond cette similarité apparente.

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 [VIJ 92] 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 [BLA 01]. 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 do-

mination et de précédence 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 rapport 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 [MON 70, CAR 98]. Le cadre utilisé pour l'exprimer est celui du lambda-calcul simplement typé de Church [CHU 40]. 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 simplement typé 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 du lambda-terme  $([aime]_{GN \rightarrow GN \rightarrow P} [Marie]_{GN} [Jean]_{GN})$  formé à partir des entrées lexicales des mots de la phrase qu'on peut représenter ainsi :

$$\begin{aligned} [aime]_{GN \rightarrow GN \rightarrow P} &: \lambda x \lambda y . (y * aime * x) \\ [Marie]_{GN} &: Marie \\ [Jean]_{GN} &: Jean \end{aligned}$$

La réduction du lambda-terme  $([aime]_{GN \rightarrow GN \rightarrow P} [Marie]_{GN} [Jean]_{GN})$  va nous fournir la forme phonologique de la phrase, c'est-à-dire *Jean \* aime \* Marie* où "\*" est l'opérateur de concaténation entre chaîne de caractères.

La représentation sémantique s'obtient par projection de chaque type syntaxique en un type sémantique. Cette projection étant un morphisme, il suffit de fixer la projection des types atomiques. Dans notre exemple, il s'agit en l'occurrence de *GN* et *P* qui se projettent sémantiquement selon les types *entité* (*e*)<sup>1</sup> et *valeur de vérité* (*t*). La structure sémantique de la phrase est une copie parfaite de la structure syntaxique

---

1. Montague associe plutôt le type sémantique  $(e \rightarrow t) \rightarrow t$  à *GN* pour exprimer qu'un individu est défini comme l'ensemble des propriétés qui le caractérisent. La simplification que nous opérons ici ne change rien à notre démonstration.

puisqu'elle s'exprime par le lambda-terme  $([[\textit{aime}]]_{e \rightarrow e \rightarrow t} [[\textit{Marie}]]_e [[\textit{Jean}]]_e)$  formé à partir des entrées lexicales sémantiques des mots de la phrase qu'on peut représenter ainsi :

$$\begin{aligned} [[\textit{aime}]]_{e \rightarrow e \rightarrow t} & : \lambda x \lambda y .\textit{aimer } x y \\ [[\textit{Marie}]]_e & : m \\ [[\textit{Jean}]]_e & : j \end{aligned}$$

La représentation sémantique de la phrase  $(\textit{aimer } m j)$  s'obtient par réduction du lambda-terme  $([[\textit{aime}]]_{e \rightarrow e \rightarrow t} [[\textit{Marie}]]_e [[\textit{Jean}]]_e)$ .

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 l'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 de 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-spécifi ée* 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) ; au niveau syntaxique la sous-spécification d'arbres est exprimée par la notion de description d'arbre et de façon similaire au niveau sémantique, la sous-spécification de DAG sera exprimée par la notion de *description 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<sup>2</sup>. Nous terminerons dans la section 4 par une comparaison avec d'autres formalismes.

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

### 1.1. La forme des descriptions d'arbres polarisées

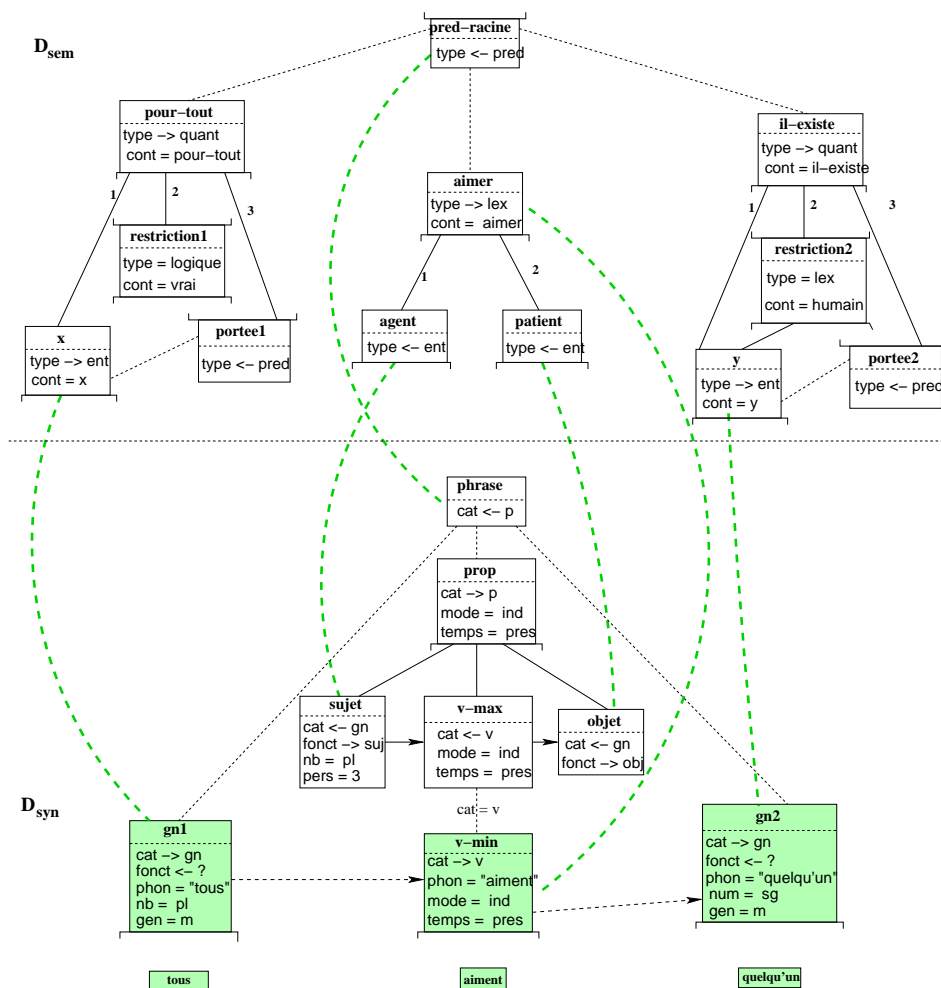
Une description d'arbre syntaxique est un ensemble de nœuds et de relations de domination et de précédence entre ces nœuds. Les nœuds représentent des syntagmes et les relations des contraintes relatives entre ces syntagmes. Les propriétés morpho-syntaxiques de ces syntagmes sont exprimées par des structures 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 en-tête leur nom et pour corps la structure de traits attachée. Les rectangles en grisé représentent les mots de la phrase qui sont les ancres des descriptions qui ont été extraites du lexique et rassemblées pour former  $D_{syn}$ .

Les relations entre les nœuds d'une description peuvent être de 4 types distingués graphiquement par le style des traits les représentant :

**relations de domination immédiate :**  $N > M$  signifie que le syntagme  $M$  est un constituant immédiat de  $N$ , ce qui est représenté graphiquement par un trait continu. On peut en plus figer l'ensemble des constituants immédiats d'un syntagme  $N$  à l'aide de la relation  $N > \{N_1, N_2, \dots, N_p\}$ . Cette relation signifie que le syntagme  $N$  a exactement  $p$  constituants immédiats  $\{N_1, N_2, \dots, N_p\}$ . Elle est exprimée graphiquement par un crochet tourné vers le bas au pied du rectangle représentant le nœud  $N$ . Par exemple dans  $D_{syn}$ , le crochet associé au nœud *prop* représente la relation  $prop > \{sujet, v-max, objet\}$  qui signifie que le syntagme *prop* se décompose en trois constituants immédiats : *sujet*, *v-max*, *objet*.

**relations de domination sous-spécifiée :**  $N \dot{>} M$  signifie que le syntagme  $M$  est inclus dans  $N$  à une profondeur indéterminée, ce qui est représenté graphiquement par un trait discontinu. À la limite,  $M$  peut s'identifier à  $N$ . Cette relation permet d'exprimer à la fois une dépendance syntaxique non bornée et la possibilité d'appliquer des modifieurs à un syntagme. Dans les deux cas, on peut contraindre la relation qui devient  $N \dot{>} [f_1 = v_1, \dots, f_n = v_n]M$  avec la signification suivante : tout syntagme qui est inclus dans  $N$  et qui contient  $M$  doit avoir sa structure de traits qui s'unifie avec  $[f_1 = v_1, \dots, f_n = v_n]$  en un sens qui sera précisé par la suite. Dans le cas d'une dépendance non bornée, cela permet de modéliser les barrières à l'extraction. Dans le cas de modifieurs, cela permet de contraindre leur type : modifieurs de verbes, de noms communs, de

2. En analyse, la fonction de liage entre les deux niveaux va permettre de diriger la composition sémantique par la composition syntaxique, alors qu'en génération, ce serait le contraire.



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

propositions . . . Ainsi, la relation de domination  $v-max \succ [cat = v] v-min$  signifie que le syntagme  $v-min$ , qui représente le verbe nu, *aiment* en l'occurrence, est inclus dans  $v-max$  qui représente le noyau verbal complet. En effet, peuvent s'adjoindre éventuellement au verbe une négation, des clitiques ou des adverbes pour former ce noyau verbal. Tous sont des modificateurs de verbes, ce qui est exprimé par la contrainte  $cat = v$  qui signifie que tout nœud dominé par  $v-max$  et dominant  $v-min$  (au sens large) a comme catégorie grammaticale  $v$ .

**relations de précedence immédiate :**  $N \prec M$  signifie que le syntagme  $N$  précède immédiatement le syntagme  $M$  dans l'ordre linéaire des mots de la phrase, ce qui est représenté graphiquement par une flèche continue. Par exemple, la relation *sujet*  $\prec$  *v-max* signifie que le sujet précède immédiatement le noyau verbal.

**relations de précedence sous-spécifiée :**  $N \prec^* M$  signifie que le syntagme  $N$  précède le syntagme  $M$ . Ainsi l'ordre des mots dans la phrase *tous aiment quelqu'un* est exprimé par les deux relations *gn1*  $\prec^*$  *v-min* et *v-min*  $\prec^*$  *gn2*<sup>3</sup>.

Ce système de relations n'est pas figé et rien n'interdit de l'enrichir pour traduire certaines spécificités d'une langue.

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.

Dans la description  $D_{syn}$ , le nœud *sujet* est porteur de trois types de polarités : les traits *cat*  $\leftarrow$  *gn* et *fonct*  $\rightarrow$  *sujet* signifient que le nœud attend un groupe nominal pour lui fournir la fonction *sujet*.

Les valeurs d'un trait  $t$  sont des ensembles finis d'atomes  $\{v_1, \dots, v_n\}$ <sup>4</sup> pris dans un domaine fini  $D_t$  associé à  $t$ . Intuitivement, cela signifie que  $t$  peut prendre indifféremment une des valeurs atomiques  $v_1, \dots, v_n$ . La valeur d'un trait peut être munie d'un indice utilisé pour exprimer le partage de cette valeur par d'autres traits.

En résumé, une description d'arbre polarisé est définie formellement sur une signature constituée d'un ensemble d'identificateurs de nœuds  $\mathcal{N}$  et d'un ensemble de traits morpho-syntaxiques  $\mathcal{T}$ , un trait étant un couple formé d'un nom  $t$  et d'un domaine fini de valeurs atomiques  $D_t$ , de la façon suivante :

**Definition 1.1** Une description d'arbre polarisée est un ensemble fini de relations de la forme :

$(N : (t, p, \{v_1, \dots, v_n\}))$  ou  $(N : (t, p, \langle i \rangle \{v_1, \dots, v_n\}))$  avec  $N \in \mathcal{N}$ ,  $t \in \mathcal{T}$ ,  $p \in \{\rightarrow, \leftarrow, =, \leftrightarrow\}$ ,  $v_k \in D_t$  et  $i \in \mathbb{N}$ ; ces relations sont des affectations de valeurs

3. L'ordre des mots dans la phrase n'est pas exprimé à l'aide de la relation de précedence immédiate car, en cours d'analyse, des catégories vides peuvent s'insérer entre les mots pleins de la phrase et ceci ne peut pas être prévu. Ces catégories vides représentent soit des traces d'arguments en position non canonique ou des sujets d'infinitives non explicites.

4. Lorsque l'ensemble se réduit à un singleton  $\{v_1\}$ , on identifie le singleton avec son unique élément  $v_1$ .



de traits ; elles signifient que le nœud  $N$  est muni du trait  $t$  avec la polarité  $p$  et la valeur  $\{v_1, \dots, v_n\}$  qui est la disjonction des valeurs  $v_1, \dots, v_n$  ; l'indice  $\langle i \rangle$  permet de partager cette valeur avec d'autres traits.

–  $M > N, M > \{N_1, \dots, N_p\}, M < N$  ou  $M \overset{*}{<} N$ , avec  $M, N, N_i \in \mathcal{N}$  ; ces relations sont dans l'ordre des relations de domination immédiate, des relations d'énumération de l'ensemble des fils d'un nœud, des relations de précédence immédiate ou de précédence large.

–  $M \overset{*}{>} [t_1 = \{v_1^1, \dots, v_1^{p_1}\}, \dots, t_n = \{v_n^1, \dots, v_n^{p_n}\}] N$  avec  $M, N \in \mathcal{N}$ ,  $t_i \in \mathcal{T}$  et  $v_i^k \in D_{t_i}$  ; cette relation indique que le nœud  $M$  domine le nœud  $N$  et que les éventuels nœuds intermédiaires sont soumis à la contrainte  $t_1 = \{v_1^1, \dots, v_1^{p_1}\}, \dots, t_n = \{v_n^1, \dots, v_n^{p_n}\}$ .

## 1.2. Modèles d'une description d'arbre

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. Pour préciser le rapport entre une description et ses modèles, il est nécessaire de préciser la forme de ces modèles, c'est-à-dire des *arbres syntaxiques*.

**Definition 1.2** *Un arbre syntaxique est un arbre fini ordonné dont chaque nœud est muni d'une structure de traits de la forme  $[t_1 = v_1, \dots, t_n = v_n]$  avec  $v_i \in D_{t_i}$ .*

La définition d'un modèle d'une description s'inscrit dans la théorie des modèles pour la logique propositionnelle, l'univers d'interprétation d'une description étant restreint aux arbres syntaxiques tels qu'ils viennent d'être définis.

**Definition 1.3** *Un modèle d'une description d'arbre polarisé  $D$  est un couple  $(T, I)$  dans lequel  $T$  est un arbre syntaxique et  $I$  une fonction d'interprétation qui associe chaque nœud de  $D$  à un nœud de  $T$  sous les conditions suivantes :*

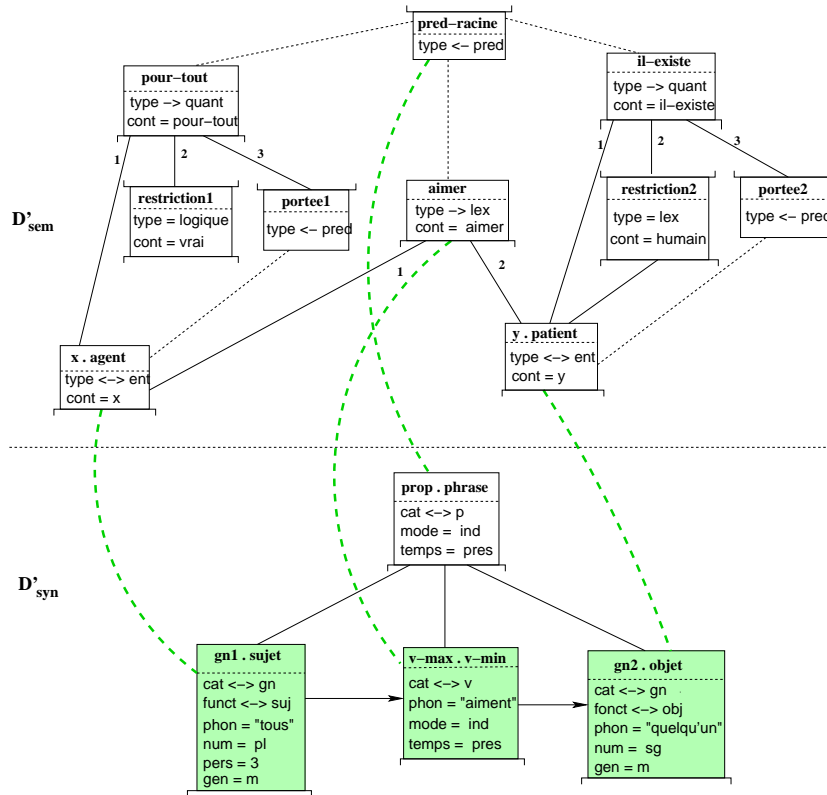
- si  $(N : (t, p, \{v_1, \dots, v_n\})) \in D$  ou  $(N : (t, p, \langle i \rangle \{v_1, \dots, v_n\})) \in D$ , il existe un indice  $k$  pour lequel  $(t = v_k)$  appartient à la structure de traits de  $I(N)$  ;
- si  $(N_1 : (t, p_1, \langle i \rangle w_1)) \in D$  et  $(N_2 : (t, p_2, \langle i \rangle w_2)) \in D$ , le même trait  $t = v$  appartient aux structures de traits de  $I(N_1)$  et  $I(N_2)$  ;
- si  $(M > N) \in D$ ,  $I(N)$  est un fils de  $I(M)$  dans  $T$  ;
- si  $(M > \{N_1, \dots, N_p\}) \in D$ ,  $I(N)$  a  $p$  fils dans  $T$ , qui sont  $I(N_1), \dots, I(N_p)$  ;
- si  $(M \overset{*}{>} [t_1 = \{v_1^1, \dots, v_1^{p_1}\}, \dots, t_n = \{v_n^1, \dots, v_n^{p_n}\}] M) \in D$ , alors  $I(M)$  domine  $I(N)$  dans  $T$  et, pour tout nœud  $N'$  de  $T$  qui est dominé par  $I(M)$  et qui

---

5. Dans un arbre ordonné, chaque nœud a l'ensemble de ses fils totalement ordonné et cet ordre est ensuite propagé à leurs descendants.

domine  $I(N)$  (au sens large), et pour tout trait  $t_i$ , il existe un indice  $k$  pour lequel ( $t_i = v_i^k$ ) appartient à la structure de traits de  $I(N')$  ;

- si  $(M \prec N) \in D$ ,  $I(M)$  précède immédiatement  $I(N)$  dans  $T^6$  ;
- si  $(M \overset{*}{\prec} N) \in D$ ,  $I(M)$  précède  $I(N)$  dans  $T$ .



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

La description d'arbre syntaxique  $D'_{syn}$  à la partie inférieure de la figure 2, après remplacement de toutes les polarités  $\leftrightarrow$  par  $=$ , représente un arbre syntaxique complètement spécifié qui est un modèle de la description  $D_{syn}$  de la figure 1 au sens qui vient d'être défini. La fonction d'interprétation  $I$  est représentée implicitement : l'en-tête de chaque nœud est une fusion des noms de ses antécédents dans  $D_{syn}$ .

6. La notion de précedence immédiate découle mécaniquement de la relation d'ordre total qui est instaurée entre les fils de chaque nœud et du fait que les arbres sont des arbres finis.

### 1.3. Modèles neutres et minimaux d'une description d'arbre syntaxique

D'un point de vue linguistique, la notion générale de modèle n'est pas suffisante pour deux raisons :

- toute description qui a au moins un modèle en a une infinité mais tous ne sont pas pertinents à représenter la syntaxe d'une phrase ; pour cela, ils doivent avoir une propriété de *minimalité*, en ce sens qu'ils doivent ajouter un minimum d'information par rapport à la description qu'ils représentent ;

- les polarités n'interviennent pas du tout dans la notion générale de modèle ; or, elles ont été introduites pour exprimer la différence entre structures syntaxiques saturées et structures syntaxiques non saturées ; les structures saturées représentent la syntaxe de phrases grammaticales, ce qui est exprimé par une propriété particulière des modèles qui est leur *neutralité*.

Définissons plus précisément ces propriétés de minimalité et de neutralité.

**Definition 1.4** *Un modèle  $(T,I)$  d'une description syntaxique  $D$  est minimal si toute relation de parenté dans  $T$  interprète une relation de domination immédiate dans  $D$  et si tout trait  $t = v$  dans  $T$  interprète un trait de  $D$ .*

**Definition 1.5** *Un modèle  $(T,I)$  d'une description syntaxique  $D$  est neutre si pour tout trait de  $T$ , tous les traits de  $D$  qu'il interprète sont neutres sauf éventuellement un trait qui est saturé ou deux traits qui sont l'un positif et l'autre négatif.*

Cette propriété de neutralité donne une sémantique aux polarités : alors que les polarités neutres ne jouent aucun rôle, toute polarité positive doit nécessairement fusionner avec une polarité négative dans un modèle neutre et réciproquement, alors qu'une polarité saturée ne peut fusionner ni avec l'une, ni avec l'autre. Ce système de polarités n'est pas le seul possible et S. Kahane propose un système plus sophistiqué qui lui permet de polariser les formalismes linguistiques les plus classiques [KAH 04].

Il est facile de vérifier que l'arbre syntaxique  $D'_{syn}$  de la figure 2 est un modèle minimal et neutre de la description  $D_{syn}$  de la figure 1.

A partir de la notion de modèle neutre et minimal, il est intéressant de définir une relation de *raffi nement* et une relation d'*équivalence* entre descriptions.

**Definition 1.6** *Une description d'arbre syntaxique  $D_1$  est un raffi nement d'une description d'arbre syntaxique  $D_2$  si, pour tout modèle neutre et valide  $(T, I_1)$  de  $D_1$ , il existe un modèle neutre et valide  $(T, I_2)$  de  $D_2$ .*

*Deux descriptions d'arbre syntaxique sont équivalentes si elles sont un raffi nement l'une de l'autre.*

La *description vide*  $\emptyset$  joue un rôle particulier dans la mesure où elle n'a aucun modèle et toute description qui lui est équivalente sera qualifiée d'*incohérente*. Il est

d'autres descriptions qui occupent une place particulière : les *descriptions saturées* qui ne contiennent aucun trait positif et négatif.

#### 1.4. Construction de modèles neutres et minimaux par neutralisation de polarités

La définition qui vient d'être établie nous permet de vérifier si un arbre syntaxique donné est un modèle neutre et minimal d'une description donnée mais elle ne nous fournit pas le moyen de construire un tel modèle. C'est l'opération de *neutralisation de polarités* qui va constituer la base du calcul qui va permettre de construire des modèles neutres et minimaux de descriptions.

**Definition 1.7** Si une description  $D$  comprend deux relations  $(N_1 : t \rightarrow v_1)$  et  $(N_2 : t \leftarrow v_2)$ , la neutralisation des deux polarités associées consiste à fusionner les nœuds  $N_1$  et  $N_2$  dans  $D$ .

Si nous notons la description résultat  $D'$ , nous écrivons :  $D \xrightarrow{N_1 \stackrel{t}{=} N_2} D'$ . La clôture réflexive et transitive de cette relation entre descriptions est notée  $\xrightarrow{*}$ .

Cette opération va permettre de traduire en termes opérationnels la propriété de neutralité des modèles, où tout trait positif de la description correspondante doit fusionner avec un trait négatif dual et réciproquement. Cette traduction se fonde sur la proposition suivante :

**Proposition 1.1** Si  $D_1$  et  $D_2$  sont deux descriptions d'arbre syntaxique telles que  $D_1 \xrightarrow{*} D_2$ , alors  $D_2$  est un raffinement de  $D_1$ .

Si une description d'arbre syntaxique a un modèle neutre et minimal  $(T, I_1)$ , il existe une description saturée  $D_2$  telle que  $D_1 \xrightarrow{*} D_2$  et  $D_2$  a un modèle neutre et minimal  $(T, I_2)$ .

En conséquence, l'itération de l'opération de neutralisation de polarités termine nécessairement sur l'alternative suivante :

- soit nous obtenons une description non saturée et aucune neutralisation supplémentaire n'est possible ; nous pouvons conclure alors que la description est incohérente ;
- soit le processus se termine avec une description saturée et nous pouvons affirmer que tout modèle neutre et minimal de cette description correspond à un modèle neutre et minimal de la description initiale.

Nous avons maintenant une méthode pour construire les modèles neutres et minimaux d'une description donnée : elle consiste à itérer les neutralisations jusqu'à ce qu'on obtienne une description saturée. Bien entendu, la méthode n'est pas déterministe et il faut explorer toutes les itérations possibles pour obtenir tous les modèles. La proposition 1.1 en garantit la correction et la complétude.

L'intérêt de la méthode est qu'elle conduit à spécifier de plus en plus la description de départ de telle façon qu'il est possible de la simplifier et que la détermination du modèle cherché devient de plus en plus simple. Il est une simplification qui peut être effectuée immédiatement à la suite d'une neutralisation : elle consiste à unifier les structures de traits des nœuds qui ont été fusionnés. Étant donné la présence des polarités, cette unification n'est pas classique et nécessite d'introduire une opération d'addition entre polarités.

**Definition 1.8** La somme  $p+q$  de deux polarités  $p$  et  $q$  est donnée par la table suivante où une case vide signifie que l'addition n'est pas définie pour les valeurs de  $p$  et  $q$  correspondantes.

	←	→	=	↔
←			↔	←
→		↔		→
=	←	→	=	↔
↔			↔	

L'unification des structures de traits s'appuie alors sur la proposition suivante :

**Proposition 1.2** Soit une description d'arbre syntaxique  $D$  qui contient deux relations  $(N : (t, p_1, v_1))$  et  $(N : (t, p_2, v_2))$ . Si la somme de  $p_1$  et  $p_2$  n'existe pas ou si  $v_1 \cap v_2$  est vide, alors  $D$  est incohérente ; sinon, on obtient une description équivalente à  $D$  en remplaçant les deux relations précédentes par  $(N : (t, p_1 + p_2, v_1 \cap v_2))$ .

Par exemple, sur la figure 1, les nœuds syntaxiques  $gnl$  et  $sujet$  de  $D_{syn}$  portent les traits duaux  $cat \rightarrow gn$  et  $cat \leftarrow gn$ . Si on choisit de neutraliser ces traits, il faut fusionner les deux nœuds  $gnl$  et  $sujet$ . On peut ensuite simplifier la description obtenue en vertu de la proposition 1.2. Si l'on note  $gnl.sujet$  le nœud résultant de la fusion des nœuds :  $gnl$  et  $sujet$ , les relations  $(gnl : cat \rightarrow gn)$  et  $(sujet : cat \leftarrow gn)$  vont être remplacées par  $(gnl.sujet : cat \leftrightarrow gn)$ . Il se trouve qu'ici, incidemment, on obtient une deuxième neutralisation : les relations  $(gnl : fonct \leftarrow ?)^7$  et  $(sujet : fonct \rightarrow suj)$  vont être remplacées par  $(gnl.sujet : fonct \leftrightarrow suj)$ . Ces simplifications apparaissent sur la figure 2 qui montre la description  $D'_{syn}$  obtenue après 4 neutralisations de traits effectués sur  $D_{syn}$ . On obtient un arbre syntaxique complètement spécifié qui constitue un modèle neutre et minimal de la description initiale.

Dans cet exemple qui est très élémentaire, les simplifications effectuées au fur et à mesure portent essentiellement sur les traits mais dans d'autres cas plus sophistiqués [PER 03], celles-ci peuvent porter sur la structure de la description. Une neutralisation de polarités peut entraîner par exemple la superposition partielle de deux sous-arbres.

Il apparaît ainsi que l'opération de neutralisation de polarités en tant qu'opération de base de composition syntaxique dans les IG est beaucoup plus souple que les opérations de composition d'arbres utilisées par les formalismes fondés sur les arbres tels

7. Le symbole “?” représente ici la valeur indéterminée, c'est-à-dire l'ensemble des valeurs du domaine associé au trait concerné.

que les TAG. En permettant la superposition partielle d'arbres, elle s'apparente beaucoup plus à l'unification de structures de traits d'HPSG mais avec le contrôle par les polarités en plus.

## 2. Le niveau sémantique et les descriptions de DAG sémantiques

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. Ce calcul permet entre autres d'exprimer la notion de portée mais cette notion pose des problèmes d'explosion combinatoire quand on cherche à calculer une représentation sémantique sur un corpus réaliste. Ce n'est pas un hasard si c'est en informatique appliquée à 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 [REY 93, BOS 95, EGG 98, COP 99].

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 des structures. La formalisation du niveau sémantique va donc reprendre partiellement celle du niveau syntaxique. Nous ne l'exposerons pas de façon complète et nous n'en développerons que les aspects originaux.

### 2.1. La forme des descriptions de DAG sémantiques

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 donc le nœud correspondant peut avoir plusieurs pères.

Une description de DAG sémantique est un ensemble de relations sur des nœuds représentant des individus ou des prédicats qui sont soit des relations structurelles soit des affectations de traits sémantiques.

Les relations structurelles sont de deux sortes :

**relations prédicat-arguments :**  $N > M$  signifie que  $M$  est un argument du prédicat  $N$ . Au niveau syntaxique, la fonction syntaxique qu'occupe un argument par rapport à une tête peut être exprimée comme un trait associé au nœud représentant cet argument car il n'y a aucune ambiguïté possible : un nœud ne peut être argument que d'une seule tête. Au niveau sémantique, ce n'est plus le cas ; une façon de différencier les rôles des arguments d'un prédicat est d'ordonner ceux-ci dans une liste ; ainsi, la relation  $N > [N_1, \dots, N_p]$  signifie que le prédicat  $N$

a exactement  $p$  arguments  $N_1, \dots, N_p$  pris dans cet ordre. La partie supérieure de la figure 1 nous montre une description de DAG sémantique  $D_{sem}$  associée par une IG particulière à la phrase *tous aiment quelqu'un*. Les pronoms *tous* et *quelqu'un* sont représentés sémantiquement comme des quantificateurs. Un quantificateur est un prédicat à trois arguments : l'individu quantifié et deux prédicats qui constituent la restriction et la portée de la quantification. Cela se traduit par exemple pour le pronom *tous* par la relation  $pour\text{-}tout \succ [x, restriction1, portee1]$ . Graphiquement, le fait que le nombre des arguments soit fixé est représenté par un crochet tourné vers le bas à la base du rectangle représentant le prédicat et le rang de chaque argument est indiqué par un entier étiquetant l'arc correspondant.

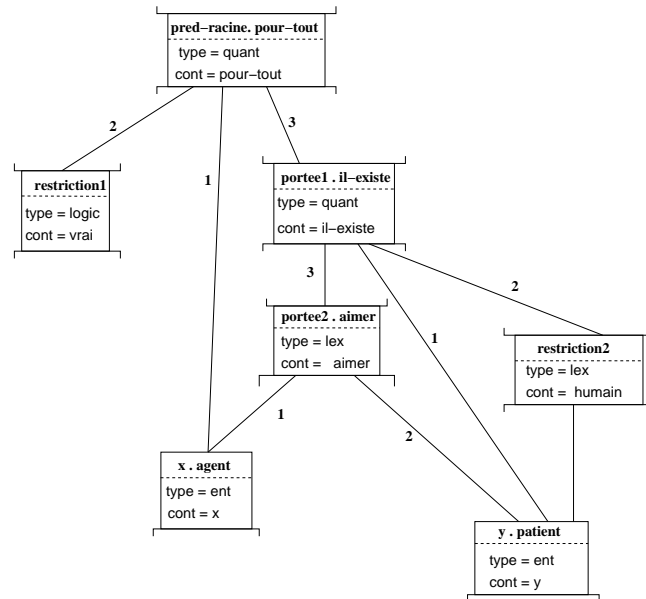
Contrairement au niveau syntaxique, le nombre de pères d'un nœud peut être variable et il est possible de figer ceux-ci par une relation du type  $\{N_1, \dots, N_p\} \succ N$  qui signifie que le nœud  $N$  a exactement  $p$  pères  $N_1, \dots, N_p$ . Dans notre exemple, la relation  $\{pour\text{-}tout\} \succ restrict1$  signifie que le nœud *restrict1* a un père unique *pour-tout*.

**relations de champ :**  $N \overset{*}{\succ} M$  signifie que  $M$  est dans le champ du prédicat  $N$ . A la limite  $M$  peut se confondre avec  $N$ . Dans notre exemple,  $portee1 \overset{*}{\succ} x$  signifie que l'individu  $x$  est l'argument d'un prédicat qui est dans le champ du prédicat *portee1*, c'est-à-dire que  $x$  est dans la portée du quantificateur *pour-tout*. Comme au niveau syntaxique, une relation de champ peut être restreinte par une structure de traits exprimant des contraintes sur les nœuds intermédiaires.

De la même façon qu'au niveau syntaxique, les propriétés des objets sémantiques sont décrits par des traits polarisés mais il est nécessaire d'assouplir le système de polarités. Au niveau syntaxique, une polarité positive doit se combiner exactement avec une polarité négative. Au niveau sémantique, il semble nécessaire de disposer de polarités avec ordre de multiplicité pour représenter la sémantique de phrases telles que *Jean qui chante réussit en travaillant* où *Jean* est en même temps l'argument de trois prédicats *chanter*, *réussir* et *travailler*. On pourrait comme S. Kahane [KAH 04] introduire une polarité absorbante qui tout en étant neutre se combine avec un nombre quelconque de polarités négatives pour donner encore une polarité absorbante. Pour contrôler plus finement les ressources utilisées, il semble préférable d'utiliser le groupe additif des entiers relatifs comme système de polarités. Ainsi, nous utiliserons trois types de traits : des traits neutres  $t = v$ , des traits positifs  $t \overset{n}{\succ} v$  et des traits négatifs  $t \overset{n}{\prec} v$ , avec dans les deux derniers cas  $n$  entier strictement positif indiquant l'ordre de multiplicité de la polarité correspondante (s'il est égal à 1, on pourra l'omettre).

Pour simplifier la présentation, nous considérerons dans les exemples que nous traitons seulement deux traits sémantiques : le trait *type* qui représente le type de l'objet sémantique et qui peut prendre les valeurs *ent* (entité ou individu), *lex* (prédicat lexical), *quant* (quantificateur), ou *logic* (connecteur logique)<sup>8</sup> et le trait *cont* qui exprime la valeur sémantique de l'objet.

8. La valeur *pred* est une abréviation pour  $\{lex, quant, logic\}$ .



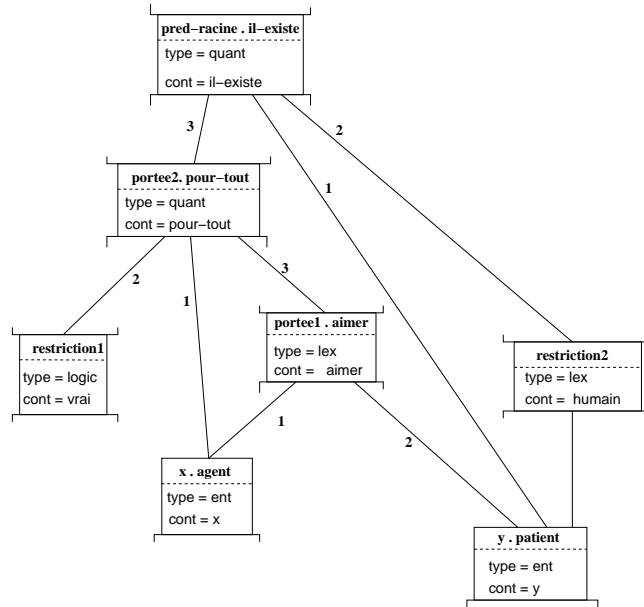
**Figure 3.** DAG sémantique représentant une première interprétation de la phrase tous aiment quelqu'un

## 2.2. Modèles neutres et minimaux de descriptions de DAG sémantiques

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.

La propriété de minimalité des modèles définie au niveau syntaxique s'applique exactement de la même façon au niveau sémantique et la propriété de neutralité doit être adaptée pour tenir compte du nouveau système de polarités avec ordre de multiplicité : pour un trait donné, la somme des polarités des traits qui s'interprètent dans un même trait du modèle doit être égale à 0. La description  $D_{sem}$  de la figure 1 a exactement deux modèles neutres et minimaux : les DAG des figures 3 et 4 qui expriment les deux relations possibles entre les portées des deux quantificateurs : le DAG de la figure 3 exprime le fait que c'est le quantificateur universel qui a la portée la plus large et celui de la figure 4 que c'est le quantificateur existentiel. Chacun de ces DAG représente une formule logique qui peut être retrouvée automatiquement. 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





**Figure 4.** DAG sémantique représentant une seconde interprétation de la phrase tous aiment quelqu'un

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  $\forall x (\exists y (\text{humain}(y) \wedge \text{aimer}(x,y)))$  et  $\exists y (\text{humain}(y) \wedge \forall x \text{aimer}(x,y))$  correspondant aux deux lectures possibles de la phrase *tous aiment quelqu'un*.

Comme au niveau syntaxique, l'opération de neutralisation des polarités fournit une méthode de calcul des modèles neutres et minimaux d'une description de DAG sémantique. Par exemple, les DAG des figures 3 et 4 s'obtiennent à la suite de 5 neutralisations opérées sur la description  $D_{sem}$  de la figure 1. Maintenant, il n'est pas toujours utile de calculer tous les modèles d'une description de DAG sémantique donné, surtout que ce calcul peut être très coûteux s'il y a multiplication des relations de portée au sein d'une même phrase. C'est d'ailleurs tout l'intérêt des langages permettant d'exprimer la sous-spécification.

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. S'il est un langage qui s'en rapproche, c'est la *Hole Semantics* [BOS 95] : 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 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 que dans l'autre sens, certains nœuds sémantiques, tels que les prédicats représentant les quantificateurs, n'ont aucun antécédent syntaxique. La fonction de liage n'est donc pas nécessairement surjective et même si cela n'apparaît pas dans notre exemple, elle n'est pas non plus forcément injective.

Les objets que nous allons maintenant manipuler sont des triplets  $(D_{syn}, D_{sem}, L)$ , que nous appellerons *descriptions SynSem* et qui sont définis ainsi :

**Definition 3.1** Une description SynSem est un triplet  $(D_{syn}, D_{sem}, L)$  tel que :

- $D_{syn}$  est une description d'arbre syntaxique ;
- $D_{sem}$  est une description de DAG sémantique ;
- $L$  est une fonction de liage qui associe tout nœud de  $D_{syn}$  à au plus un nœud de  $D_{sem}$ .

La définition d'un modèle minimal et neutre d'une description SynSem découle naturellement des définitions correspondantes au niveau syntaxique et au niveau sémantique.

Les IG étant lexicalisées, la fonction de liage entre niveau syntaxique et niveau sémantique est définie dans le lexique. Voyons comment en précisant ce qu'est une IG particulière.

**Definition 3.2** Une grammaire d'interaction  $G$  est un couple  $(Ax_G, Lex_G)$  tel que :

- $Ax_G$  est une description SynSem particulière appelée l'axiome de  $G$ . Elle se réduit à un unique nœud syntaxique lié à un unique nœud sémantique.

–  $Lex_G$  est une fonction qui associe chaque mot  $w$  de la langue à un ensemble fini de couples  $(D_w, Anc_w)$  tel que  $D_w$  est une description SynSem et  $Anc_w$  est un nœud distingué de la partie syntaxique de  $D_w$  appelé l'ancre.  $Lex_G$  est le lexique de  $G$ .

L'axiome  $Ax_G$  de la grammaire est la description SynSem exprimant le résultat attendu de l'analyse d'une phrase. Par exemple, dans la figure 1, l'axiome de la grammaire correspondante est constitué du nœud syntaxique *phrase* lié au nœud sémantique *pred-racine*.

Par ailleurs, dans une entrée  $(D_w, Anc_w)$  du lexique  $Lex_G$ , l'ancre  $Anc_w$  représente la position du mot  $w$  dans l'arbre sous-spécifié représenté au niveau syntaxique de  $D_w$ . Dans la figure 1, les ancres sont représentées par des rectangles en grisé.

Analyser une phrase  $w_1.w_2 \dots w_n$  avec une IG consiste tout d'abord à sélectionner une entrée  $(D_i, Anc_i)$  dans le lexique pour chacun des mots  $w_i$  de la phrase. Ensuite, nous effectuons l'union disjointe  $D_0 = \uplus_{i=1}^n D_i$  des descriptions sélectionnées<sup>9</sup>. La description  $D_0$  doit être complétée par des relations de précédence entre les ancres  $Anc_i$  exprimant l'ordre des mots dans la phrase et par l'axiome  $Ax_G$ . Il en résulte une description  $D$  qui représente une spécification de la phrase  $w_1.w_2 \dots w_n$  à l'aide de la grammaire  $G$ , telle celle que l'on peut voir à la figure 1, qui constitue le point de départ de l'analyse.

Le problème maintenant est de calculer tous les modèles neutres et minimaux de  $D$ . Chaque modèle a deux versants : un versant syntaxique qui constitue l'arbre d'analyse syntaxique de la phrase et un versant sémantique qui en donne une représentation 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 la partie syntaxique  $D_{syn}$ , jusqu'à l'obtention d'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 la partie sémantique  $D_{sem}$  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.

À 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, selon les neutralisations effectuées, les deux DAG sémantiques des figures 3 et 4.

9. Si deux éléments  $D_i$  et  $D_j$  participant à cette union ont des nœuds qui ont le même nom ou des valeurs de traits qui portent le même indice, nous effectuons un renommage des nœuds et indices problématiques.

Cet exemple est très simple mais le lecteur trouvera des exemples de modélisation de phénomènes linguistiques plus complexes dans [PER 03]. Par exemple, on y montre une seule entrée lexicale pour le pronom relatif *que*, qu'il introduise une proposition relative appositive ou une proposition relative restrictive ; les deux usages donnent lieu au même modèle syntaxique mais à deux modèles sémantiques.

D'un point de vue théorique, un tel processus d'analyse est extrêmement coûteux mais, comme nous le montrons dans [BON 03], 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<sup>10</sup> qui intègre la sémantique et qui est téléchargeable librement avec une grammaire couvrant la plus grande partie et un lexique de 400 lemmes et un millier de formes fléchies couvrant la totalité de la suite de phrases tests de la TSNLP pour le français [LEH 96].

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

Les IG présentent certaines similarités avec les TAG synchrones [SHI 90, SHI 94]. 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 [RAM 96].

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 [CAN 98], ce qui a d'ailleurs amené à l'introduction de nouveaux formalismes [RAM 95]. 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 [GAR 03] 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 95]. L'interface syntaxe-sémantique utilise en plus des traits associés aux nœuds des arbres syntaxiques dérivés et qui indiquent avec quels individus 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.

---

10. L'URL où le logiciel peut être téléchargé est : [www.loria.fr/equipes/calligramme/leopard](http://www.loria.fr/equipes/calligramme/leopard).

Du côté des CG, de Groote [GRO 01] et Muskens [MUS 03] 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 de 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 [KAH 02], les *Grammaires d'Unification Sens-Texte (GUST)*. Suivant la théorie Sens-Texte [MEL 88], 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 [KAH 04].

L'intégration de la *Minimal Recursion Semantics* dans HPSG [COP 99] 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 tant au niveau sémantique que syntaxique est guidé par un système de polarités et l'interface syntaxe-sémantique est

souple et simple : elle est assurée par une simple fonction du niveau syntaxique vers le niveau sémantique.

Une façon de démontrer la pertinence du formalisme proposé est de la faire de façon expérimentale. Nous proposons pour cela d'utiliser l'analyseur syntaxique LEOPAR précédemment mentionné. Ce dernier a été conçu au départ pour la syntaxe des langues mais le niveau sémantique peut y être ajouté avec des modifications mineures (il suffit d'enrichir les structures de traits par des traits sémantiques). Ce qui demandera plus de travail est d'enrichir la grammaire et le lexique du français qu'utilise actuellement LEOPAR avec la sémantique.

## 5. Bibliographie

- [BLA 01] BLACHE P., *Les Grammaires de Propriétés : des contraintes pour le traitement automatique des langues naturelles*, Hermès Sciences, 2001.
- [BON 03] BONFANTE G., GUILLAUME B., PERRIER G., « Analyse syntaxique électrostatique », *Traitement Automatique des Langues*, vol. 44, n° 3, 2003, p. 93-120.
- [BOS 95] BOS J., « Predicate Logic unplugged », DEKKER P., STOKHOF M., Eds., *10th Amsterdam Colloquium*, 1995, p. 133-142.
- [CAN 98] CANDITO M.-H., KAHANE S., « Can the derivation tree represent a semantic graph ? An answer in the light of Meaning-Text Theory », *TAG+4, Philadelphia*, 1998, p. 21-24.
- [CAR 98] CARPENTER B., *Type-logical Semantics*, MIT Press, Cambridge, Massachusetts, 1998.
- [CHU 40] CHURCH A., « A formulation of the simple theory of types », *Journal of Symbolic Logic*, vol. 5, 1940, p. 56-68.
- [COP 99] COPESTAKE A., FLICKINGER D., SAG I., « Minimal Recursion Semantics - an Introduction », Draft, 1999.
- [EGG 98] EGG M., NIEHREN J., RUHRBERG P., XU F., « 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*, 1998.
- [GAR 03] GARDENT C., KALLMEYER L., « Semantic construction in FTAG », *EACL'2003, Budapest, Hungary*, 2003.
- [GRO 01] DE GROOTE P., « Towards Abstract Categorical Grammars », *Association for Computational Linguistics, 39th Annual Meeting and 10th Conference of the European Chapter, Toulouse, France*, 2001, p. 148-155.
- [KAH 02] KAHANE S., « Grammaire d'Unification Sens-Texte - Vers un modèle mathématique articulé de la langue », *Habilitation à diriger des recherches*, Université Paris 7, 2002.
- [KAH 04] KAHANE S., « Grammaires d'unification polarisées », *11ième Conférence annuelle sur le Traitement Automatique des Langues Naturelles (TALN'04), Fès, Maroc, France*, 2004.
- [LEH 96] LEHMANN S., OEPEN S., REGNIER-PROST S., NETTER K., LUX V., KLEIN J., FALKEDAL K., FOUVRY F., ESTIVAL D., DAUPHIN E., COMPAGNION H., BAUR J., BAL-

- KAN L., ARNOLD D., « TSNLP — Test Suites for Natural Language Processing », *Proceedings of COLING 1996, Copenhagen*, 1996.
- [MEL 88] MEL'CUK I., *Dependency Syntax : Theory and Practice*, Albany, N.Y. : The SUNY Press, 1988.
- [MON 70] MONTAGUE R., « Universal grammar », *Theoria*, vol. 36, 1970, p. 373-398, Reprinted in R. Thomason, editor, *Formal Philosophy*, 188-221, New Haven : Yale University Press.
- [MUS 03] MUSKENS R., « Lambda Grammars », *Prospects and Advances in the Syntax/Semantics Interface, Lorraine-Saarland Workshop Series, Nancy*, 2003, p. 29-32.
- [PER 02] PERRIER G., « Descriptions d'arbres avec polarités : les Grammaires d'Interaction », *9ième Conférence annuelle sur le Traitement Automatique des Langues Naturelles (TALN'02), Nancy, France, 2002*, 2002.
- [PER 03] PERRIER G., « Les grammaires d'interaction », Habilitation à diriger des recherches, Université Nancy2, 2003, Film de la soutenance visible à l'URL : <http://www.inria.fr/multimedia/Didactheque-fra.html>.
- [RAM 95] RAMBOW O., VIJAY-SHANKER K., WEIR D., « D-Tree Grammars », *ACL'95, Cambridge, USA*, 1995, p. 151-158.
- [RAM 96] RAMBOW O., SATTI G., « Synchronous Models of Language », *ACL'96, Santa Cruz*, 1996.
- [REY 93] REYLE U., « Dealing with ambiguities by underspecification : Construction, Representation and Deduction », *Journal of Semantics*, vol. 10, 1993, p. 123-179.
- [SHI 90] SHIEBER S., SCHABES Y., « Synchronous Tree-Adjoining Grammars », *CoLing'90, Helsinki*, vol. 3, 1990, p. 1-6.
- [SHI 94] SHIEBER S., « Restricting the Weak-Generative Capacity of Synchronous Tree-Adjoining Grammars », *Computational Intelligence*, vol. 10, n° 4, 1994, p. 371-385.
- [STA 97] STABLER E., « Derivational Minimalism », RETORÉ C., Ed., *Logical Aspects of Computational Linguistics, LACL'96, Nancy, France, September 1996*, vol. 1328 de *Lecture Notes in Computer Science*, 1997, p. 68-95.
- [TES 34] TESNIÈRE L., « Comment construire une syntaxe », *Bulletin de la Faculté des Lettres de Strasbourg*, vol. 7 - 12<sup>ième</sup> année, 1934, p. 219-229.
- [VIJ 92] VIJAY-SHANKAR K., « Using Description of Trees in a Tree Adjoining Grammar », *Computational Linguistics*, vol. 18, n° 4, 1992, p. 481-517.