

Le rôle des contraintes dans les théories linguistiques et leur intérêt pour l'analyse automatique : les Grammaires de Propriétés

Philippe Blache

LPL-CNRS, Université de Provence
29 Avenue Robert Schuman
13621 Aix-en-Provence, France
pb@lpl.univ-aix.fr

1. Introduction

Tous les formalismes linguistiques font usage de la notion de contrainte qui, dans son sens le plus large, indique une propriété devant être satisfaite. Les contraintes sont extrêmement utiles à la fois pour représenter l'information linguistique, mais également pour en contrôler le processus d'analyse¹. Cependant, l'usage qui est fait des contraintes peut être très différent d'une approche à l'autre : dans certains cas, il s'agit simplement d'un mécanisme d'appoint, dans d'autres, les contraintes sont au cœur de la théorie.

Il existe cependant un certain nombre de restrictions à leur utilisation, en particulier pour ce qui concerne leur implantation. Plus précisément, s'il semble naturel (au moins dans certains paradigmes) de considérer l'analyse syntaxique comme un problème de satisfaction de contraintes, on constate cependant qu'il est extrêmement difficile de réaliser concrètement une telle implantation. Ce constat est en fait révélateur d'un problème dépassant le simple cadre de l'implémentation : nous montrons dans cet article qu'une approche totalement basée sur les contraintes (permettant donc de concevoir l'analyse comme un problème de satisfaction) est incompatible avec une interprétation générative classique accordant un statut particulier à la relation de dominance. Nous proposons ici un cadre permettant à la fois de tirer parti des avantages des grammaires syntagmatiques tout en s'affranchissant des problèmes liés aux approches génératives pour ce qui concerne l'usage des contraintes en tant qu'unique composant grammatical. Nous présentons ici cette approche, les *Grammaires de Propriétés*, ainsi que leur implémentation.

2. Les contraintes dans les théories, leur utilité

L'utilisation de contraintes est désormais systématique dans les théories linguistiques. Les exemples qui suivent illustrent leur utilisation dans différents paradigmes mettant les contraintes

1. On remarque d'ailleurs que cette double fonction (représentation des connaissances et contrôle) est également explicite pour la programmation par contraintes.

au cœur de l'analyse à la fois en termes de représentation des connaissances et de traitement.

– *Les Grammaires de contraintes* (cf. (Karlsson90)) :

Dans cette approche, les contraintes sont utilisées pour filtrer en fonction d'informations contextuelles les caractéristiques d'une catégorie. Les contraintes peuvent porter sur la catégorisation elle-même ou sur d'autres types d'information selon le niveau de spécificité des traits utilisés. L'exemple suivant donne un ensemble de contraintes pour l'anglais :

1. (@w = 0 "PREP" (-1 DET))
2. (@w = 0 VFIN (-1 TO))
3. ("that" =! "< Rel >" (-1 NOMHEAD) (1 VFIN))

La première contrainte indique qu'une préposition ne peut être précédée d'un déterminant, la seconde que la particule *to* ne peut précéder un verbe tensé, la dernière indiquant quant à elle que *that* est un relatif s'il suit un tête nominale et précède un verbe tensé.

– *Les Grammaires de Dépendance par Contraintes* (cf. (Maruyama90)) :

Les contraintes dans ce cas s'appliquent aux relations de dépendances. Elles permettent d'en contrôler la construction en établissant un lien entre le type de relation de dépendance et les caractéristiques des mots reliés.

1. $word(pos(x))=SP \Rightarrow (word(mod(x)) \in \{SP, SN, V\}, mod(x) \prec pos(x))$
2. $mod(x) \prec pos(y) \prec pos(x) \Rightarrow mod(x) \preceq mod(y) \preceq pos(x)$

La première contrainte indique qu'un *SP* peut modifier un autre *SP*, un *SN* ou un *V* et que ces éléments le précèdent. La deuxième contrainte implémente la projectivité en interdisant le croisement des relations de modification.

– *HPSG* (cf. (Pollard94)) :

Toute information linguistique en HPSG est considérée comme contrainte. Elles s'appliquent sur les structures de traits construites en cours d'analyse.

$$\left[\begin{array}{l} \text{SYNSEM} \left[\text{SYN} \left[\text{COMPS} \langle \rangle \right] \right] \\ \text{HD-DTR} \left[\begin{array}{l} \text{word} \\ \text{SS} \left[\text{SYN} \left[\text{COMPS} \langle \boxed{1} \dots \boxed{n} \rangle \right] \right] \end{array} \right] \\ \text{NHD-DTRS} \langle \left[\text{SS} \boxed{1} \right], \dots, \left[\text{SS} \boxed{n} \right] \rangle \end{array} \right]$$

Cette contrainte, portant sur les syntagmes à complément, indique que les compléments d'une tête doivent apparaître en tant que fils non-tête du signe syntagmatique.

Il est clair que du point de vue linguistique, l'utilisation systématique de contraintes est extrêmement intéressante. Elles permettent en effet de spécifier des informations de différents niveaux mais offrent de plus un cadre général au problème de l'analyse linguistique en optant pour une vision incrémentale pour laquelle nombre d'arguments psycholinguistiques sont avancés. Une approche reposant sur les contraintes offre de plus un cadre opérationnel, celui de la satisfaction de contraintes. Il s'agit en soi d'une propriété intéressante qui, comme le soulignent les auteurs dans (Johnson99) (p. 67), permet une connexion directe entre la théorie de la grammaire utilisée et son interprétation informatique. Les mêmes concepts sont mis en œuvre aussi bien du point de vue linguistique qu'informatique. De plus, la satisfaction de contrainte permet d'envisager la manipulation de données incomplètes, ce qui est d'une grande importance

du point de vue de la robustesse. Il devient ainsi possible de construire des solutions partielles ou approchées.

C'est ainsi que la plupart des approches contemporaines prétendent (souvent à tort) effectivement implémenter le problème de l'analyse syntaxique comme un problème de satisfaction de contraintes. Sans entrer dans les détails, une telle approche consiste à spécifier un ensemble de variables, leurs domaines ainsi qu'un système de contraintes portant sur ces variables. La satisfaction de contrainte consiste à vérifier en permanence la consistance de ce système ce qui permet de réduire au fur et à mesure du traitement les domaines de définition des variables, offrant un meilleur contrôle des processus. Nous verrons dans la section suivante qu'un des problèmes majeurs réside dans la spécification des variables contraintes : dans la plupart des théories linguistiques, elles représentent des structures de haut niveau, ce qui empêche un accès immédiat à leurs valeurs.

3. Le niveau des contraintes dans les théories

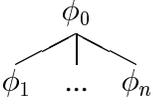
Les exemples donnés dans la section précédente reposent sur des théories faisant explicitement référence à la notion de contrainte. Il est intéressant d'examiner ce qu'il en est par rapport à d'autres approches.

On trouve dans (Johnson99) un analyse intéressante de la notion de contrainte et de l'usage qui en est fait par le programme minimaliste (cf. (Chomsky95)) d'un côté et les théories basées sur les contraintes, en particulier HPSG (cf. (Pollard94) et (Sag99)) de l'autre. Une distinction peut ainsi être établie entre les méthodes séquentielles, reposant sur des systèmes de règles et construisant séparément différents niveaux de représentation, et les méthodes incrémentales avec lesquelles l'analyse est conçue comme un processus parallèle. Le programme minimaliste se situe dans le premier cas. Il s'agit dans ce cadre de sélectionner parmi les dérivations convergentes (i.e. dérivation générant une structure $\langle \text{forme phonologique}, \text{forme logique} \rangle$ bien formée, cf (Chomsky95), p.171) la dérivation optimale. Dans ce cas, la satisfaction de contraintes est nécessaire, mais pas suffisante pour la grammaticalité. Dans les approches incrémentales, typiquement les théories basées sur les contraintes, les contraintes forment un véritable système. Cette idée qui consiste à préférer un ensemble de contraintes interagissant entre elles (cf. en particulier (Carpenter92) et (Pollard96)) à la place d'une approche basée sur les règles est fondamentale dans une vision parallèle de l'analyse. Elle permet de plus, comme le soulignent les auteurs dans (Johnson99) une connection plus directe entre la théorie de la grammaire et l'implantation.

3.1. Le cas de HPSG

HPSG s'inscrit totalement dans cette perspective et la notion de grammaticalité se décrit en termes de satisfaction de contraintes (cf. (Sag99), p.144). On retrouve dans cette définition l'idée que toute information linguistique, quel que soit son niveau, joue le rôle de contrainte :

- *Bonne formation* : Φ est une structure arborescente bien formée ssi tous les sous-arbres de Φ satisfont une entrée lexicale ou une règle de grammaire ρ .
- *Satisfaction lexicale* : la structure lexicale $\begin{matrix} F \\ | \\ \omega \end{matrix}$ satisfait l'entrée lexicale $\langle \omega, \delta \rangle$ seulement si F satisfait δ .

- *Satisfaction syntagmatique* : le sous-arbre local $\Phi =$  satisfait une règle de grammaire $\rho = \delta_0 \rightarrow \delta_1 \dots \delta_n$ ssi :

1. la séquence $\langle \phi_0, \phi_1, \dots, \phi_n \rangle$ satisfait la description $\langle \delta_0, \delta_1, \dots, \delta_n \rangle$,
2. Φ satisfait le Principe de Compositionnalité Sémantique,
3. si ρ est une règle portant sur les têtes, alors Φ satisfait le Principe des Traits de Tête le Principe de Valence et le Principe d'Héritage Sémantique.

Dans cette caractérisation de la bonne formation, toute information linguistique, qu'il s'agisse de description de structures lexicales ou de principes universels, joue le rôle de contraintes. La question est alors de savoir sur quels objets portent les contraintes. Plus précisément, on observe que la satisfaction est exprimée sur des arbres. Cela signifie qu'une contrainte ne peut être vérifiée qu'après avoir construit un arbre local. En HPSG, cela consistera à sélectionner une règle de la grammaire², correspondant à un schéma d'arbre puis à instancier cet arbre en fonction des diverses informations disponibles. On a donc un processus de "generate-and-test" induisant une utilisation passive de contraintes ; le domaine de définition des variables ne peut être réduit *a priori* par le système de contraintes.

Cette limitation vient en HPSG du fait que toute l'information est dépendante de la connaissance des têtes : la vérification de restriction de sélection entre deux unités lexicales, par exemple, ne peut se faire directement mais doit remonter à la racine de l'arbre local puis redescendre vers l'objet contrôlé. La racine d'un arbre local ne pouvant être construite que par projection de la tête, il est donc indispensable de préparer le schéma de l'arbre contenant la tête et la forme générale des autres constituants : ce mécanisme repose essentiellement sur trois règles (règle *Tête-complément*, règle *Tête-spécifieur* et règle *Tête-modifieur*). La motivation fondamentale de ces règles, est donc de préparer la structure permettant de spécifier les objets linguistiques sur lesquels les contraintes pourront être vérifiées. Ces schémas de règles sont donc en nombre réduit, mais indispensables car permettant de construire l'information pré-requise pour le processus de satisfaction. Il s'agit d'une limitation dans la mesure où les variables contraintes (les structures syntaxiques) ne sont pas directement accessibles.

3.2. Grammaires de dépendance et contraintes

De leur côté, les théories non génératives, et en particulier les grammaires de dépendance, s'appuient sur un idée somme toute proche de celle des théories basées sur les contraintes pour ce qui concerne la représentation des données linguistiques consistant à considérer l'information sous la forme d'un ensemble d'équations (cf. (Mel'čuk88), p.45). L'utilisation des contraintes dans les grammaires de dépendances (cf. (Maruyama90), (Duchier99)) tire parti de cette caractéristique en proposant une interprétation de l'analyse totalement basée sur les contraintes.

Des travaux récents ont systématisé ces propositions notamment autour du projet DAWAI (cf. (Foth00), (Schroder00)). Comme cela a été montré dans la section précédente, le premier problème à traiter concerne l'accessibilité des valeurs des variables contraintes, condition *sine qua non* pour une interprétation du processus d'analyse comme un problème de satisfaction de contrainte. C'est une des raisons pour lesquelles les auteurs de (Foth00) indiquent que les théories basées sur les contraintes ne correspondent pas à un problème de satisfaction de contraintes.

2. On peut trouver un exemple de grammaire HPSG récapitulant notamment ces règles dans (Sag99), p.401).

La méthode proposée par les CDG consiste à générer sans contrôle l'ensemble des relations possibles pouvant relier deux mots, le système de contraintes étant utilisé pour filtrer le réseau ainsi obtenu. Les contraintes en CDG (cf. exemple de la première section) sont exprimées sur une relation de dépendance. Le type de la relation n'est pas spécifié ce qui permet dans un premier temps de générer toutes les relations possibles entre deux mots situés à une position quelconque de la phrase. Le domaine des variables contraintes est formé par cet ensemble de relations. Mais, de la même façon que les têtes sont un pré-requis en HPSG, les contraintes ne peuvent être vérifiées que dans une relation de dépendance donnée, il n'est pas possible de spécifier une relation entre deux mots sans la définir par rapport à une dépendance. Pratiquement, ceci motive dans cette approche la restriction à des contraintes binaires.

4. Les Grammaires de Propriétés

La difficulté rencontrée dans les approches basées sur les contraintes réside donc dans le fait que les contraintes portent sur des informations structurées d'un niveau non élémentaire. Dans ce cas, l'ensemble de contraintes ne peut fonctionner comme un système global : la satisfaisabilité ne peut être vérifiée que sur un sous-ensemble de variables structuré (en HPSG un arbre local et en CDG une relation de dépendance). Une approche s'appuyant effectivement sur la satisfaction de contraintes doit pouvoir accéder directement aux valeurs des variables, sans passer par la construction d'une structure élémentaire.

Nous proposons une nouvelle approche, les *Grammaires de Propriétés* (cf. (Bès99), (Blache99) ou (Blache00)), allant dans cette direction en proposant de spécifier la totalité de l'information syntaxique directement sur les catégories (et non pas sur les structures). Les *Grammaires de propriétés* (notées dorénavant *GP*) se situent dans le paradigme de grammaires syntagmatiques et préservent la notion de structure syntaxique hiérarchisée.

Nous donnons dans cette section une présentation des éléments essentiels de ce formalisme et de son fonctionnement.

4.1. Les catégories

Les catégories en *GP* sont formées par une structure de traits comportant les informations susceptibles d'intervenir dans la spécification de contraintes. L'exemple suivant décrit une entrée verbale, avec ses traits sémantiques.

$$(1) \quad \left[\begin{array}{l} \text{PHON } \langle \textit{aimes} \rangle \\ \text{LABEL } V \\ \text{FORME } [\textit{tensé}] \\ \text{CAT } \left[\begin{array}{l} \text{ACC } [\text{NUMBER } \textit{sing}] \\ \text{PERS } 2 \end{array} \right] \\ \text{SEM } \left[\begin{array}{l} \text{REL } \textit{aimer} \\ \text{AGT } x \\ \text{THEME } y \end{array} \right] \end{array} \right]$$

Comme d'habitude dans ce genre de représentation, il est toujours possible de compléter la structure par de nouveaux traits sans que cela n'affecte le processus d'analyse. On peut ainsi envisager la représentation de plusieurs niveaux de l'analyse (prosodie, syntaxe, sémantique) au sein d'une même structure.

A la différence des théories basées sur les contraintes, ces structures sont statiques dans le sens où elles ne contiennent aucune information opérationnelle (typiquement le partage de

structure). La question de l'interface entre les niveaux d'analyse évoqués plus haut concerne bien entendu les informations de base, mais est effectivement traitée par les propriétés.

4.2. Les Propriétés

Les propriétés sont des contraintes portant sur des ensembles de catégories. Toutes les propriétés sont au même niveau, c'est à dire qu'elles sont indépendantes les unes des autres et ne sont pas ordonnées entre elles. Chaque catégorie est associée à un sous-ensemble de propriétés. Dans cet article, nous ne parlerons que des propriétés syntaxiques, les catégories concernées sont donc celles de niveau syntagmatique. Mais d'autres propriétés peuvent concerner le niveau lexical (comme les propriétés morphologiques ou phonologiques). La liste de propriétés qui suit n'est donc pas exhaustive.

- **Constituance** (*Const*): Définit l'ensemble maximal de catégories pouvant apparaître dans une unité syntaxique.

Exemple: Const(SN) = {Det, SA, N, Sup, SP, Pro}

- **Obligation** (*Oblig*): Spécifie les noyaux. L'une de ces catégories (à l'exclusion des autres) doit être présente.

Exemple: Oblig(SV) = {N, Pro, SA}

- **Unicité** (*Unic*): Ensemble de catégories uniques dans un syntagme.

Exemple: Unic(SN) = {Det, N, SA, SP, Sup, Pro}

- **Exigence** (\Rightarrow): Cooccurrence entre des ensembles de catégories.

Exemple: {le, être_[acc]} \xRightarrow{VP} Clit_[refl,acc] (Je me le suis dit)

Si le clitique *le* et le verbe tensé *être* cooccurrent, alors un clitique réfléchi s'accordant avec le verbe est obligatoire.

- **Exclusion** (\nrightarrow): Restriction de cooccurrence entre des ensembles de catégories.

*Exemple: Clit_[refl] \nrightarrow lui^{VP} (*Je me lui dis)*

Dans un SV, un clitique réfléchi ne peut pas apparaître avec le clitique *lui*.

- **Linéarité** (\prec): Contraintes de précédence linéaire.
- **Dépendance** (\rightsquigarrow): Relations de dépendance entre les catégories.

Une grammaire en GP est formée par un ensemble de propriétés exprimant différentes relations entre les catégories formant la structure syntaxique. Ces propriétés peuvent être très spécifiques et concerner un ensemble limité de catégories ou au contraire très générales. L'exemple de la figure (1) présente une grammaire du SN illustrant le fonctionnement des différentes propriétés. Il s'agit bien entendu d'un exemple limité, on trouvera des exemples plus complets dans (Blache99).

On remarque dans cet exemple qu'il est possible de spécifier des catégories en indiquant des valeurs de trait, comme c'est le cas pour la contrainte d'exigence entre un nom commun et un

		<u>Propriétés du SA</u>
	(1) $Const = \{Det, N, SA, Sup\}$	(1) $Const = \{Adj, Adv\}$
	(2) $Oblig = \{N, SA\}$	(2) $Oblig = \{Adj\}$
	(3) $N[com] \Rightarrow Det$	(3) $Adv \prec Adj$
	(4) $Det \prec N$ (5) $Det \prec SA$	(4) $Adv \rightsquigarrow Adj$
<u>Propriétés du SN</u>	(6) $Det \prec Sup$ (7) $N \prec Sup$	<u>Propriétés du Sup</u>
	(8) $SA \not\prec Sup$	(1) $Const = \{Det, Adv, Adj\}$
	(9) $Det \rightsquigarrow N$ (10) $SA \rightsquigarrow N$	(2) $Oblig = \{Adj\}$
	(11) $Sup \rightsquigarrow N$	(3) $Adj \Rightarrow Det$
		(4) $Adj \Rightarrow Adv$
		(5) $Det \prec Adv$ (6) $Det \prec Adv$ (7) $Adv \prec Adj$
		(8) $Det \rightsquigarrow Adj$ (9) $Adv \rightsquigarrow Adj$

FIG. 1 – Un ensemble de propriétés pour le SN

déterminant. On remarquera également que si aucune contrainte spécifique n’est indiquée, une catégorie pourra apparaître ou pas. Ainsi, dans le cas de construction verbales, un verbe pourra être contruit de avec des valences différentes (intransitives ou transitives) par le même ensemble de propriétés.

On remarquera également qu’il est possible d’indiquer des propriétés sur des ensembles de catégories, permettant ainsi la représentation de contraintes contextuelles. Par ailleurs, l’unification permet de propager des valeurs de traits entre plusieurs catégories. Le premier exemple de (2) montre ainsi l’instanciation de la valeur du trait agent du *SV* par le contenu des traits sémantique du *SN* sujet. Le second exemple concerne un cas de contrôle et spécifie le partage du sujet entre le verbe tensé et le verbe à l’infinitif.

$$\begin{array}{ccc}
 \text{(2)} \quad \begin{array}{c} SN \\ \left[\begin{array}{c} CAS \textit{ nom} \\ SEM \left[\begin{array}{c} \square \end{array} \right] \end{array} \right] \rightsquigarrow SV \\ \left[\begin{array}{c} SEM \left[\begin{array}{c} AGT \left[\begin{array}{c} \square \end{array} \right] \end{array} \right] \end{array} \right]
 \end{array} & & V \left[\begin{array}{c} VFORM \textit{ inf} \\ SEM \left[\begin{array}{c} AGT \left[\begin{array}{c} \square \end{array} \right] \end{array} \right] \end{array} \right] \rightsquigarrow V \left[\begin{array}{c} VFORM \textit{ fin} \\ SEM \left[\begin{array}{c} AGT \left[\begin{array}{c} \square \end{array} \right] \end{array} \right] \end{array} \right]
 \end{array}$$

4.3. La satisfaction de contraintes

Comme nous l’avons vu, un problème de satisfaction de contraintes doit en premier lieu spécifier les variables et leurs domaines. L’accessibilité de ces valeurs est essentielle pour une utilisation active des contraintes. En *GP*, toutes les contraintes sont exprimées sur des catégories. Le principe consiste donc à déterminer en début de traitement l’ensemble des catégories nécessaire à l’analyse.

Une des particularités des *GP* est de représenter toutes les contraintes de façon indépendante, les informations contenues dans les catégories étant statiques. De plus, les contraintes sont regroupées en plusieurs systèmes également indépendants les uns des autres, chaque système caractérisant une catégorie. Le mécanisme d’analyse consiste donc à vérifier pour un ensemble de catégories sa satisfaisabilité sur l’ensemble des sous-sytèmes de contraintes. L’ensemble des catégories, nous allons le voir, peut être fourni d’emblée. Il s’agit donc simplement d’énumérer les sous-ensembles de catégories possibles et de vérifier pour chacun d’entre eux la consistance du système de contraintes correspondant.

Nous touchons là au cœur des *GP* : aucun processus de dérivation n’est utilisé, en d’autres termes, aucune règle syntagmatique ni schéma de règle n’est nécessaire pour calculer la structure syntaxique d’un énoncé. Il s’agit d’un aspect très important aussi bien d’un point de vue

formel que computationnel. On s'affranchit en effet du cadre génératif et de la relation particulière (de dérivation) existant entre grammaire et langage. Une des conséquences immédiates réside dans le fait que la notion de grammaticalité (liée à la dérivation) peut être remplacée par la notion plus générale de *caractérisation*. Il est en effet possible, quelle que soit la forme de l'énoncé, de fournir un ensemble de systèmes de contraintes le décrivant, avec les contraintes satisfaites et celles qui ne le sont pas. Une *caractérisation* est simplement formée par cet ensemble de systèmes. Précisons qu'aucune hiérarchisation des contraintes n'est nécessaire pour ce traitement, à la différence par exemple de la théorie de l'optimalité.

La souplesse des *GP* réside donc en particulier dans le fait qu'on spécifie un ensemble de systèmes de contraintes plutôt qu'un système unique. Il est donc possible de spécifier de façon simple et directe des contraintes contextuelles ou locales là où un système de contraintes unique nécessite l'ajout d'informations de contrôle permettant d'accéder à ces valeurs.

Nous décrivons rapidement dans le reste de cette section le mécanisme de détermination des variables contraintes et le calcul des caractérisations à partir de ces variables. Ce processus se décompose en plusieurs étapes.

Détermination des variables : La première étape consiste, pour une phrase donnée, à déterminer l'ensemble des catégories, de niveau lexical et syntagmatique, qui vont participer à la construction de la structure syntaxique correspondante. On applique pour cela une première étape classique de catégorisation (cf. exemple (1b)). Pour chacune des catégories lexicales ainsi spécifiées, on recherche la catégorie de niveau supérieur à laquelle elle peut appartenir. Ce type d'information (pouvant être prédéfini) est contenu dans les propriétés de constituance (cf. (1b)).

- (1) a. *le livre le plus vieux*
 b. $\Sigma = \{ Det_1, Pro_1, N_2, V_2, Det_3, Pro_3, Adv_4, Adj_5 \}$
 c. $\Sigma' = \{ Det_1, Pro_1, SN_1, Sup_1, N_2, SN_2, V_2, Det_3, Pro_3, SN_3, Sup_3, Adv_4, SA_4, Sup_4, Adj_5, SA_5, Sup_5, \}$

On remarquera que les catégories sont représentées par leur label auquel on associe un indice correspondant à la position du mot dans la phrase. Les ensembles sont construits sans contrôle particulier, l'objectif étant de fournir toutes les catégories pouvant participer à la description de la structure syntaxique.

Construction des suites de catégories : Cette seconde étape génère par simple énumération l'ensemble des suites de catégories possible. L'ensemble S est donc formé par l'ensemble des suites possibles de Σ' :

- (2) a. $S = \{ \{ Det_1, N_2 \}, \{ Det_1, SN_2 \}, \{ Det_1, N_2, Sup_3 \}, \dots \}$

Ces suites de catégories sont construites sans autre information linguistique que celle permettant de spécifier la catégorie dominante. Il est donc possible de construire pour tout type d'énoncé, et sans accéder aux informations grammaticales, l'ensemble d'objets (les variables) sur lesquels vont porter les contraintes.

Caractérisation des suites : Le processus d'analyse proprement dit consistera à calculer, pour chacune des suites, sa caractérisation. Il s'agit en d'autres termes de rechercher le système

de contraintes adéquat et à vérifier, pour la suite en question, sa satisfaisabilité. L'état du système après cette vérification (i.e. l'ensemble des contraintes satisfaites P^+ et l'ensemble des contraintes non satisfaites P^-) formera la caractérisation de la suite.

$$(3) \quad \begin{array}{ll} \text{Det}_1/\text{N}_2 : & P^+(\text{SN}) = \{1, 2, 3, 4, 10, 11\} \quad P^-(\text{SN}) = \emptyset \\ \text{Det}_1/\text{Sup}_2 : & P^+(\text{SN}) = \{1, 6, 10\} \quad P^-(\text{SN}) = \{2\} \\ \text{Adv}_4/\text{Adj}_5 : & P^+(\text{SA}) = \{1, 2, 3, 4\} \quad P^-(\text{SA}) = \emptyset \\ & P^+(\text{Sup}) = \{1, 2, 4, 7, 9\} \quad P^-(\text{Sup}) = \{3\} \end{array}$$

L'exemple (3) présente la caractérisation de quelques suites de S . Les ensembles P^+ et P^- sont formés par les indices des contraintes correspondantes (cf. figure (1)).

Une caractérisation ne contenant que des contraintes satisfaites correspondra à une suite grammaticale. La grammaticalité est donc un cas particulier de la caractérisation. La construction de la structure syntaxique se fera sur la base de ces ensembles de caractérisation. Il suffit en effet de rechercher un ensemble de suites de catégories caractérisées et couvrant la totalité de l'input. Il est bien entendu possible de contrôler ce processus en spécifiant un seuil maximum de contraintes non satisfaites. Si ce seuil est égal à zéro, on réduira les constructions possibles aux seules constructions grammaticales. Dans les autres cas, on pourra caractériser tout type de phrase, y compris celles correspondant à des structures ne satisfaisant pas toutes les propriétés de la grammaire (les phrases "non grammaticales").

Une implantation naïve de ce processus ne serait bien entendu pas efficace. En particulier, l'étape de construction des suites de catégories (qui revient à énumérer toutes les affectations possibles) a un coût exponentiel si elle n'est pas contrôlée. Il est donc nécessaire d'introduire des critères (par exemple la juxtaposition) qui pourront réduire de façon importante le nombre de suites créées. En revanche, le fait d'utiliser des catégories pouvant être sous-spécifiées ne pose pas de problème particulier pour la résolution elle-même. D'une façon plus générale, le nombre de variables est fini à condition de déterminer les valeurs de niveau non lexical en fonction des caractérisations déjà construites. En d'autres termes, une nouvelle variable correspondant à une catégorie non lexicale n'est introduite que si elle correspond à une suite caractérisée.

5. Conclusion

Les *Grammaires de Propriétés* constituent une alternative intéressante pour l'utilisation systématique de contraintes. Elles proposent en effet une solution aux restrictions des grammaires génératives basées sur les contraintes et qui requièrent la construction d'une structure locale avant de pouvoir utiliser les contraintes elles-mêmes. Les avantages des *GP* sont nombreux. Il s'agit tout d'abord d'un formalisme simple et permettant de représenter directement toutes les propriétés linguistiques requises pour la description d'une langue. De plus, la satisfaction de contrainte étant effectivement le seul mécanisme utilisé, il est possible de dépasser la notion de grammaticalité en proposant de fournir des descriptions syntaxiques pour tout type d'énoncé. Il s'agit donc d'une approche intrinsèquement robuste. Enfin, les contraintes peuvent être spécifiées sur tous les objets manipulés, mais il est également possible d'associer aux positions de la phrase tout type de description. Il est donc possible d'associer à un même énoncé les caractérisations de différents niveaux de l'analyse linguistique (prosodie, syntaxe, sémantique) sur des représentations pouvant être parallèles.

Références

- Archangeli D. & D.T. Langendoen (eds.) (1997) *Optimality Theory*, Blackwell.
- Bès G. & P. Blache (1999) "Propriétés et analyse d'un langage", in proceedings of *TALN'99*.
- Blache P. (1999) *Contraintes et théories linguistiques : des Grammaires d'Unification aux Grammaires de Propriétés*, Habilitation à diriger des recherches, Université Paris 7.
- Blache P. (2000) "Constraints, Linguistic Theories and Natural Language Processing", in proceedings of *NLP-2000*.
- Borsley R. (1996), *Modern Phrase Structure Grammars*, Blackwell.
- Carpenter B. (1992) *The Logic of Typed Feature Structures*, Cambridge University Press.
- Chomsky N. (1995) *The Minimalist Program*, MIT Press.
- Duchier D. & S. Thater (1999) "Parsing with Tree Descriptions: a constraint based approach", in proceedings of *NLULP'99*.
- Foth K., I. Schröder & W. Menzel (2000) "A Transformation-based Parsing Technique with Anytime Properties", in proceedings of *IWPT-2000*.
- Johnson D. & S. Lappin. (1999) *Local Constraints vs. Economy*, CSLI.
- Karlsson F. (1990), "Constraint Grammar as a Framework for Parsing Running Text", in Proceedings of *COLING'90*.
- Maruyama H. (1990), "Structural Disambiguation with Constraint Propagation", in proceedings of *COLING-ACL'98 workshop on Dependency-based Grammars*.
- Mel'čuk I. *Dependency syntax : theory and practice*, SUNY Press, Albany, 1988.
- Menzel W. & I. Schröder (1998), "Decision Procedures for Dependency Parsing Using Graded Constraints", in proceedings of *COLING-ACL'90*.
- Pollard C. & I. Sag (1994), *Head-driven Phrase Structure Grammars*, CSLI, Chicago University Press.
- Pollard C. (1996), "The Nature of Constraint-Based Grammar", PACLIC conference, reprinted in *Constructions: an HPSG Perspective*, ESSLLI'98 Lecture Notes.
- Sag I. & T. Wasow (1999), *Syntactic Theory. A Formal Introduction*, CSLI.
- I. Schröder, W. Menzel, K. Foth & M. Schulz (2000), "Modeling Dependency Grammars with Restricted Constraints", soumis à la revue *TAL*.
- Shieber S. (1992) *Constraint-Based Grammar Formalisms*, MIT Press.