Propriétés et analyse d'un langage

Gabriel G. Bès, Philippe Blache

Propriétés et analyse d'un langage

Gabriel Bès (1) et Philippe Blache (2)

(1) GRIL, Université de Clermont-Ferrand II, 34 Avenue Carnot, 63000 Clermont-Ferrand
(2) LPL-CRNS, Université de Provence, 29 Avenue Robert Schuman, 13621 Aix-en-Provence

Résumé

Nous présentons dans cet article une nouvelle approche, que nous appelons *5P*, permettant la description des propriétés d'un langage et son utilisation pour une analyse automatique. Nous montrons comment cette approche permet la prise en compte de la dimension descriptive de la linguistique. Par ailleurs, nous présentons une technique d'analyse, appelée analyse par *Filtrage et Fusion*, qui tire parti de cette description en propriétés. Nous montrons en quoi ces deux projets (description d'une langue et analyse automatique) convergent et ouvrent de nouvelles perspectives.

1. Introduction

Le projet de *décrire* une langue naturelle et de cerner ce qui lui est propre reste difficilement accessible aux théories linguistiques tout autant qu'aux approches descriptives non formelles et rare sont les travaux qui s'y attaquent (cf. les travaux autour de (Abeillé & al., en prép.) ou (Blanche-Benveniste & al. 91)).

Le projet *d'implanter* un système permettant le traitement automatique de descriptions des langues naturelles est, dans état actuel de la technologie, lui aussi difficilement accessible. Le problème qui se pose est celui de la relation inversement proportionnelle entre, d'une part, la couverture et la finesse de l'analyse et, d'autre part, sa robustesse et sa généralité.

Ces deux projets peuvent réussir à condition de les faire converger. Nous pensons que, si l'on veut vérifier les hypothèses proposées et les confronter avec des données observées, elles doivent être exprimées par des formulations déclaratives dont il est possible de calculer les conséquences. On parlera de *Propriétés* pour désigner les formulations censées décrire les énoncés d'une langue particulière, de *Projections* pour les formulations plus abstraites et générales déduites des Propriétés et de *Principes* pour les formulations d'encore plus grande généralité et qui sont censées contraindre les Propriétés des langues naturelles. Les déductions à partir des Propriétés doivent être confrontées à des *Protocoles*, c'est-à-dire à des données elles-mêmes observées et représentées de manière systématique et contrôlée. En résumé, l'ambition ultime est d'ancrer effectivement la linguistique parmi les sciences du réel.

Le projet informatique suppose l'implantation d'un système permettant à la fois une analyse détaillée tout en utilisant des mécanismes généraux. En d'autres termes, on souhaite obtenir

un système qui soit effectivement réutilisable et permette l'utilisation des *Processus* implantés pour des niveaux d'analyse différents (et donc pour des applications différentes).

Voici donc la convergence que nous supposons nécessaire à la réussite des deux projets : la prise en compte de la dimension descriptive et l'intégration effective de la linguistique parmi les sciences du réel doivent reposer sur la capacité d'exprimer les connaissances sous forme de Propriétés et de les valider en examinant leurs effets sur des données. La réutilisabilité d'un système informatique doit également reposer sur cette capacité d'isoler des Propriétés dont on peut régler la portée ou le niveau de précision.

Nous nous efforçons d'obtenir cette convergence dans un cadre que nous appelons *5P* (pour Protocoles, Propriétés, Projections, Principes et Processus. Nous ne présenterons pas ici l'architecture d'ensemble du Paradigme *5P* ni le flux d'information entre ces différents modules (cf. (Bès & al. 99)). Les sections qui suivent sont plutôt consacrées à la description des caractéristiques essentielles des Propriétés et leur exploitation pour une analyse linguistique modulaire qui prend de plus en charge le traitement de l'ambiguité.

1.1. Les Propriétés et leur exploitation

Les Propriétés sont des relations exprimées sur des catégories. Une catégorie est un ensemble de traits, chaque trait étant une étiquette associée à une seule valeur (pouvant être monadique ou une catégorie). Les catégories se définissent dans un système d'héritage multiple monotone.

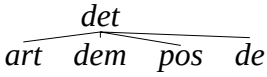
Etant données les catégories C_i et C_j, on dit que C_i subsume C_j si l'ensemble de traits de C_i est inclus dans C_j. C_i est une *catégorie maximale* s'il n'existe pas de C_j telle que C_i subsume C_j (avec $C_j \neq C_i$). Un lexique est un ensemble de signifiants, chaque signifiant étant associé à une ou plusieurs catégories maximales.

Prenons l'exemple, en français, de l'ensemble F formé par les démonstratifs, les articles définis, les articles indéfinis, les possessifs et la forme *de* (e.g.. "*de bonnes intentions*"). Les formes de F ont des Propriétés intéressantes, par exemple :

- S'il y a une des formes de F dans un *SNn* (syntagme nominal noyau), toutes les autres formes de F sont exclues de ce *SNn*.
- Selon les formes dans *SNn* qui n'appartiennent pas à F, pour que la suite *SNn* deviennne bien formée il suffit de la compléter par une des formes de F (ainsi, la suite *belles fleurs* devient bien formée en lui ajoutant une des formes *mes, ces, des, les* ou *de*).

Mais toutes les formes de F n'ont pas le même comportement par rapport aux autres formes dans le *SNn*. Ainsi si l'on a un démonstratif, on peut avoir *-ci* à droite de *fleurs* mais pas si on a un possessif (**mes fleurs-ci*). De même, si l'on n'a pas un adjectif on ne peut pas avoir *de*. Ou encore, si l'on a un cardinal, on ne peut pas avoir l'article indéfini (**des trois fleurs*).

On perçoit l'intérêt des catégories : en les utilisant on peut manipuler un ensemble de signifiants selon les Propriétés que l'on souhaite exprimer sur eux. La figure (1) décrit l'organisation hiérarchique (le treillis d'héritage) d'un ensemble de catégories héritant des propriétés du déterminant.



(1)

Supposons par ailleurs un trait booléen *def* (pour *défini*) caractérisant la catégorie *det*. Il est possible de spécifier la valeur de ce trait pour les catégories héritant de *det* : le trait [*def* +] sera spécifié pour les catégories *dem* et *pos* tandis que le trait [*def* -] sera spécifié pour la catégorie *de*. Avec cette organisation, la catégorie *det* désignera les articles, démonstratifs, possessifs et *de* tandis que la catégorie [*def* -] correspondra seulement aux articles indéfinis et à *de* et que la catégorie [*def* +] désignera les articles définis, démonstratifs et possessifs. On peut ainsi caractériser un certain comportement pour toutes les entrées associées à des catégories maximales subsumées par *det* et un autre comportement pour celles associées à des catégories maximales subsumées par [*def* -].

Les Propriétés relèvent de trois grands types : *existence, linéarité et fléchage*. On peut les caractériser à partir de l'idée qu'il est possible de décrire un énoncé en indiquant les pièces ultimes qui le composent (les catégories), leur ordre ainsi que les liens entre ces pièces.

Dans l'exemple suivant décrivant l'énoncé (1), les différents alinéas décrivent respectivement (1a) l'ensemble E des catégories associées aux signifiants, (1b) les Propriétés de précédence (relation notée $\prec$) et (1c) les Propriétés de fléchage. Dans les Propriétés ci-dessous, le noyau (cf. infra) est précédé du symbole +.

(1) *les belles fleurs*
- (1a) $E = \{art, adj, +n\}$
- (1b) $art \prec adj; adj \prec +n$
- (1c) $\{ \langle art \rightsquigarrow +n \rangle, \langle adj \rightsquigarrow +n \rangle \}$

On peut regrouper l'ensemble de ces informations au sein d'une même structure que nous appelons *modèle*. Dans notre exemple, étant donné que tous les éléments de E sont des catégories, on parlera de *modèle terminal*. En général :

Un *modèle* est un *triplet* de la forme $\langle L, F, ID \rangle$ où :

- L est un ensemble de couples $\langle S, i \rangle$ (où S est une catégorie ou un *ID* et i un entier représentant une position).
- F est un ensemble de couples $\langle i, j \rangle$ indiquant les relations de fléchage entre les positions i et j.
- *ID* est un identificateur du modèle décrit.

Un modèle (noté dorénavant *m-ID*) est donc une possibilité de réalisation associée à *ID*. On peut associer à tout modèle un ensemble E formé par les symboles qui instancient tous les S (les catégories et les identificateurs de l'ensemble l'ensemble L). On appelle cet ensemble E le *domaine* du modèle. On note E_{ID} le domaine du modèle *m-ID*. On note *M-ID* l'ensemble de modèles *m-id* qui satisfont les Propriétés d'existence, de linéarité et de fléchage de ID (notées *Propriétés-ID*). Les Propriétés-ID expriment de manière analytique et purement déclarative, factorisée et avec la granularité souhaitée, les contraintes qui doivent être satisfaites par tout *m-ID*. Ceci est possible car elles sont spécifiés séparémment et sans aucune influence sur la façon de les contrôler dans un processus d'analyse et/ou de génération. On développe ainsi jusqu'à ses dernières conséquences le concept de description analytique proposé dans GPSG et partiellement continué dans HPSG

- Les *Propriétés-ID d'existence* spécifient le plus petit ensemble E^+ tel que :
  $E^+ = \bigcup E_{ID}$ associés aux modèles de *M-ID*.

– Les *Propriétés-ID de linéarité* expriment les relations d'ordre telles que tout ensemble L d'un *m-ID* est susceptible d'être contraint par ces relations d'ordre appliquées au domaine E_{ID}.
– Les *Propriétés-ID de fléchage* expriment les dépendances entre les entités qui occupent les positions dans la liste. Ces dépendances permettent de calculer de manière compositionnelle la représentation sémantique.

Prenons un exemple :

(1) a. {les, ces, mes, des, de} belles fleurs
b. {les, ces, mes} trois belles fleurs
c. trois belles fleurs

La spécification des relations d'ordre de tous les modèles qui doivent être associés aux énoncés de l'exemple précédent se fait grâce aux *Propriétés-SNn* de linéarité exprimées de manière compacte dans : *det* $\prec$ *card* $\prec$ *adj* $\prec$ *n*.

Ces Propriétés le linéarité sont applicables à tous les domaines E_{SNn}. Evidemment, elles ne sont pas à elles seules suffisantes pour spécifier ces domaines. C'est aux Propriétés d'existence de définir E^+. Celles de linéarité ne disent pas ce qu'on peut trouver dans L. Une Propriété de linéarité telle que celle de l'exemple précédent doit se lire : si dans un domaine E_{SNn} il y a une entité subsumée par *det* et une subsumée par *card*, alors dans la liste L correspondante, l'entité subsumée par *det* précède l'entité subsumée par *card*. Mutatis mutandis, ce principe s'applique aussi aux Propriétés de fléchage.

Les Propriétés-ID spécifient *M-ID* avec la granularité souhaitée dans la mesure où on peut moduler l'organisation des catégories ainsi que l'expression des Propriétés qui les utilisent en fonction de ce que l'on veut exprimer. Par exemple (dans le sens de l'analyse) les Propriétés d'existence du *SNn* vont permettre à un premier niveau de définir comme éléments possibles de E^+ les ensembles {*det, adj, n, ...*}, {*det, n, ...*} et {*det, card, adj, n, ...*}. Si l'on veut plus de finesse, on pourra ajouter à ces Propriétés d'existence d'autres précisant par exemple que dans un même E on ne peut pas inclure [*def* -] et *card*, ou bien que si un E contient un *de* il doit également contenir un *adj*.

La spécification de E_{ID} par les Propriétés-ID d'existence permet de définir le domaine E de tout *m-ID* . Parmi ces pièces il y a des catégories et des ID. Par ailleurs, il y aura un symbole distingué, le noyau. Ainsi les énoncés *une belle fleur / des assez belles fleurs / une pas assez belle fleur* seront tous, parmi d'autres, associés au *m-SNn* suivant :

⟨ { ⟨ *art*, 1 ⟩, ⟨ SAn, 2 ⟩, ⟨ +*n*, 3 ⟩ }, {⟨ 1, 3 ⟩, ⟨ 2, 3 ⟩ }, *SNn* ⟩ .

Pour ce qui concerne le *SAn*, parmi les ensembles E_{SAn}, nous pourront trouver (on note *advc* l'adverbe comparatif) : {*advc, +adj*}, {*advc, pas, +adj*}, {*pas, +adj*} ou {*+adj*}. Ces ensembles permettront de spécifier respectivement *assez belle(s) / pas assez belle(s) / pas belle(s) / belle(s)*.

Puisque les éléments de E_{ID} sont des ensembles, la fonctionnalité essentielle des Propriétés d'existence est de spécifier les éléments de tout E_{ID}. Elles se formalisent par des opérations ensemblistes sur les vocabulaires utilisés (les catégories et les identificateurs de modèles) et en tenant compte de la relation de subsomption. Les Propriétés d'existence sont groupées dans les 5 types de stipulations suivantes :

- **Alphabet du modèle** : ensemble des catégories et identificateurs susceptibles pour un *ID* d'être utilisés dans tout E_{ID}.
  Exemple : *amod(SNn)* = {*det, card, n* })
- **Unicité** (unicité): indique les symboles ne pouvant apparaître qu'une fois dans un domaine.
  Exemple : *uniq(SNn)* = {det}. Toute catégorie subsumée par *det* ne peut apparaître qu'une fois dans un E_{SNn}.
- **Obligation** : exprime une disjonction exclusive des noyaux possibles. La réalisation d'un noyau (à l'exclusion des autres) dans un domaine est obligatoire.
  Exemple : *oblig(SNn)* = {*n, SAn*}. Le noyau d'un *m-SNn* est soit une catégorie *n* (p.ex. dans *la chaise*) soit un *m-SAn* (p.ex. dans *la très belle est partie*).
- **Exigence** : exprime l'exigence de cooccurrence entre plusieurs ensembles de symboles dans un E_{ID}.
  Exemple : {*n*} $\overset{SNn}{\Rightarrow}$ {{det}, {card}, W}. Si un E_{SNn} contient {n}, il doit également contenir soit {*det*}, soit {*card*}, soit {*det, card*} soit tout autre ensemble indiqué par W.
- **Exclusion** : exprime l'exclusion de cooccurrence entre plusieurs ensembles de symboles dans un E_{ID}.
  Exemple : {*card*} $\overset{SNn}{\not\Rightarrow}$ {[*def* -]}. Un cardinal et un déterminant indéfini ne peuvent coexister dans un E_{SNn}.

On suppose (cf. section suivante) que les stipulations *amod* et *uniq* donnent les informations minimales extraites des Propriétés d'existence pour pouvoir analyser les suites d'un langage. L'inspection de ces stipulations fournit les *Projections* les concernant. Ainsi on peut déduire des Propriétés que la catégorie *adj* est le noyau obligatoire d'un *m-SAn*, que ce *m-SAn* peut ou non être le noyau d'un *m-SNn*, que si le *m-SAn* est dans un *m-SNn* et qu'il n'est pas noyau, alors il y a un autre symbole noyau, et ainsi de suite. De même, on peut déduire des Propriétés de fléchage que dans un *m-SAn*, s'il y a *pas*, cette catégorie flèche sur le noyau ou sur un adverbe comparatif, que le *m-SAn* dans un *m-SNn*, s'il n'est pas le noyau, flèche sur le noyau, et ainsi de suite.

Ce sont les Projections calculées à partir des stipulations *amod* et *uniq* (Propriétés d'existence), des Propriétés de fléchage et de certaines Propriétés de linéarité, qui ont été utilisées par la suite pour analyser le langage décrit par ces Propriétés. On présente dans la section suivante un traitement possible à l'aide de ces Propriétés. Une autre approche est également décrite dans (Hagège98).

2. Filtrage et Fusion

L'analyse dans *5P* est un processus de propagation et de satisfaction de contraintes. La réutilisabilité du système vient précisément de cette caractéristique : les contraintes utilisées sont des Propriétés fournies dans *5P* et contiennent exclusivement des informations linguistiques. Rien dans *5P* ne présuppose une technique de calcul ou un mécanisme d'analyse particulier.

Nous présentons dans cette section les caractéristiques fondamentales de notre technique d'analyse appelée *filtrage et fusion* (notée *F&F*).

2.1. Organisation de l'information requise

La variabilité de la granularité de l'analyse repose sur la capacité à faire varier le sous-ensemble de Propriétés à vérifier. Il est donc indispensable de déterminer le niveau minimal

d'information en deçà duquel il n'est pas possible de donner des indications sur un énoncé.

Cet ensemble minimal est constitué des informations de dominance (décrites dans les alphabets de modèles), de dépendance (donnée par les Propriétés de fléchage) et d'ordre linéaire. Le second ensemble d'informations est constitué par toutes les autres Propriétés. Une granularité d'analyse plus ou moins fine sera obtenue en vérifiant tout ou partie de ce second ensemble de Propriétés.

2.2. Schéma général d'analyse

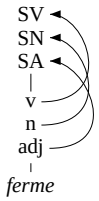
Utiliser différents niveaux d'information repose sur la capacité de les encoder et les traiter séparément. La technique d'analyse décrite ici distinguera donc les informations de dominance et de dépendance : la sortie de l'analyseur sera formée d'un *arbre* représentant la dominance et d'un ou plusieurs *graphes* pour la dépendance. La technique d'analyse par filtrage et fusion consistera donc *a minima* à créer et vérifier ces deux types d'information.

Par ailleurs la technique de vérification de Propriétés changera en fonction du type d'information traité : le filtrage concernera les informations de dépendance tandis que la fusion permettra de vérifier les Propriétés en fonction des informations de dominance. Nous sommes ainsi amenés à distinguer deux grandes étapes dans le traitement :

1. Elaboration des structures de base : préparation des structures de dominance élémentaires (les quasi-arbres unaires) et du réseau de dépendance.
2. Filtrage du réseau de dépendance et construction par fusion de la structure de dominance.

La construction des quasi-arbres unaires (dorénavant QAU) et celle du réseau de dépendance sont établies selon une technique décrite dans (Blache98). Rappelons-en les grandes lignes.

Quasi-arbres unaires : La première phase d'analyse consiste à associer à chaque entrée une structure de dominance élémentaire indiquant les Projections possibles de la catégorie maximale (ou des catégories maximales dans le cas d'assignations ambiguës) associée(s) à l'entrée. Ces Projections sont obtenues par inspection des *amod* contenant des catégories subsumant les catégories maximales associées aux entrées [1].



Les nœuds d'un QAU sont formés par une disjonction exclusive de valeurs (une valeur étant une catégorie ou un identificateur de modèle). Chacune de ces valeurs indiquera explicitement la valeur qui la domine dans le nœud supérieur. On dira qu'une valeur et celle qui la domine *covarient*. Un QAU est donc une structure de dominance encodant l'ambiguïté de catégorisation.

Réseau de dépendance : Les Propriétés de fléchage et de linéarité permettent dans un second temps d'établir l'ensemble de relations de dépendance possible entre les valeurs des QAU. La construction de ce graphe de dépendance se fait sans aucun contrôle : soit α et β deux valeurs présentes dans les QAU, s'il existe une relation de fléchage entre α et β (quelle que soit la distance les séparant dans la liste de QAU) alors une arrête est créée entre ces deux valeurs.

Le résultat est un graphe contenant bien entendu un très grand nombre d'arrêtes ne correspondant pas à une dépendance correcte. Il possède cependant une propriété essentielle : il

1. Soit α et β deux valeurs. Si il existe un α' subsumant α et tq $\alpha' \in$ *amod*(β), on dit que β est la projection de α.

contient au moins la totalité des relations de dépendances correspondant à une construction syntaxique décrite.

Filtrage du réseau : La première opération consiste donc à filtrer le réseau de dépendance ainsi construit. Ceci se fait par l'intermédiaire des *Principes* de *5P*. Pour une langue à morphologie faible comme le français, deux principes peuvent être évoqués : (*i*) deux relations de dépendances ne peuvent se croiser[2] et (*ii*) une même valeur ne peut dépendre de deux valeurs différentes.

Ces principes permettent de supprimer un grand nombre de relations et d'appliquer un premier filtrage. Cette technique consiste donc non pas à rechercher les bonnes relations de dépendance, mais à supprimer les incorrectes. La différence est évidemment fondamentale et la variabilité de la granularité de l'analyse en découle directement.

2.3. Fusion

Nous décrivons dans cette section la technique de *fusion* permettant, à partir des informations de dominance contenues dans les QAU et du réseau de dépendance filtré, de construire une structure de dominance hiérarchisée sous la forme d'un arbre. Le principe consiste, après avoir déterminé le point de fusion, à fusionner deux arbres en un seul. Cette opération doit respecter un certain nombre de conditions décrites ici.

1. **Choix des arbres à fusionner :** Deux arbres A et B sont candidats à la fusion si (*i*) ils sont juxtaposés et (*ii*) ils sont reliés par une relation de fléchage connectant certaines de leurs valeurs.
2. **Filtrage des arbres à fusionner par covariation :** Les deux arbres A et B précédemment sélectionnés sont filtrés de façon à ne conserver que les valeurs covariant avec des valeurs reliées par une relation de fléchage.
3. **Niveau de fusion :** Le niveau de fusion est constitué par les nœuds dans ces arbres qui vont être fusionnés : *les nœuds à fusionner sont ceux dominant immédiatement les valeurs reliées par une relation de fléchage.* Dans le cas de QAU, les nœuds à fusionner sont toujours les racines.
4. **Fusion de deux nœuds :** Soit N_1 et N_2 deux nœuds (deux ensembles de valeurs) à fusionner tels que $\mid N_1 \mid < \mid N_2 \mid$ (i.e. N_1 contient moins d'éléments que N_2). On dit que ces nœuds sont fusionnables si $\forall \alpha \in N_1$, alors $\exists \beta \in N_2$ tq $\alpha \sqsubseteq \beta$ (on note l'unification par $\sqsubseteq$). De plus, une seule des valeurs α ou β peut être projection d'un noyau.
5. **Fusion d'arbres:** Si deux arbres A et B à fusionner contiennent deux nœuds N_1 et N_2 fusionnables, alors le résultat de la fusion de A et B est un arbre ayant pour racine la fusion de N_1 et N_2, et pour descendants les sous-arbres de A et B dominés par N_1 et N_2.
   Remarque : si N_1 et N_2 ne sont pas les racines de A et B, alors soit D_1 et D_2 l'ensemble de nœuds dominant N_1 et N_2, on doit avoir $D_1 \subset D_2$. Dans ce cas, la D_2 dominera le nœud fusionné dans l'arbre résultat.
6. **Propagation de la fusion :** Lorque deux arbres A et B sont fusionnés, N étant leur nœud de fusion, on propage la fusion aux nœuds inférieurs selon les mêmes critères : si les deux nœuds dominés par N sont racines de sous-arbres juxtaposés, connectés par un lien de dépendance et qu'ils sont unifiables, alors les sous-arbres sont fusionnés.

Cette opération de fusion prend donc en entrée une liste d'arbres (la liste des QAU au premier passage) et retourne une nouvelle liste dans laquelle certains arbres ont été fusionnés. La fusion est appliquée à cette nouvelle liste et le processus recommence jusqu'à ce qu'aucun nouvel arbre ne puisse être crée. A partir du second passage, un niveau supplémentaire est systématiquement projeté avant toute nouvelle fusion.

2. On retrouve ici le principe de projectivité des gramaires de dépendance, cf. (Mel'čuk88).

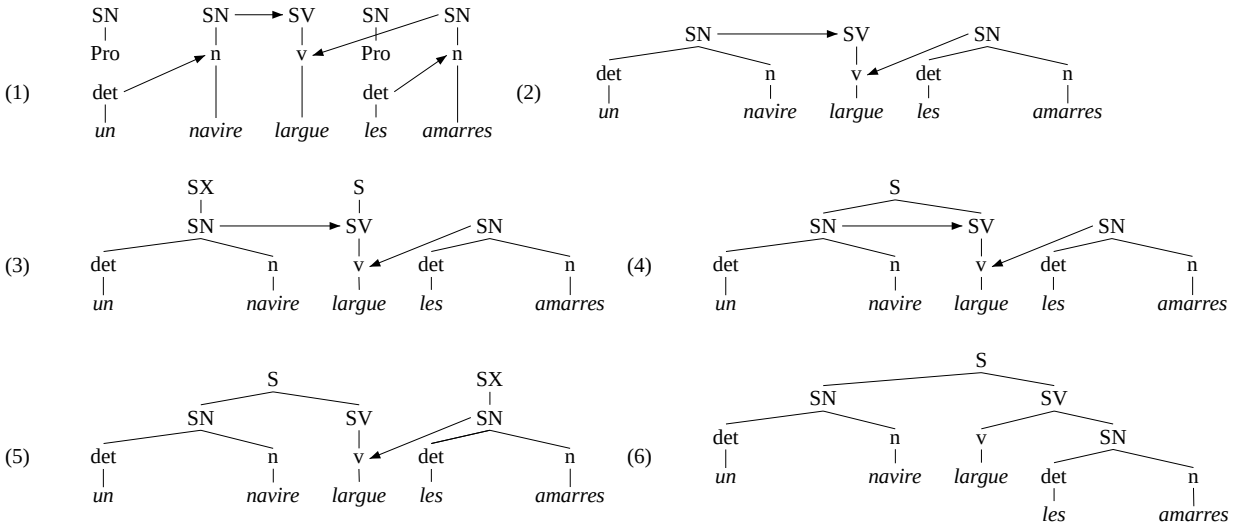


FIG. 1 – *Exemple de fusion*

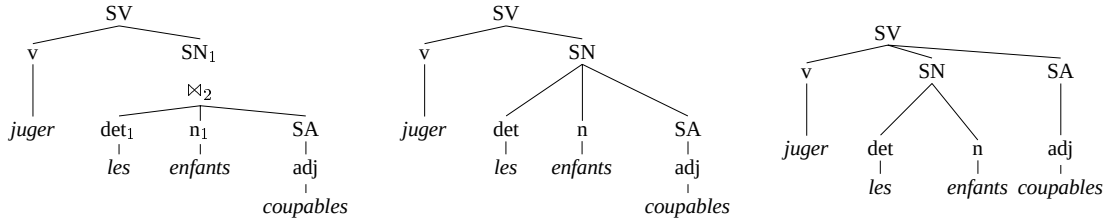


FIG. 2 – *Exemple d'arbre partagé*

La figure (1) décrit les différentes étapes d'un exemple simple (non ambigu) de fusion. Dans cet exemple, les relations de dépendance sont indiquées par des flèches. Pour des raisons de lisibilité, seules les relations définitives ont été indiquées. On remarquera de plus l'introduction d'un symbole *SX* indiquant une valeur variable.

2.4. Traitement de l'ambiguïté : les arbres partagés

Le processus de fusion repose sur la capacité de représenter toutes les possibilités de dominance et de dépendance. Cela ne pose pas de difficultés pour le réseau de dépendance : plusieurs arcs peuvent relier deux nœuds, représentant ainsi l'ambiguïté. En revanche, nous devons proposer une solution pour représenter dans une structure unique les ambiguïtés de dominance. Les solutions classiques proposées reposent soit sur la superposition de structures (les *forêts partagées* proposées dans (Lang91)) soit sur la sous-spécification (les *arbres à rattachements multiples* de (Chen97)). Ces solutions ne sont cependant pas satisfaisantes : les forêts partagées ne forment pas une structure unique et les arbres à rattachements multiples ne sont en fait pas des arbres et n'encodent pas suffisamment d'information.

Nous proposons une solution consistant à introduire un *élément neutre* (noté $\bowtie$) pour la relation de dominance. Celui-ci permettra d'encoder au sein d'une structure arborescente unique des arbres de profondeurs différentes. Nous appelons *arbres partagés* ces arbres dont les nœuds sont des disjonctions de valeur (covariant avec des valeurs d'autres nœuds) et pouvant contenir un élément neutre. L'exemple de la figure (2) décrit le pouvoir expressif de tels arbres. Les

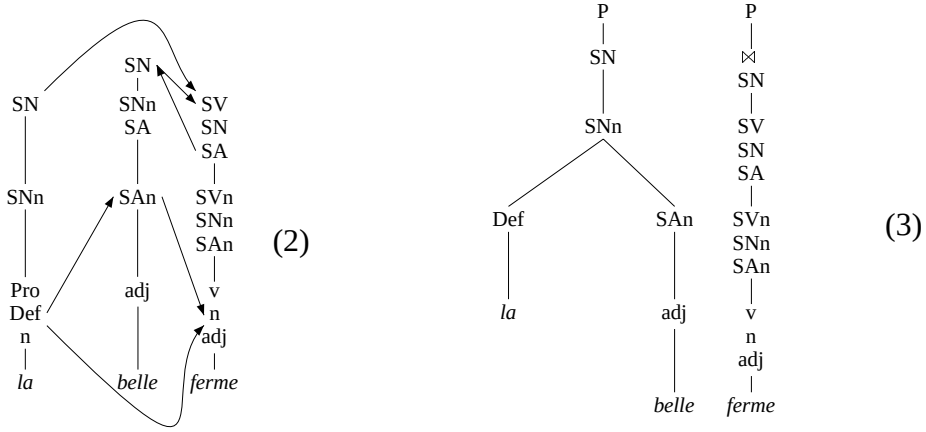
Modèle	*Alphabet*	*Linéarité*	*Fléchage*
SAn	{Adj, SAdvn}		
SA	{SAn, Comp}		
SNn	{N, SAn, pro, dem, def, ind}	def $\prec$ {N, SAn} SAn $\prec$ n	SAn $\rightsquigarrow$ n def $\rightsquigarrow$ n def $\rightsquigarrow$ SAn
SN	{SNn, SA, Comp, -ci}	SNn $\prec$ SA	SA $\rightsquigarrow$ SNn
SVn	{V, aux, être, clit }		
SV	{SVn, SN, SAdvn		
P	{SV, SN }	SN $\prec$ SV	SN $\rightsquigarrow$ SV

FIG. 3 – *Exemples de Propriétés*

contraintes de covariation sont représentées par les indices : ainsi, les valeurs det_1 et n_1 ne peuvent que covarier avec la valeur SN_1. En revanche, la valeur *SA* n'ayant pas de contrainte, elle peut covarier soit avec SN_1 soit $\bowtie_2$. Dans le cas d'une covariation avec $\bowtie_2$, la propriété de l'élément neutre indique que la relation de dominance arrivant sur lui sera en quelque sorte "détournée" vers la valeur du nœud supérieur avec lequel $\bowtie$ covarie (ici *SV*). L'arbre partagé de gauche représente donc les deux arbres simples à droite dans la figure (2).

3. Un exemple de l'utilisation de Propriétés pour F&F

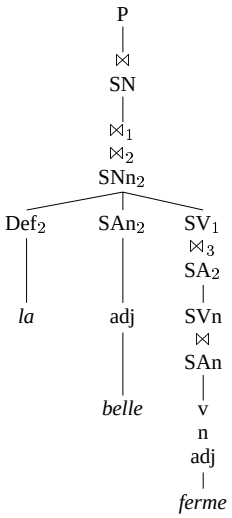
Nous décrivons dans cette section la construction de l'arbre partagé associé à l'énoncé ambigu "*la belle ferme*". Le tableau (3) contient les Propriétés utiles au besoin de l'exempe décrit plus loin. Parmi les Propriétés d'existence, nous ne retiendrons ici que les alphabets. On distingue de plus dans ces Propriétés les syntagmes (e.g. *SN*) des syntagmes noyaux (e.g. *SNn*).



La première étape consiste à construire la liste des QAU associés aux mots de l'énoncé grâce aux Propriétés définissant les alphabets des modèles. Le second traitement établit toutes le relations de dépendance en fonction des Propriétés de fléchage de ces modèles. Dans la suite de QAUs (2), certaines valeurs sont connectées par des liens de dépendance.

Le processus de fusion lui-même consiste à analyser les deux premiers arbres de cette liste. Ceux-ci sont fusionnables par leur racine : deux de leurs valeurs (*Def* et *SAn*) sont en effet connectées par un lien de dépendance. L'arbre obtenu est représenté dans la suite d'arbre partagés (3) dans laquelle un niveau supplémentaire a été projeté pour préparer la fusion de l'étape suivante : *SN* et *SV* se projettent en *P* tandis que *SA* se projette en *SN*. Un élément neutre a été introduit dans le dernier arbre. Il s'agit là du premier cas d'introduction de $\bowtie$ qui est possible lorqu'une des valeurs d'un nœud est une projection de l'autre (ici, *P* est projection de *SN*). Dans ce cas, on remplace dans ce nœud la projection par $\bowtie$ et on crée une nouvelle racine composée

de cette projection.



Les deux arbres de la figure ci-contre sont à nouveau fusionnables : ils sont juxtaposés, connectés par une relation de dépendance et leurs racines sont unifiables. Il est possible de propager la fusion à des nœuds inférieurs (*SN* et *SNn*) selon les valeurs instanciées dans l'arbre de *ferme* : si *ferme* est catégorisé en *v*, on fusionne seulement la racine *P*, s'il est catégorisé en *adj*, on fusionne (en partant de la racine) *P* et *SN*, s'il est catégorisé en *n*, on fusionne *P*, *SN* et *SNn*.
La représentation de ce phénomène dans un arbre partagé se fait en introduisant à nouveau des éléments neutres. Chaque fois qu'une possibilité de propagation de fusion est décelée, on crée un *tronc commun* composé de la propagation maximale de la fusion, chaque nœud de ce tronc étant formé par la valeur fusionnable et un élément neutre ⋈ . Les branches partant de ce tronc sont les deux sous-arbres dominés par les racines des arbres initiaux à fusionner.

4. Conclusion

Nous avons présenté dans cet article les éléments fondamentaux de l'approche *5P* et montré en quoi les critères qui la motivent convergent avec ceux d'une approche générale et réutilisable d'analyse automatique. Cette convergence repose sur une description linguistique à l'aide de Propriétés qui sont autant de contraintes pour l'analyse. La technique de *filtrage et fusion* tire parti de cette représentation en proposant la construction d'une représentation syntaxique distinguant les informations de dépendance et de dominance. Cette structure, en fonction du nombre et du niveau des Propriétés utilisées, pourra être plus ou moins détaillée. Nous sommes donc en mesure d'envisager une technique unique (et un même système) pour des applications visant une analyse superficielle ou détaillée. Nous avons de plus proposé un traitement original de l'ambiguïté totalement intégré à *F&F*.

Références

Anne Abeillé, Danièle Godard & Ivan Sag (en préparation) "The Major Syntactic Structures of French".

Gabriel Bès (1999) "La phrase verbale noyau en français", in *Recherches sur le français parlé* 15, Université de Provence.

Gabriel Bès, Philippe Blache & Caroline Hagège (1999) "*The 5P Paradigm*", rapport de recherche, GRIL/LPL.

Philippe Blache (1998) "Une stratégie de contrôle pour l'analyse syntaxique", in actes de *TALN'98*.

Claire Blanche-Benveniste & al. (1991) *Le français parlé*, Editions du CNRS.

John Chen & K. Vijay-Shanker (1997) "Towards a Reduced Commitment, D-Theory Style TAG Parser", in proceedings of the *International Workshop on Parsing Technologies*.

Caroline Hagège & Gabriel Bès (1998) "Da observaçao da propriedades linguisticas à formalizo numa gramàtica do processamento da lingua", in proceedings of *Encontro para o processament computacional da lingua portuguesa escrita e falada*.

Bernard Lang (1991) "Towards a Uniform Formal Framework for Parsing", In M. Tomita (ed.), *Current Issues in Parsing Technology*, Kluwer Academic Publishers.

Igor Mel'čuk (1988) "*Dependency Syntax*", SUNY Press.