

# Chapitre 7

## Systeme de traduction d'icônes en langue

### 7.1 Introduction : Architecture du système PVI

Ce chapitre décrit la conception et le fonctionnement du programme de compréhension de séquences d'icônes et de génération de phrases en français mis en œuvre dans le système PVI. Le système complet intègre ce programme et une interface graphique sophistiquée conçue pour lui dans une architecture spécifique que nous décrivons ici brièvement. Le lecteur se reportera, pour une description plus particulière de l'interface de PVI, au rapport de Michaël Checler [1995].

Le système PVI est conçu pour fonctionner sur un ordinateur **Apple™** de type **Macintosh**, sous **Multifinder**, avec un système d'exploitation **AppleOS** version 7, muni des extensions **AppleEvents** (en standard dans les versions 7.5 et ultérieures d'AppleOS) et QuickTime. Il requiert une configuration de 8 Mo de mémoire vive (il en consomme 7 au total : 4 Mo pour l'interface et 3 Mo pour le programme décrit ici).

L'ordinateur doit être également muni d'un système de synthèse vocale, qui peut être matériel (carte de synthèse) ou logiciel (programme de synthèse compatible avec l'architecture **Apple™ Speech Manager**

Le principe de fonctionnement en est la désignation de symboles iconiques disposés sur l'écran, puis leur conversion en phrases françaises, et l'énonciation vocale de celles-ci par un programme de synthèse vocale artificielle.

Le système PVI comprend le programme de conversion icônes-phrases et le programme d'interface graphique. Il ne comprend pas le système de synthèse vocale.

#### 7.1.1 Éléments fonctionnels du système PVI

- I. En fonctionnement, PVI met en jeu deux processus parallèles :
  - le programme de conversion de séquences d'icônes en phrases françaises (dorénavant « *module linguistique* »), programme PROLOG développé sur

- la version 3.1 du langage **Prolog II+<sup>TM</sup>** de PROLOGIA, et ;
- un programme d'interfaçage graphique modulaire et configurable (dénommé « *IHM* »), programme exécutable développé et compilé dans l'environnement de développement **Think C**.

Le module linguistique nécessite un environnement d'exécution **Prolog II+** pour fonctionner, tandis que l'IHM est un programme exécutable autonome.

- II. Les deux processus communiquent par l'intermédiaire d'une *ressource de code*, c'est-à-dire d'une bibliothèque de fonctions externes permettant au processus IHM de faire appel à des fonctions du module linguistique. Appelée **CIA** (*Communication Inter-Applications*), cette ressource de code permet un fonctionnement de type client/serveur, l'IHM s'adressant comme client au serveur linguistique (pour une description plus complète, cf. [Checler, 1995]).

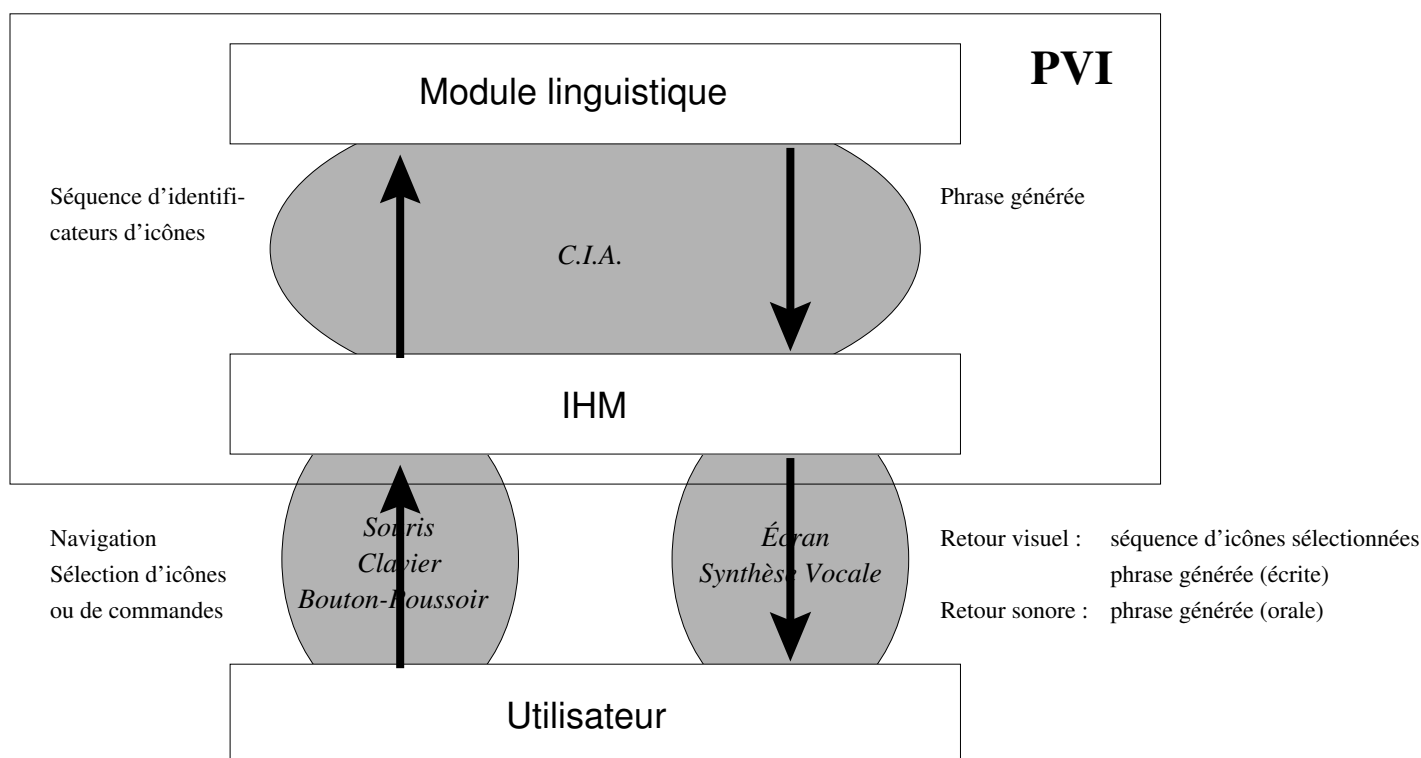


FIG. 7.1: Les modules du système PVI

La figure 7.1 présente une vision générale des interfaces du système PVI.

Des détails sur le fonctionnement pratique du système peuvent être trouvés dans la notice d'utilisation [Vaillant, 1995].

### 7.1.2 Architecture du module linguistique

Le module linguistique, auquel est consacré cette description, est lui-même conçu en trois modules qui correspondent aux grandes étapes de traitement du message construit un message en langue naturelle à partir du réseau sémantique linguistique (fig.7.2).

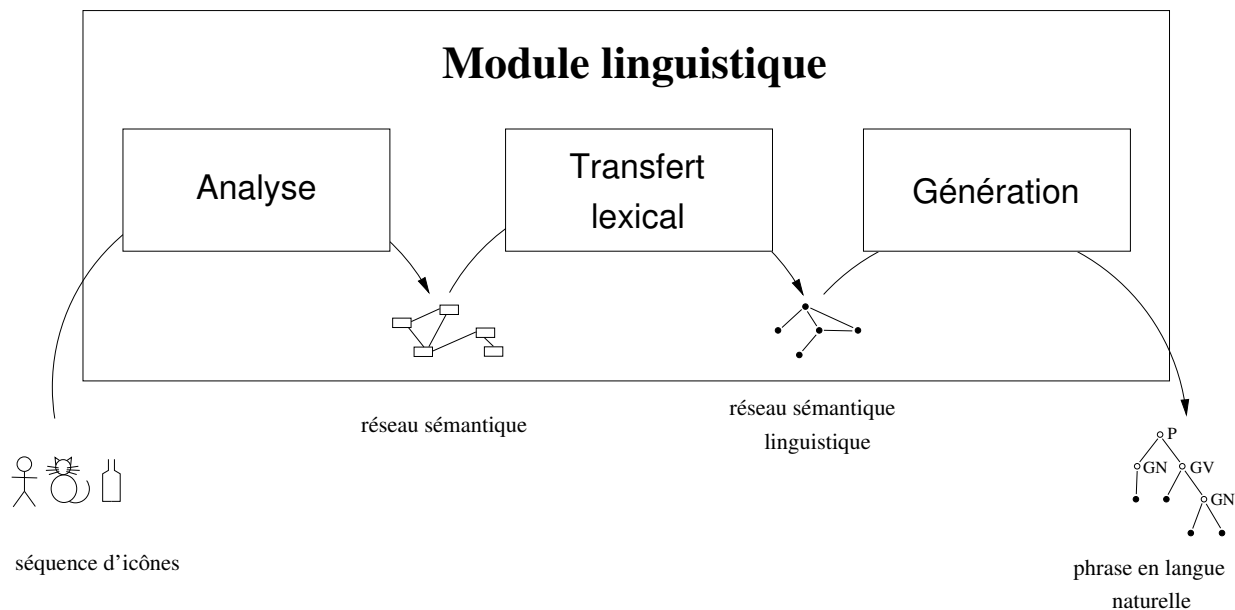


FIG. 7.2: Les sous-modules du module linguistique

→ Dans la suite, les identificateurs de fichier du type `Fichier.p2` désigneront des fichiers sources PROLOG, et les identificateurs de règles du type `regle/3` désigneront des règles PROLOG et leur arité.

## 7.2 Le module d'analyse sémantique

Le module d'analyse prend en entrée une séquence d'identificateurs d'icônes, et la convertit, sans utiliser d'information grammaticale, en une représentation du sens.

Le principe de base de l'analyse est, ici comme partout, de *reconstituer les relations sémantiques* entre prédicats et attributs pour aboutir à une représentation du sens où chaque actant se voit attribuer son rôle casuel. Ainsi dans HOMME/EAU/BOIRE, on doit retrouver l'information [BOIRE : *agent*=HOMME, *objet*=EAU]. Toutefois, ce système postule qu'il ne peut compter ni sur l'ordre des mots (fig.7.3.a), ni sur une information morphologique (fig.7.3.b), pour attribuer « à coup sûr » les rôles casuels.

C'est pourquoi l'analyse doit se fonder ici sur des informations de nature sémantique : une sorte de sens commun modélisé informatiquement qui permette de reconnaître en « homme » l'agent, et en « eau » l'objet, de l'action de boire.

### 7.2.1 Le dictionnaire d'icônes

L'ensemble de ces informations sont regroupées dans un lexique sémantique des icônes reconnues par l'application. La modélisation du contenu s'inspire, dans sa forme et dans sa terminologie, des travaux de Rastier [1987, 1994] sur la langue : nous en avons transposé une partie sur le langage d'icônes.

Afin d'éviter les dérives traditionnellement liées à l'organisation du lexique au-

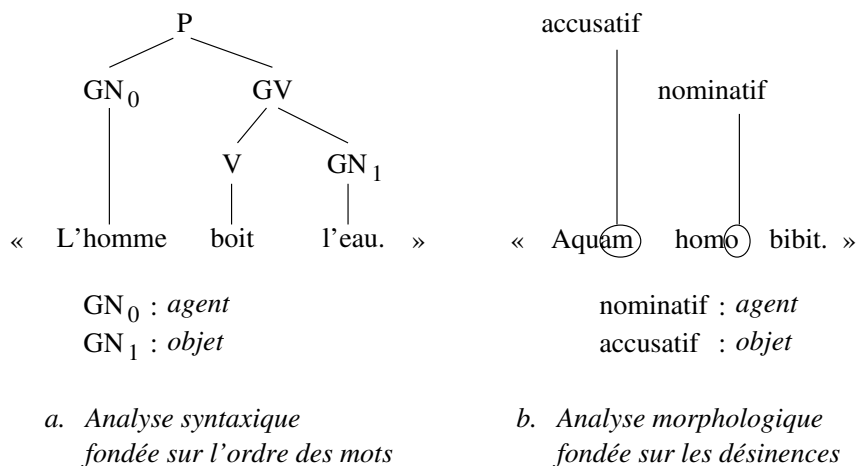


FIG. 7.3: Analyse grammaticale des langues naturelles (a. syntaxique, b. morphologique)

tour d'une ontologie *a priori* — à savoir, l'inadaptation de la connaissance modélisée (insuffisante ou redondante) avec les besoins de l'application —, le contenu sémantique pertinent, considéré comme un objet purement linguistique, a été extrait d'un corpus recueilli auprès d'enfants handicapés du Centre de Kerpape.

La méthodologie théorique permettant de décrire la sémantique des symboles de l'application en se fondant uniquement sur le corpus, et non sur une connaissance du monde définie « de l'extérieur », est celle décrite par Hjelmslev [1968]. Une décomposition systématique d'échantillons suffisants d'un système de signes devrait ainsi permettre d'en décrire formellement la structure. Dans la pratique, nous avons fait intervenir notre connaissance « naturelle » lors de l'analyse du corpus pour constituer les catégories ; cette intervention est légitime dans le cas d'un langage sans syntaxe. Le fait de se fonder sur le corpus préserve de toute façon les exigences d'exhaustivité et de description pertinente des traits spécifiques.

### 7.2.1.1 Structuration en niveaux

Le dictionnaire ainsi conçu comprend trois niveaux (fig.7.4) :

- Le *domaine*, niveau de caractérisation de grands champs sémantiques (ex. // *alimentation* //, // *distraktion* // ...). Il n'a pas de contenu propre, mais factorise des isotopies macrogénériques (et peut ainsi éviter des erreurs d'interprétation de base) ;
- Le *taxème*, niveau de la catégorie : il regroupe les icônes pouvant avoir le même rôle dans un contexte de base (lié bien sûr au domaine auquel elles sont rattachées). À ce niveau, on peut avoir un contenu sémantique : il s'agit du contenu commun à tous les éléments de la catégorie. Les icônes elles-mêmes hériteront automatiquement de ce contenu.
- Enfin, l'*icône* regroupe à son niveau le contenu *spécifique*, qui permet de la différencier des autres icônes du même taxème.

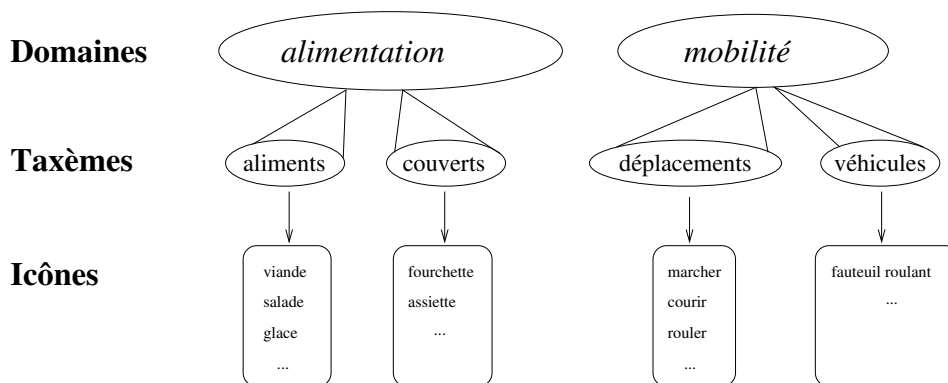


FIG. 7.4: La structuration en niveaux du dictionnaire d'icônes

Le contenu est lui-même modélisé sous la forme de *sèmes*, ou *traits sémantiques*, éléments de sens atomiques représentant des différences minimales pertinentes. Une icône possède ainsi trois niveaux de sèmes : au niveau macrogénérique tout d'abord, les sèmes d'« étiquetage » du domaine et du taxème (ils sont au nombre de deux) ; au niveau microgénérique, les sèmes qui représentent le contenu commun à toutes les icônes du même taxème ; enfin au niveau spécifique, les sèmes qui représentent le contenu propre de l'icône, par différences avec celui des icônes du même taxème.

### 7.2.1.2 Sèmes intrinsèques et sèmes extrinsèques

Il existe des sèmes de deux sortes : les *sèmes intrinsèques*, qui représentent le contenu « propre » de l'icône, sans la considérer en relation avec aucune autre, et les *sèmes extrinsèques*, ou *traits de sélection*, qui représentent les conditions que les icônes de nature prédicative imposent à leurs éventuels actants. Les premiers sont des attributs simples, généralement bivalués (+1 ou -1 selon qu'ils sont présents ou absents du noyau de sens) ; les seconds sont des attributs attachés à un rôle casuel particulier. Ainsi <canin,+1> est-il un sème *intrinsèque* pour la définition de “chien”, alors que <agent,<canin,+1>> est un sème *extrinsèque* pour la définition d’“aboyer”.

Les sèmes extrinsèques font partie de la définition du sens d'un concept aussi bien que les sèmes intrinsèques<sup>1</sup>. Ils sont simplement rattachés à un rôle casuel, donc conditionnent les *relations sémantiques* qui peuvent surgir entre ce concept et son contexte — alors que les sèmes intrinsèques concernent le cœur du concept.

Selon l'icône considérée, on pourra donc soit n'avoir que des sèmes intrinsèques, soit n'avoir que des sèmes extrinsèques, soit avoir les deux. Un exemple représentatif, correspondant à la structure du contenu de l'icône “courir”, est donné dans le tableau 7.1.

Cette représentation, distribuée en sèmes génériques et sèmes spécifiques, est équivalente à la représentation « compilée » exposée fig.7.5.

Tous ces éléments sont convoqués lors du calcul d'« affinité sémantique » en quoi

<sup>1</sup>Ainsi, en langue, on les trouve même souvent mentionnés dans les dictionnaires (par exemple « ABOYER : donner de la voix, en parlant d'un chien. », (Petit Robert).

traits sémantiques	intrinsèques		extrinsèques
domaine	//mobilité//		
taxème	'déplacement'	<mobilité,+1> <concret,+1> <animé,-1>	(étiquetage du domaine) <agent,<animé,+1>>
icône	'courir'	<déplacement,+1> <rapide,+1>	(étiquetage du taxème) <agent,<mobile,+1>> <destination,<lieu,+1>> <instrument,<véhicule,-1>>

TAB. 7.1: Le contenu d'une icône décomposé en sèmes

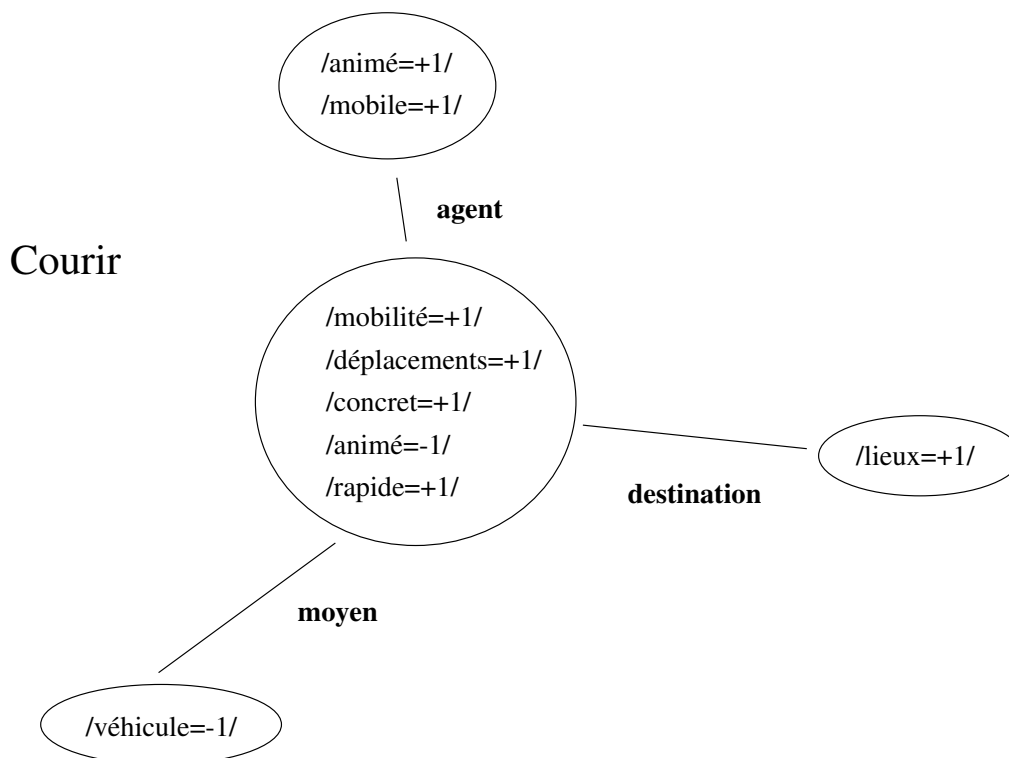


FIG. 7.5: Vue synthétique du contenu d'une icône

consiste l'analyse (§ 7.2.2).

### 7.2.1.3 Héritage de traits

Une petite *doxa* des subdivisions fondamentales en genres et en espèces doit être représentée dans le dictionnaire pour prendre en compte les phénomènes de compréhension, ou implications naturelles entre traits sémantiques fondamentaux, sans allonger exagérément les listes de sèmes (nous ne faisons pas ici de suppositions ontologiques fortes, mais nous mettons en facteur, pour des raisons d'économie, des afférences généralement admises dans les corpus considérés).

Le mécanisme d'analyse est en effet conçu de telle sorte qu'il essaye de trouver les actants « les mieux adaptés » pour chaque rôle casuel possible. On cherche donc à disposer dans le lexique d'informations précises sur les actants idéaux attendus

par un prédicat, de sorte que cette recherche du mieux adapté soit effectivement discriminante. Il est néanmoins nécessaire que des candidats partageant un assez grand nombre de traits génériques en commun avec le candidat idéal soient considérés comme mieux adaptés que ceux n'en partageant pas ; ainsi un animal peut-il être considéré comme un « remplaçant » acceptable pour un humain comme agent d'un prédicat, alors qu'une notion abstraite, par exemple, ne l'est pas du tout et doit être rejetée lors de l'analyse.

Or nous avons choisi, pour ménager une grande souplesse au modèle du lexique, et pour assurer sa pertinence dans un corpus particulier, de ne pas organiser celui-ci selon une taxonomie rigide (§ 7.2.1.1). Nous n'avons donc pas d'« arbre d'héritage » universel qui permette, par simple examen de la filiation d'un concept en genres d'extension de plus en plus grande, de retrouver ce type d'information ; de savoir par exemple qu'/humain/ comprend /animé/.

Il n'est en outre pas judicieux, dans le modèle de lexique utilisé ici, de stocker au niveau de chaque rôle casuel l'ensemble des traits sémantiques généraux qui pourraient éventuellement s'y appliquer (en notant par exemple, lorsqu'un prédicat a un sème extrinsèque <agent,<humain,+1>>, qu'il a aussi les sèmes extrinsèques <agent,<animé,+1>>, <agent,<vivant,+1>>, <agent,<concret,+1>> ...) : ces sèmes ne seraient pas pertinents, mais seulement présents pour l'exhaustivité de la représentation.

Nous avons donc défini un arbre d'héritage (fig.7.6), sur des sèmes et non sur des concepts. Cet arbre décrit une « ontologie » primitive : il représente les liens de compréhension ou d'exclusion entre quelques sèmes très courants.

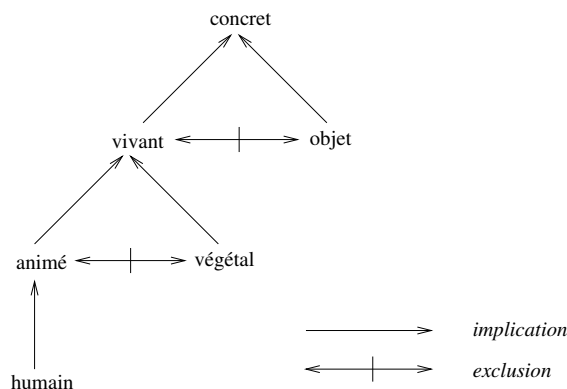


FIG. 7.6: Arbre d'héritage de sèmes

Lors de l'analyse, les sèmes « ascendants » d'un sème extrinsèque attaché à une icône prédicative donnée, même s'ils ne sont pas explicitement spécifiés, sont ainsi convoqués eux aussi lors du calcul. Pour que leur importance globale n'excède pas, à cause de leur nombre, celle des sèmes explicitement présents dans la représentation lexicale, un coefficient constant appelé `droits_de_succession` leur est affecté.

Cette forme de représentation, qui consiste à considérer l'héritage comme concernant les composants sémantiques primitifs et non les concepts, permet d'intégrer une ontologie au lexique sans pour autant considérer le lexique lui-même comme une ontologie : le lexique n'est pas un arbre de Porphyre, et un même sème peut

être générique dans un taxème donné, spécifique dans un autre, tout en gardant sa propre définition.

#### 7.2.1.4 Cadres sémantiques

Nous avons exposé (§ 7.2.1.2) comment une certaine doxa, ou « connaissance naturelle » reçue du monde, modélisée dans le lexique, fournit les données de l'analyse sémantique. Cette connaissance porte essentiellement sur les relations qu'une icône peut lier avec une autre : il s'agit en effet des propriétés qu'une icône prédicative donnée  $i$  peut imposer à ses actants  $c(i)$  en les propageant le long de ses relations casuelles :

$$i \rightarrow c(i)$$

Il existe un second niveau de connaissance naturelle, celui des relations particulières que *deux* icônes peuvent lier par l'intermédiaire d'une troisième : il s'agit des propriétés qu'une icône prédicative  $i$  impose à l'un de ses actants  $c_2(i)$  lorsqu'un autre de ses actants,  $c_1(i)$ , possède lui-même des propriétés particulières :

$$c_1(i) \rightarrow c_2(i)$$

Cette implication *conditionnelle*, où le prédicat joue un rôle de pivot entre deux types d'actants particuliers pouvant entrer conjointement en relation avec lui, doit également être modélisée dans le lexique pour prendre en compte certains phénomènes. Ainsi doit-on pouvoir prédire qu'un agent */carnivore=+1/* pour le verbe “*manger*” renforce la plausibilité d'un objet */animaux=+1/* (autrement improbable dans ce rôle).

Cette sorte de connaissance est représentée par des « cadres », de la forme représentée fig.7.7.

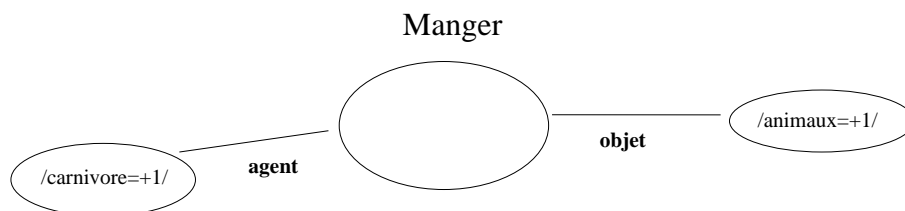


FIG. 7.7: Un cadre sémantique pour l'icône ‘*manger*’

Ces cadres se superposent le cas échéant, lors de l'analyse, à la structure prédicative de base de l'icône (fig.7.5), pour former une structure du même type, mais enrichie.

#### 7.2.1.5 Emplois multiples

Dans certains cas, il est nécessaire de pouvoir gérer l'utilisation de la même icône pour des emplois différents. Il ne s'agit pas véritablement d'homonymie, puisque les icônes de l'application constituent un système de signes artificiel où celle-ci est *a priori* bannie, mais d'emploi du même signifiant dans des taxèmes distincts. Ainsi



l'icône représentant un téléphone peut-elle être utilisée pour représenter l'objet “*téléphone*” (au sens de « poste téléphonique »), dans le taxème des appareils domestiques, autant que pour représenter l'action “*téléphoner*”, dans le taxème des communications.

Ces doubles emplois apparaissent fréquemment lors des premiers essais du programme, alors qu'ils n'avaient pas été prévus au départ : on s'aperçoit par exemple que l'utilisateur emploie spontanément l'icône “*téléphone*” au sens du verbe téléphoner, sans avoir subi d'apprentissage. On est donc amené à prévoir cet emploi tant il semble surgir naturellement.

→ *Ce type de dérivation n'a rien de surprenant comme on peut le voir à l'œuvre de façon systématique dans la langue. Dans l'exemple donné, il s'agit de dérivation métonymique. Dans d'autres cas, il peut s'agir de variation d'aspect ou de structure casuelle d'un même concept, ou même simplement de changement de cas sémantique (“table” dans le taxème des meubles vs. “table” dans le taxème des lieux — correspondant l'un à l'accusatif, l'autre au locatif).*

*Les langues, nous le savons, possèdent des mécanismes morphologiques pour marquer ces dérivations (suffixation, préfixation). Le langage BLISS [Hehner, 1980], langage d'idéogrammes artificiel auquel on a songé à adjoindre un système morphologique de base (cf. chap. 5, §5.4), en possède aussi ; on pourrait donc imaginer une configuration BLISS de PVI, dans laquelle ces différents emplois seraient marqués au niveau du signifiant — encore que l'utilité en soit contestable, car les utilisateurs handicapés ont l'habitude d'utiliser BLISS comme catalogue d'idéogrammes, sans se soucier souvent de son système de dérivation par signes diacritiques. Dans la configuration implémentée en tout cas, ce n'est pas le cas.*

Nous avons donc tout simplement modélisé ces différents emplois en faisant correspondre à un seul signifiant iconique plusieurs entrées du dictionnaire figurant dans des taxèmes différents, par exemple “*téléphone1*” (poste téléphonique), “*téléphone2*” (téléphoner) :

`emplois(telephone,telephone2.telephone1.nil) ->;`

Lors de l'analyse d'une séquence d'icônes, chaque icône est donc examinée pour voir si elle ne correspond pas à plusieurs emplois. Si c'est le cas, la séquence de signifiants est convertie en autant de séquences de signifiés. C'est au bout du compte le plus haut score d'analyse qui détermine quel emploi est pris en compte.

## 7.2.2 Algorithme d'analyse

Le principe de base de l'analyse dans PVI est de reconstituer la structure sémantique du message en affectant de façon adéquate les actants des prédicats. Dans une séquence d'icônes d'entrée, l'analyse commence donc par repérer les icônes prédictives, c'est-à-dire celles qui ont une structure casuelle implicite, et, pour chacune d'elles, recherche dans son contexte quelles sont les meilleures icônes candidates pour « remplir » les rôles casuels.

La question se ramène donc à celle de la détermination de ces « meilleures »

candidates; en d'autres mots, il est nécessaire de définir une grandeur, un « score » de compatibilité sémantique, à l'aune duquel on détermine la valeur des affectations d'actants.

Le processus fonctionne ensuite en calculant la meilleure affectation globale possible d'icônes à cas sur l'ensemble de la séquence.

Dans la suite, nous noterons  $s_1, s_2, \dots, s_n$  la séquence de symboles iconiques en entrée de l'analyse. Chacune de ces icônes a un ensemble de sèmes intrinsèques :

$$\mathcal{SI}(s_i) = e_i$$

(où  $e_i$  est un ensemble de traits sémantiques simples, du type couple attribut-valeur),

et certaines d'entre elles ont des sèmes extrinsèques qui constituent une structure casuelle ( $\mathcal{SC}$ ) lorsqu'on les factorise par cas :

$$\mathcal{SC}(s_i) = \{\langle c_1, e_{i1} \rangle, \langle c_2, e_{i2} \rangle, \dots, \langle c_N, e_{iN} \rangle\}$$

(où chacun des  $N$   $c_j$  est un type casuel comme *agent*, *objet*, *instrument* ..., et chaque  $e_{ij}$  un ensemble de traits sémantiques simples de type couple attribut-valeur).

On pourra noter plus précisément l'ensemble des traits sémantiques simples (attribut-valeur) rattachés à  $s_i$  en tant que traits extrinsèques pour un cas  $c_j$  :

$$\mathcal{SE}(s_i, c_j) = e_{ij}$$

ce qui est équivalent à :

$$\langle c_j, e_{ij} \rangle \in \mathcal{SC}(s_i)$$

### 7.2.2.1 Compatibilité sémantique

La *compatibilité sémantique* est la valeur que l'on cherche à maximiser pour déterminer les meilleures affectations. C'est une relation binaire asymétrique : elle mesure le degré de compatibilité d'une icône à une place d'actant d'une autre icône.

**Au niveau du sème :** la compatibilité atomique se calcule au niveau du sème, entre un sème intrinsèque de l'icône « candidat » et un sème extrinsèque de l'icône « prédicat ». La compatibilité est nulle si les sèmes sont orthogonaux (attribut différent). Si l'attribut est le même, elle est égale au produit des valeurs attachées à cet attribut dans l'un et l'autre sème. Par exemple la compatibilité sémantique est nulle entre */animé= +/* et */petit= +/*, positive entre */animé= +/* et */animé= +/*, négative entre */animé= +/* et */animé= -/*.

$$\begin{aligned} \mathcal{C}(\langle a_1, v_1 \rangle, \langle a_2, v_2 \rangle) &= 0 && \text{si } a_1 \neq a_2 \\ \mathcal{C}(\langle a, v_1 \rangle, \langle a, v_2 \rangle) &= +1 && \text{si } v_1 \text{ et } v_2 \text{ sont entiers et égaux} \\ &= -1 && \text{si } v_1 \text{ et } v_2 \text{ sont entiers et distincts} \\ &= v_1 \cdot v_2 && \text{si l'une des deux valeurs est réelle} \end{aligned} \tag{7.1}$$

Les valeurs des sèmes extrinsèques sont des coefficients réels. Ce choix de modélisation permet de régler finement le caractère plus ou moins contraignant (« sélectif ») d'un trait extrinsèque. Les valeurs des sèmes intrinsèques, elles, sont entières : généralement  $\{+1, -1\}$ , ou, dans le cas du sème spatial /dimension/,  $\{1, 2, 3\}$ .

**Au niveau des structures de traits :** les signifiés des icônes sont représentés par des ensembles de traits. Le calcul de la compatibilité sémantique entre deux ensembles de traits se fait entre deux ensembles homogènes de couples attribut-valeur. Ce sont d'une part les sèmes extrinsèques attachés à un actant donné de l'icône prédicat — dépouillés donc du type de relation casuelle, avec seulement le couple attribut-valeur —, et d'autre part les sèmes intrinsèques de l'icône candidat.

La compatibilité du deuxième ensemble *avec* le premier est définie comme la somme des compatibilités sémantiques des sèmes communs aux deux ensembles, rapportée au nombre total de sèmes dans le premier ensemble :

$$\mathcal{C}(e_1, e_2) = \frac{\sum_{\langle a, v_1 \rangle \in e_1, \langle a, v_2 \rangle \in e_2} \mathcal{C}(\langle a, v_1 \rangle, \langle a, v_2 \rangle)}{\text{cardinal de } e_1} \quad (7.2)$$

( $e_1$  est l'ensemble des sèmes filtrants)

**Au niveau d'une relation actancielle :** la compatibilité sémantique est à ce niveau le degré d'affinité d'une icône candidate  $s_j$  à une place actancielle d'une icône prédicative  $s_i$ .

C'est donc la compatibilité sémantique de l'ensemble des sèmes intrinsèques de  $s_j$  avec l'ensemble des sèmes extrinsèques de  $s_i$  pour le cas  $c_k$  :

$$\mathcal{C}(\mathcal{SE}(s_i, c_k), \mathcal{SI}(s_j)), \quad (7.3)$$

Cette grandeur exprime la plus ou moins grande « conformité » du contenu sémantique intrinsèque de  $s_j$  avec les contraintes que  $s_i$  impose à son actant pour le cas  $c_k$ , contraintes modélisées par les sèmes extrinsèques (§ 7.2.1.2). Elle conditionne l'unification de  $s_j$  comme actant de  $s_i$  (fig.7.8).

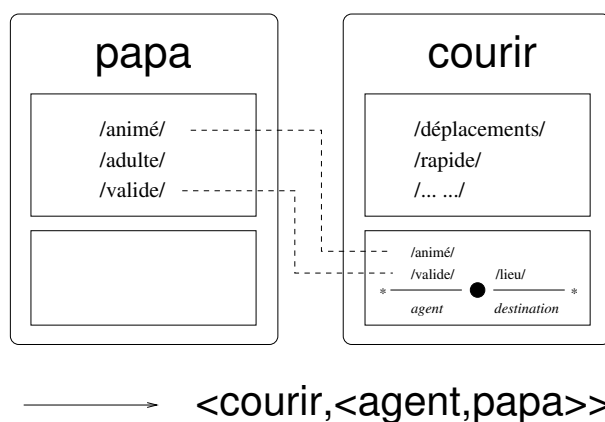


FIG. 7.8: La compatibilité sémantique entre une icône et la place actancielle d'une icône prédicative

### 7.2.2.2 Affectation d'actants

Pour chaque icône prédicative de la séquence d'entrée, il s'agit donc de remplir au mieux tous les rôles casuels : le mécanisme de distribution d'actants à l'icône prédicative fonctionne en essayant de trouver la combinaison qui maximise globalement la valeur des attributions d'actant.

Nous appelons *affectation* une combinaison possible d'attributions d'actants pour l'icône prédicative  $s_i$ . C'est donc une application de l'ensemble des cas de  $s_i$  :

$$\{c_1, c_2, \dots, c_k\},$$

dans un ensemble d'icônes de la séquence d'entrée candidates à en être les actants :

$$\{s_{i1}, s_{i2}, \dots, s_{ij}\}.$$

Soit  $A$  une application de ce type :

$$A = \{\langle c_x, s_{iy} \rangle\}, \text{ où } x \in [1, k] \text{ et } y \in [1, j]. \quad (7.4)$$

Nous cherchons à maximiser une valeur globale pour cette affectation. Cette valeur globale est la somme des valeurs de chacune des attributions individuelles  $\langle c_x, s_{iy} \rangle$  qui constituent l'application  $A$  :

$$\mathcal{V}(A) = \mathcal{V}(s_i, \{\langle c_1, s_{i1} \rangle, \langle c_2, s_{i2} \rangle, \dots, \langle c_k, s_{ik} \rangle\}) = \sum_{j \in [1, k]} \mathcal{V}(s_i, c_j, s_{ij}) \quad (7.5)$$

La valeur de chaque attribution individuelle d'actant,  $\mathcal{V}(s_i, c_j, s_{ij})$ , n'est pas la simple compatibilité sémantique de l'icône  $s_{ij}$  avec le cas  $c_j$  de l'icône  $s_i$  (Éq. 7.3). Il faut en effet tenir compte, pour évaluer la valeur d'une attribution d'actant à un prédicat dans le contexte d'une séquence d'icônes, d'un facteur qui est la *distance* de l'icône candidate au prédicat.

Les séquences d'icônes en entrée de PVI s'inscrivent en effet dans une linéarité syntagmatique, et sont sujettes aux phénomènes les plus universels liés à cette dimension des systèmes de signes, à savoir l'effet de récence et ses corrélats. On considère certes que les messages produits dans le contexte d'utilisation de ce système n'ont pas de *syntaxe*, mais cela n'exclut pas qu'ils aient une *syntagmatique*.

Ainsi lors de l'analyse d'une séquence longue comme PAPA/VOIR/CHAT/GENTIL/MANGER/VIANDE, au moment de décider de l'affectation du sujet du prédicat GENTIL, c'est la proximité de CHAT qui devra décider de son choix de préférence à PAPA — qui autrement, hors de tout contexte, a la même compatibilité sémantique avec cet adjectif.

Nous définissons donc la *valeur* de l'attribution de l'icône  $s_{ij}$  comme actant, pour le cas  $c_j$ , de l'icône prédicat  $s_i$ , comme étant la compatibilité sémantique de  $s_{ij}$  avec le cas  $c_j$  de  $s_i$ , multipliée par un coefficient qui est une fonction  $D$  de la distance entre  $s_i$  et  $s_{ij}$  :

$$\mathcal{V}(s_i, c_j, s_{ij}) = D(s_i, s_{ij}) \cdot \mathcal{C}(\mathcal{SE}(s_i, c_j), \mathcal{SI}(s_{ij})) \quad (7.6)$$

→ Le choix de la fonction  $D$  s'est fait purement par tâtonnements, en essayant diverses fonctions décroissantes et en testant leur adéquation avec les résultats attendus sur un corpus donné. Nous avons finalement arrêté le choix sur la fonction  $e^{n^2 \cdot \ln(L)}$ , où  $n$  est la distance entre les deux icônes, et  $L$  une constante comprise entre 0 et 1, appelée *localite*, et qui détermine la vitesse de décroissance en fonction de la distance (une constante de localité égale à 1 signifie aucune décroissance quelle que soit la distance, une constante égale à 0 signifie que le coefficient tombe à 0 dès la première icône après le prédicat). Cette fonction ne prétend à aucune validité de modélisation cognitive ou autre.

Une affectation n'intègre pas systématiquement une attribution d'icône à chacun des actants possibles : les attributions individuelles ayant une valeur inférieure à un certain seuil (constante *seuil*) sont rejetées, et la place actancielle reste vacante.

Chaque affectation  $A$  calculée par le moteur PROLOG à ce niveau, pour chaque icône prédicative  $s_i$ , est donc un ensemble de couples  $\langle c_j, s_{ij} \rangle$  tel que chacun de ces couples vérifie :

$$\mathcal{V}(s_i, c_j, s_{ij}) = D(s_i, s_{ij}) \cdot \mathcal{C}(\mathcal{SE}(s_i, c_j), \mathcal{SI}(s_{ij})) > \text{seuil}$$

et dont la valeur totale est

$$\mathcal{V}(A) = \sum_{\langle c_j, s_{ij} \rangle \in A} \mathcal{V}(s_i, c_j, s_{ij}) \quad (7.7)$$

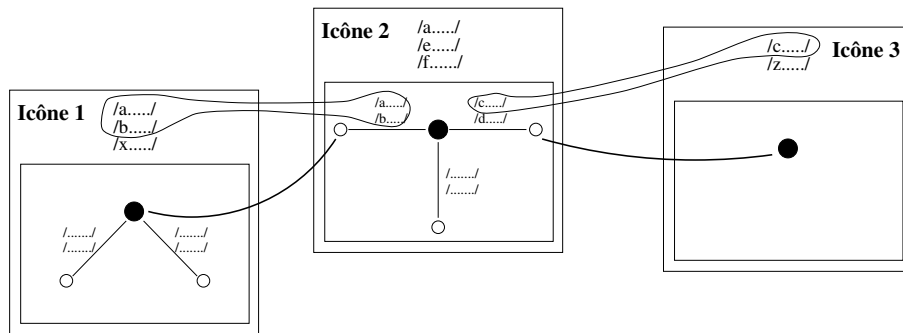


FIG. 7.9: Une affectation d'actants fondée sur la comatibilité sémantique

### 7.2.2.3 Interprétation de la séquence

Une *interprétation* de la séquence d'icônes d'entrée est un ensemble d'affectations correspondant chacune à l'une des icônes prédicatives de la séquence.

Le mécanisme de base de l'analyse de la séquence d'entrée doit aboutir à une interprétation de celle-ci. Il consiste donc en :

- (a) repérer toutes les icônes (potentiellement) prédicatives présentes dans la séquence ;

- (b) calculer pour chacune d'elles toutes les affectations possibles, triées par ordre de valeur (au sens de l'éq. 7.7) ;
- (c) unifier les différentes affectations obtenues en un graphe représentant une interprétation du sens global du message.

L'étape (a) est effectuée lors d'un premier parcours rapide de la séquence, où l'on regarde dans le dictionnaire, pour chaque icône, si elle a des sèmes extrinsèques ou non.

L'étape (b) consiste, pour chaque icône identifiée comme prédicative, à en construire toutes les affectations possibles, à calculer pour chacune sa valeur, puis à les ordonner par un algorithme de tri de type "*quicksort*" [Bratko, 1990].

Pour ce qui concerne enfin l'étape (c), l'unification est implicite (car les mêmes identificateurs PROLOG sont utilisés dans le calcul de chaque affectation) ; en revanche, trois opérations sont effectuées avant que le résultat de l'analyse ne soit rendu au module suivant :

- tout d'abord, on tente d'affecter d'éventuelles icônes restées isolées en surchargeant certaines places d'actant (§ 7.2.2.5 ; ces surcharges sont rendues, à la génération, par des coordinations, § 7.3.2) ;
- les icônes restées malgré tout isolées, sans avoir pu être affectées à aucun prédicat de la séquence, sont adjointes comme nœuds isolés au graphe résultat ; elles seront générées « à part » ;
- enfin des incohérences possibles (graphes cycliques) sont détectées et éliminées<sup>2</sup>.

Les affectations étant triées par ordre de valeur (eq. 7.7) décroissante, le programme PROLOG, lorsqu'il calcule toutes les interprétations possibles par produit cartésien des affectations des prédicats de la séquence, renvoie donc ses résultats dans un ordre correspondant. Ainsi si les affectations possibles pour la première icône prédicative de la séquence sont, dans l'ordre,  $A_{11}, A_{12}, \dots, A_{1m_1}$  ; si les affectations possibles pour la seconde icône prédicative sont, dans l'ordre,  $A_{21}, A_{22}, \dots, A_{2m_2}$  ; et ainsi de suite jusqu'à la  $n^{\text{ième}}$  et dernière icône prédicative, dont les affectations possibles sont  $A_{n1}, A_{n2}, \dots, A_{nm_n}$  ; alors l'ordre dans lequel apparaissent les interprétations possibles de la séquence est :

$$\begin{array}{cccc}
 A_{11}, & A_{21}, & \dots & A_{n1} \\
 A_{11}, & A_{21}, & \dots & A_{n2} \\
 & & & \vdots \\
 A_{11}, & A_{21}, & \dots & A_{nm_n} \\
 & & & \vdots \\
 A_{1m_1}, & A_{2m_2}, & \dots & A_{nm_n}
 \end{array}$$

Si nous considérons que la somme des valeurs des affectations de chaque icône prédicative constitue une grandeur représentant une sorte d'« harmonie sémanti-

---

<sup>2</sup>ces cycles sont la manifestation d'incohérences dans l'analyse ; toutefois, une exigence d'efficacité a fait choisir de les éliminer à l'aveugle sans chercher à lancer une deuxième passe d'analyse plus poussée.

que » globale de l'interprétation, alors cet ordre, sans assurer une décroissance de cette grandeur, assure que la première interprétation fournie en est au moins un maximum absolu (c'est la somme des maxima locaux).

#### 7.2.2.4 Catégories particulières

Deux catégories d'icônes ont un traitement spécial au cours de l'analyse.

- La première regroupe les icônes qui sont ignorées lors du calcul de la distance entre une icône prédicat et une icône candidate à une place d'actant. Les icônes de cette catégorie sont considérées comme « sans épaisseur », c'est-à-dire qu'on fait comme si elles n'introduisaient pas de distance syntagmatique entre les icônes entre lesquelles elles s'intercalent.

Cette catégorie regroupe les icônes des taxèmes '*qualifieurs*' (adjectifs comme '*bon*', '*drôle*', '*grand*' ...), '*modalités*' ('*non*', '*peut-être*', '*sûrement*'), '*mots simples*' (mots isolés comme '*bonjour*', '*merci*', '*pardon*' ...), '*temps*' ('*passé*', '*futur*') et '*emphase*' ('*interrogation*', '*exclamation*').

Le fait de considérer ces icônes comme « sans épaisseur » est un artifice qui traduit la récursivité du processus psychologique de prédication. En pratique, cette « astuce » permet de prendre en compte certains phénomènes présents dans les corpus, comme l'« entassement » d'adjectifs.

- La seconde, qui recoupe partiellement la première, est celle des prédicats qui ne peuvent eux-même en aucune circonstance devenir les actants d'un prédicat de plus haut niveau.

Ces icônes « non prédicables » sont les '*temps*', les '*modalités*', les '*emphases*' d'une part ; les '*expressions figées*' et les '*phrases à trous*' d'autre part.

Les trois premiers taxèmes sont des marqueurs d'inflexions sémantiques de haut niveau, qui s'appliquent déjà à des prédicats ; il est pratiquement inimaginable, dans un langage ordinaire, qu'ils puissent eux-mêmes devenir objets d'un autre prédicat<sup>3</sup>.

Les deux derniers regroupent des phrases toutes faites, sans actant ou complétables au plus par un actant optionnel, présentes dans l'interface de PVI pour permettre des saisies rapides de messages urgents du type « *attendez!* », « *je ne comprends pas* », ou « [*j'ai*] *un problème* », « [*je m'*] *appelle x* » ... Ces icônes forment des messages préformatés, et il ne convient pas qu'elles entrent elles-mêmes dans la composition de messages classiques.

#### 7.2.2.5 Coordination ; surcharge des rôles casuels

La première passe d'analyse (§ 7.2.2.3) fonctionne en cherchant une combinaison d'affectations pour chaque icône prédicative de la séquence d'entrée, donc à chaque fois une combinaison d'*applications* de l'ensemble des cas vers l'ensemble des icônes. Elle ne prend donc en compte qu'une seule icône par place d'actant.

<sup>3</sup>cela se traduirait par des phrases du type « *Je vois que l'action x s'est déroulée dans le passé* », ou « *Tu declares que la connaissance du fait y est entachée d'incertitude* » ...

Il existe pourtant, dans les corpus, de nombreux exemples d'énumérations, qu'il conviendrait de comprendre comme des surcharges d'un même rôle casuel par « entassement » d'actants homogènes.

Une deuxième passe d'analyse prend en charge ces phénomènes. Elle consiste à reprendre une à une les icônes qui n'ont pas été affectées lors de la première passe, et à calculer pour chacune d'elles quel serait son score d'affectation à tel ou tel actant de tel ou tel prédicat. Le meilleur de ces scores, s'il dépasse un seuil donné (*seuil\_icone\_isolee*), détermine alors une décision de réaliser l'affectation concernée.

On peut par suite avoir, dans le graphe résultant de cette analyse, une surcharge d'actants du type :

[ALLER : *agent*=MOI, *destination*=PLAGE, *agent*=PAPA, *agent*=MAMAN]

(résultat de l'analyse de MOI/ALLER/PLAGE/PAPA/MAMAN).

## 7.3 Le module de transfert lexical

### 7.3.1 Format de la représentation pivot

Le résultat d'une analyse est une interprétation de la séquence (§ 7.2.2.3), c'est-à-dire une séquence d'affectations du type  $\langle s_i, \langle c_j, s_{ij} \rangle \rangle$ . Ce format est topologiquement équivalent à un graphe conceptuel [Sowa, 1984]. Ainsi les interprétations [MANGER : *agent*=MOI, *objet*=VIANDE, *instrument*=FOURCHETTE] (« je mange la viande avec la fourchette »), et [VOIR : *agent*=MOI, *objet*=MANGER], [MANGER : *agent*=CHAT] (« je vois que le chat mange ») sont respectivement équivalentes aux graphes conceptuels représentés fig.7.10 et fig.7.11.

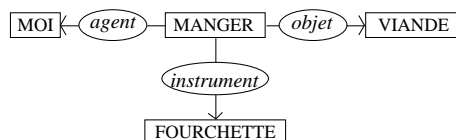


FIG. 7.10: Graphe conceptuel résultant de l'analyse de FOURCHETTE/MOI/MANGER/VIANDE

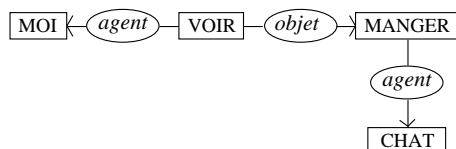


FIG. 7.11: Graphe conceptuel résultant de l'analyse de MOI/VOIR/CHAT/MANGER

Il y a néanmoins dans la représentation linéarisée du graphe une information supplémentaire, qui réside dans l'*ordre* dans lequel les affectations sont mises. Re-



trouver cette information dans le graphe conceptuel équivaldrait à en numéroté les concepts.

Cet ordre a une importance en ce qu'il reflète en partie l'ordre originel de la séquence d'icônes d'entrée. Celui-ci peut éventuellement avoir une signification, non en tant que manifestation d'une structure syntaxique, mais en tant que la *topicalité* du message, c'est-à-dire l'ordre d'importance des concepts exprimés.

Cet ordre est préservé à travers l'ordre des icônes prédicatives de la séquence d'entrée. Ainsi l'analyse des séquences MOI/VOIR/CHAT/MANGER et CHAT/MANGER/VOIR/MOI conduit-elle respectivement aux interprétations [VOIR : *agent*=MOI, *objet*=MANGER], [MANGER : *agent*=CHAT], et [MANGER : *agent*=CHAT], [VOIR : *agent*=MOI, *objet*=MANGER]. Ces deux représentations linéaires sont de la même façon topologiquement équivalentes au graphe de la figure 7.11, mais la première reflète l'attention portée sur l'action VOIR, alors que la seconde reflète l'attention sur l'action MANGER.

L'information de topicalité du message, présente dans cette représentation pivot, n'est malheureusement pas assez souvent transcrite au niveau de la phrase générée, car le module de génération de PVI est trop rudimentaire pour mettre en œuvre toutes les tournures par lesquelles la langue française peut traduire ce type d'information (parmi lesquelles l'une des plus importantes est par exemple la voix passive, qui n'est pas gérée par le module de génération). Pour un exemple de traduction de la topicalité au niveau de la phrase française, voir §7.4.2.2.

### 7.3.2 Format du réseau sémantique linguistique

Le graphe conceptuel « orienté » qui constitue la représentation fournie à la sortie du module d'analyse est converti en un format adapté à l'entrée du module de génération. Ce format est une sorte de nouveau graphe conceptuel « enrichi », où à chaque noeud conceptuel est attaché une place libre réservée à des informations morphosyntaxiques. Ces informations ne sont pas instanciées à l'entrée du module de génération, mais s'instancient au fur et à mesure de l'application de règles d'accord et de recton.

Les noeuds conceptuels du graphe en sortie de l'analyse sont des entrées du dictionnaire d'icônes, alors que les noeuds conceptuels du nouveau graphe doivent être des entrées du lexique linguistique utilisé pour la phase de génération. Cela signifie, en théorie, que le sens pourrait en outre être redistribué sur les concepts, reflétant en cela des différences entre les sémantiques des deux langages, le langage d'icônes utilisé en entrée, et la langue utilisée en sortie. Ce phénomène est un poncif de la traduction : ainsi, en traduisant l'allemand « *über den Fluß schwimmen* » par le français « *traverser la rivière à la nage* »<sup>4</sup>, le sème /*nage*/ passe du verbe principal à un complément circonstanciel, tandis que dans le même temps le sème /*traversée*/, d'une préposition spatiale, est réintégré au verbe principal. Cependant, dans l'application PVI, ce phénomène n'est quasiment pas représenté. En effet le code pictographique utilisé comme source principale de la base de données lexicale d'icônes (le code « Commun-I-Mage ») a déjà, à la base, été conçu comme un catalogue

<sup>4</sup>Exemple emprunté à Bertil Malmberg.

de transcriptions pictographiques de lexèmes français.

Un exemple d'utilité actuelle du module de transfert lexical est la transformation de la coordination. Celle-ci se manifeste à la sortie de l'analyse comme une surcharge d'actants. Elle est transformée, dans le réseau linguistique fourni en entrée au module de génération, en un nouveau nœud prédicatif, étiqueté “*et*”, et dont les actants sont les différents nœuds coordonnés qui se trouvaient originellement dans le même rôle casuel. Ainsi un graphe du type [ALLER : *agent*=MOI, *destination*=PLAGE, *agent*=PAPA, *agent*=MAMAN], devient-il un graphe [ALLER : *agent*=ET, *destination*=PLAGE], [ET : *coordonne1*=MOI, *coordonne2*=PAPA, *coordonne3*=MAMAN]. Ce nœud “*et*” est destiné lors de la phase de génération à devenir un groupe nominal subordonnant plusieurs groupes nominaux plus simples entre lesquels s'intercalent des conjonctions de coordination « *et* ».

L'existence de la couche de transfert lexical est en tout état de cause un élément important qui assure l'indépendance des modules d'analyse et de génération. Même si cette couche joue un rôle ténu dans la version actuelle de PVI, elle pourrait s'étoffer si un changement du langage d'icônes utilisé en entrée rendait cette évolution nécessaire. Réciproquement, garder le module d'analyse tel qu'il existe dans l'application actuelle et « brancher », à la place du module de génération en français, un module de génération dans n'importe quelle autre langue, ne demande pas d'autre évolution qu'une modification du système de transfert lexical.

## 7.4 Le module de génération

### 7.4.1 Le lexique TAG

La langue dans laquelle les messages sont générés est modélisée par un « lexique-grammaire » qui englobe tant l'information lexicale que l'information grammaticale servant à la production de phrases.

Le formalisme utilisé pour la représentation est celui des grammaires d'arbres adjoints (ou « TAG » : *Tree Adjoining Grammar*) [Joshi *et coll.*, 1975, Abeillé, 1991, Abeillé, 1993]. Chaque entrée lexicale est représentée dans ce formalisme par au moins un arbre élémentaire dont l'une des feuilles, appelée *l'ancre*, est instanciée. Cette ancre correspond à l'élément lexical : c'est le mot français qui exprime cette entrée. L'arbre syntaxique élémentaire au sein duquel elle s'inscrit correspond à la tournure en laquelle elle s'emploie. Ainsi aucun lexème n'est-il représenté dans le dictionnaire sans un minimum d'information sur les structures grammaticales dans lesquelles il s'insère.

Un exemple minimal d'arbre élémentaire correspondant à une entrée lexicale est montré fig.7.12. Il s'agit du groupe nominal lexicalisant le concept ‘*papa*’.

#### 7.4.1.1 Les arbres élémentaires et l'opération de substitution

Tous les exemples ne sont pas aussi simples que celui qui précède. L'intérêt de stocker l'information lexicale dans des arbres syntaxiques est de disposer, au niveau

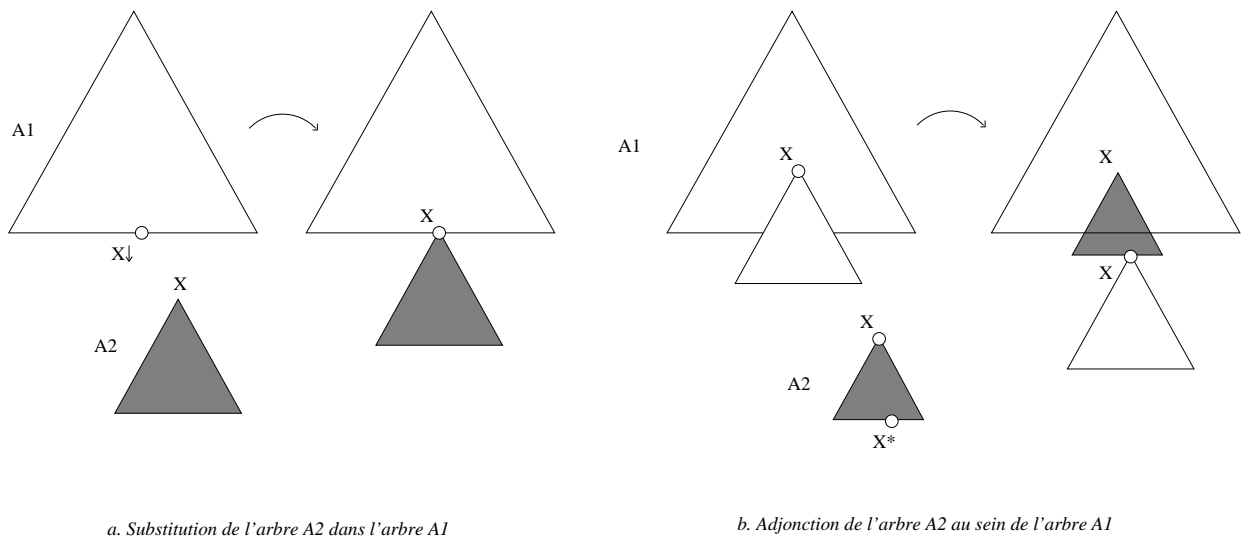


FIG. 7.12: L'arbre élémentaire correspondant au lexème "Papa"

même de l'entrée lexicale, de la connaissance sur la façon dont la langue opère pour exprimer les relations microsémantiques du sémème avec son contexte.

Il s'agit en somme de répondre à la question suivante : connaissant un concept, comment l'exprimer dans un syntagme français correct ; et si ce concept est prédicatif, comment exprimer dans un syntagme français correct ce concept et ses relations actanciennes ?

Les arbres élémentaires correspondant à des sémèmes prédicatifs contiennent la réponse à cette question sous la forme de *nœuds non instanciés* qui peuvent s'unifier avec des syntagmes correspondant aux actants. L'opération d'unification mise en œuvre est appelée *substitution* (fig.7.13.a).



a. Substitution de l'arbre A2 dans l'arbre A1

b. Adjonction de l'arbre A2 au sein de l'arbre A1

FIG. 7.13: Les deux opérations définies sur les arbres dans le modèle TAG

Ainsi le sémème 'donner' s'exprime-t-il souvent en français par le verbe "donner"<sup>5</sup> dans une tournure du type : « X donne Y à Z » ; où X, Y et Z sont des groupes nominaux exprimant respectivement l'actant *agent*, l'actant *objet* et l'actant *récepteur* du sémème. L'entrée du lexique correspondant à cette tournure est l'arbre élémentaire représenté fig.7.14.

Si l'agent de "donner" se trouvait être "papa", une unification de l'arbre élémen-

<sup>5</sup>il y a aussi le substantif "don", qui permet de regrouper le concept et ses actants sous un syntagme nominal, à condition qu'il n'y ait pas de nuance aspectuelle ou temporelle à exprimer, et que l'on puisse ou veuille éluder l'un des actants *agent* ou *objet*.

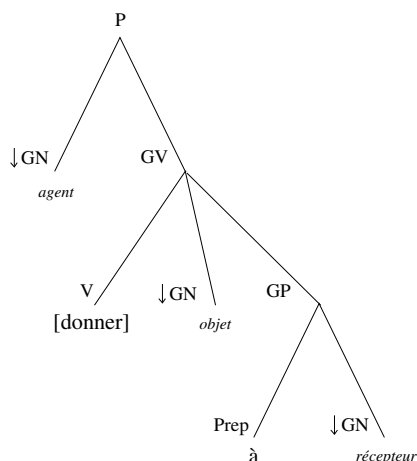


FIG. 7.14: Un arbre élémentaire correspondant à la tournure « *X donne Y à Z* »

taire exprimant “*papa*” (dont la tête est un nœud de type GN) serait possible avec le GN non instancié correspondant à l’agent, X, dans l’arbre fig.7.14. L’opération de *substitution*, à partir de ces deux arbres élémentaires, donnerait alors l’arbre représenté fig.7.15.

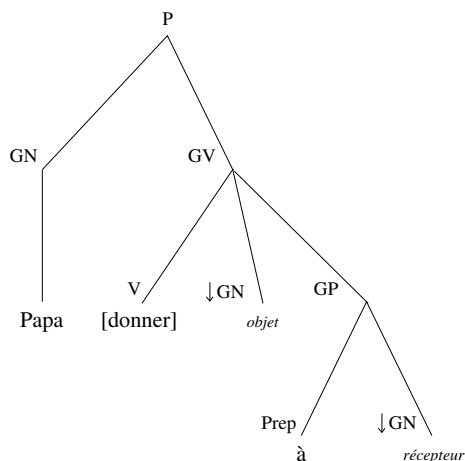


FIG. 7.15: L’arbre correspondant à « *Papa donne Y à Z* » (il reste deux places à substituer)

Le lexique contient parfois plusieurs arbres élémentaires pour une seule entrée ; ces différents arbres correspondent à des tournures différentes en lesquelles le sémème peut se lexicaliser. Ainsi les deux arbres rangés sous l’entrée “*voir*” (fig.7.16.a. et b.) permettent-ils d’exprimer tous deux le sémème “*voir*” avec un agent et un objet ; mais le premier convient à un objet pouvant s’exprimer sous forme de groupe nominal (par exemple « *Papa voit le chat* »), tandis que le second convient à un objet pouvant s’exprimer sous forme de phrase (par exemple « *Papa voit que le chat mange* »).

Ce modèle de stockage lexicalisé des connaissances grammaticales réemploie deux fois une information quasiment identique lorsque deux verbes, par exemple, s’expriment avec la même tournure (« *X donne Y à Z* », « *X prête Y à Z* »). Mais vue la disponibilité immédiate de l’information qu’il apporte, il ne pourrait devenir

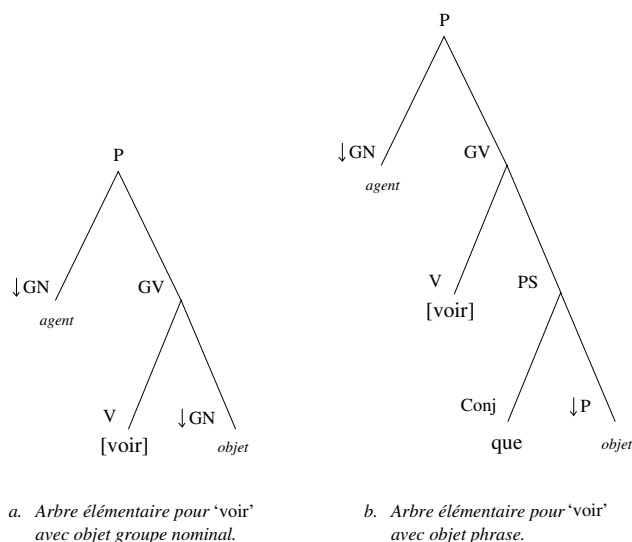


FIG. 7.16: Deux arbres élémentaires pour le sémème “voir”

rentable de tenter de faire des économies sur ce point que si l'on avait à gérer d'importantes classes de verbes à emploi similaire. Ce n'est pas le cas dans l'application PVI.

#### 7.4.1.2 Les arbres auxiliaires et l'opération d'adjonction

Le modèle TAG prévoit une opération d'unification complexe capable de rendre compte des modifications et des ajouts : l'*adjonction* [Abeillé, 1993]. Cette opération consiste en quelque sorte en deux substitutions simultanées ; plus précisément, elle permet d'insérer un arbre, dit *auxiliaire*, au niveau d'un nœud intermédiaire d'un arbre syntaxique. La tête de l'arbre auxiliaire s'unifie au niveau de ce nœud intermédiaire, tandis qu'en un nœud particulier, le  *pied*  de l'arbre auxiliaire, se rebranche la partie de l'arbre d'origine située en aval du nœud intermédiaire (fig.7.13.b).

La modélisation sous forme d'arbres élémentaires auxiliaires convient particulièrement aux entrées lexicales qui ne constituent pas en elles-mêmes des têtes sémantiques, mais qui apportent juste une modification à un autre sémème : adverbess, adjectifs ... Ainsi l'arbre élémentaire correspondant à “*peut-être*” est-il représenté fig.7.17.

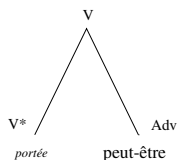


FIG. 7.17: L'arbre élémentaire auxiliaire correspondant à “peut-être”

Dans le cas des adjectifs, on peut trouver sous la même entrée lexicale à la fois un arbre élémentaire « initial », utilisé lorsque l'adjectif est la tête sémantique d'une phrase, (fig.7.18.a), et un arbre élémentaire auxiliaire, utilisé lorsque l'adjectif n'est le modifieur d'un actant plus important dans le message (fig.7.18.b).

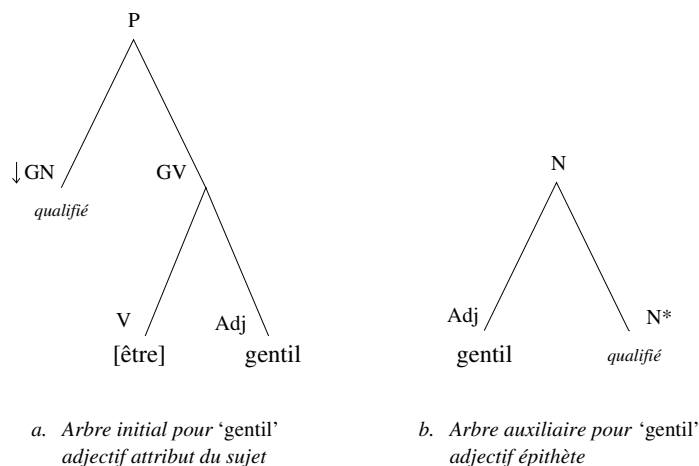


FIG. 7.18: Deux arbres élémentaires pour l'adjectif "gentil"

### 7.4.1.3 Arbres non terminaux

Il convient de noter que notre modèle du lexique TAG n'est pas entièrement lexicalisé, contrairement à celui que Schabes ou Abeillé utilisent pour l'analyse [Schabes *et coll.*, 1988]. Il contient des arbres non-terminaux que nous utilisons pour factoriser des tournures grammaticales systématiques; ainsi le complément instrumental peut, dans le cas général — s'il n'est pas spécifié dans l'arbre élémentaire du verbe auquel il s'applique — s'exprimer par un groupe prépositionnel « avec X ». Cette tournure est représentée dans le lexique par l'arbre de la figure 7.19, auquel le groupe nominal complément d'instrument est substitué (nœud en bas à droite), tandis que l'arbre dérivé est lui-même adjoint à un autre arbre, au niveau du nœud correspondant au prédicat.

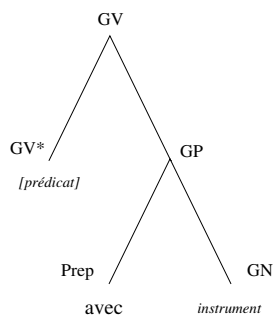


FIG. 7.19: La formation du complément d'instrument avec "avec"

D'autre part, au cours de la génération, il est possible qu'un prédicat, considéré comme annexe relativement à un autre, et normalement exprimable par un arbre P (phrase) complet uniquement, vienne s'adjoindre sous la forme d'une proposition subordonnée relative enchâssée dans un groupe nominal à la phrase du prédicat « principal ». Ceci est rendu possible par l'existence de procédures de transformation régulière d'arbre P en arbre CP (complément propositionnel), du type :

« le chat mange la viande » → « [le chat] qui mange la viande »  
 « le chat mange la viande » → « [la viande] que le chat mange »

Nous reviendrons sur ces mécanismes en §7.4.2.2. Il reste que ce lexique TAG non entièrement lexicalisé se révèle d'une particulière souplesse pour les besoins d'un module de génération.

#### 7.4.1.4 Les règles d'accord

Les mécanismes d'unification de traits proposés dans le modèle TAG n'ont pas été ici implantés rigoureusement. Les phénomènes d'accord sont implantés de façon plus « efficace », mais moins « propre », sous la forme de buts PROLOG gélés ([PrologIA, 1993], §2.2, p. R 2-9).

Un but gelé, en PROLOG, est un but qui conditionne le résultat final du but qui l'appelle, mais qui n'entrave pas le déroulement de la résolution des buts suivants. Il reste en attente de résolution tant qu'une certaine variable, que l'on précise lors de l'appel, n'est pas instanciée. Lorsque la variable s'instancie, la résolution du but reprend normalement, et conduit à une réponse définitive (solutions possibles ou échec).

Le but gelé permet donc en quelque sorte de fournir une instruction au système, en lui laissant la possibilité de ne s'en préoccuper que lorsqu'il aura réuni les données suffisantes pour le faire.

Dans le cas qui nous intéresse, les règles d'accord sont codées sous la forme de conditions posées à l'émergence même d'une solution au prédicat de base de données. Ainsi, alors qu'un arbre initial simple, comme celui de “*papa*” (fig.7.12), est une solution inconditionnelle au prédicat de base de données `arbre` :

```
arbre(<lex_papa,n>,<<GN,
                <<Nom,<lex_papa,n>>>>,
      nil>) ->;
```

un arbre initial, en revanche, qui subordonne des groupes devant s'accorder, sera codé avec une condition gelée (donc en partie droite de la clause de Horn) correspondant à l'instanciation de l'accord. Ainsi pour “*donner*” (fig.7.14) :

```
arbre(<lex_donner,n>,<<Ph,
                <<GN,_n0>,
                <GV,
                <<Vb,<lex_donner,n>>,
                <GN,_n1>,
                <GP,_n2>>>>>,
      <agent,<GN,_n0>>.<objet,<GN,_n1>>.<recepteur,<GP,_n2>>.nil>) ->
freeze(_n0,nominatif(_n0))
freeze(_n1,accusatif(_n1))
freeze(_n2,datif(_n2))
freeze(_n0,accord(<GN,_n0>,<Vb,<lex_donner,n>>));
```

On voit dans cet exemple quatre buts gelés :

- d'une part,

```
freeze(_n0,nominatif(_n0))
freeze(_n1,accusatif(_n1))
freeze(_n2,datif(_n2))
```

correspondent à des règles de sélection de cas syntaxiques qui ne servent bien entendu, en français, qu'aux pronoms. Ainsi *nominatif*, *accusatif*, *datif*, s'appliquent-ils trivialement à tout groupe nominal composé d'un substantif, mais déterminent en revanche, quand il s'agit par exemple du pronom personnel de la première personne du singulier, respectivement les formes « *je* », « *me* », et « *me* ». Ces buts dépendent de l'instanciation des variables correspondant aux nœuds GN à substituer.

– d'autre part,

```
freeze(_n0,accord(<GN,_n0>,<Vb,<lex_donner,n>>))
```

correspond à l'accord du verbe avec le sujet. Il dépend de l'instanciation de la variable correspondant au groupe nominal sujet. Lorsque celui-ci s'instancie (c'est-à-dire lorsque l'opération de substitution est effectuée sur ce nœud), la résolution de ce but *accord* est relancée. Elle va aboutir à aller chercher les traits morphosyntaxiques *nombre* et *personne* du GN sujet, et à les instancier ainsi pour le verbe.

La règle *accord/2* est une règle complexe, qui s'oriente vers des actions différentes selon qu'il s'agit d'accorder un verbe avec un nom, un adjectif avec un nom . . . . Elle sait dans quel cas il s'agit d'accorder en personne et en nombre (verbes), en genre et en nombre (adjectifs) . . .

Dans tous les cas, les données qui s'instancient tout d'abord sont celles qui ont un ancrage lexical : ainsi le nom commun a-t-il de façon univoque, dans le lexique morphologique (§ 7.4.3.2), un trait de genre et un trait de personne (c'est toujours la troisième)<sup>6</sup>. Dès que cet ancrage lexical est trouvé, les buts *accord* en attente se dégèlent successivement, au fur et à mesure de l'instanciation des traits morphosyntaxiques sur les nœuds terminaux de l'arbre, pour arriver en fin de compte à une instanciation de ces traits sur l'ensemble de l'arbre.

→ *Les seuls phénomènes d'unification relativement complexes qui justifient le modèle des traits d'unification amont et aval en TAG [Vijay-Shanker & Joshi, 1988] sont les opérations d'adjonction qui modifient la structure de traits des lemmes situés sous le nœud où se fait l'adjonction. Ainsi le passé composé est-il modélisé par une adjonction de verbe auxiliaire sur un nœud V. Certains traits de nombre et de personne sont reportés sur le participe passé, qui change en outre de mode verbal ; l'auxiliaire, lui, reçoit les traits de personne et de nombre du sujet, ainsi que le mode verbal du verbe principal avant l'adjonction. Ce phénomène, l'accord des participes, n'est actuellement pas correctement modélisé dans PVI.*

<sup>6</sup>Le trait nombre est dans cette version de PVI également toujours fixé — la plupart du temps au singulier —, car la marque du pluriel n'est pas traitée en analyse. Il existe donc également des noms fixés au pluriel, qui correspondent à des collectifs (p.ex. “*fruits*”).



## 7.4.2 La hiérarchisation du réseau en arbres

La génération de phrases en français à partir d'une représentation du sens consiste en fin de compte à transformer un graphe en un (ou plusieurs) arbres. Ce processus, qui est donc une *hiérarchisation* de graphe, est décrit dans les paragraphes qui suivent.

### 7.4.2.1 Le premier parcours du graphe

Une première passe permet d'engendrer à partir du graphe un premier arbre syntaxique, qui correspond à l'expression simple du premier prédicat et de ses arguments, sans ajout ni modifieur.

Il faut rappeler que le graphe se présente sous la forme d'une liste de structures prédicat-liste d'arguments; ainsi un graphe relativement complexe comme celui de la figure 7.20 (« *Papa voit que le gentil chat a mangé la viande* ») est-il représenté linéairement par la séquence [VOIR : *agent*=PAPA, *objet*=MANGER], [GENTIL : *qualifie*=CHAT], [MANGER : *agent*=CHAT, *objet*=VIANDE], [AVANT : *quand*=MANGER] :

```
voir(agent(papa).objet(manger).nil).gentil(qualifie(chat).nil).\
manger(agent(chat).objet(viande).nil).avant(quand(manger).nil).nil
```

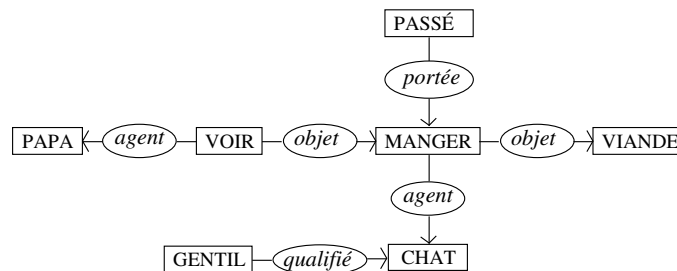


FIG. 7.20: Graphe conceptuel résultant de l'analyse de PAPA/VOIR/CHAT/GENTIL/MANGER/AVANT/VIANDE

La première passe de hiérarchisation correspond donc à la recherche d'un *arbre de couverture* du graphe, dont le nœud tête, celui par lequel commence le parcours du graphe, est le premier prédicat de la liste (celui qui arrive en tête dans l'ordre de topicalité). Ainsi, la première passe sur le graphe [MANGER : *agent*=CHAT, *objet*=VIANDE], [AVANT : *quand*=MANGER] donne-t-elle l'arbre correspondant à la phrase « *le chat mange la viande* ». La première passe sur le graphe [VOIR : *agent*=PAPA, *objet*=MANGER], [GENTIL : *qualifie*=CHAT], [MANGER : *agent*=CHAT, *objet*=VIANDE], [AVANT : *quand*=MANGER] donne l'arbre correspondant à la phrase « *Papa voit que le chat mange la viande* ».

Ce parcours se réalise par appel récursif de la règle de hiérarchisation lorsqu'elle se porte sur des actants qui sont eux-même des prédicats (tels MANGER qui, dans l'exemple ci-dessus, est à la fois l'objet de “*voir*” et la tête sémantique d'une autre proposition).

En effet, l'opération atomique de conversion de la structure de graphe (§ 7.3.2) en structure d'arbre syntaxique (§ 7.4.1) est la sélection, pour un nœud donné du

graphe, d'un arbre élémentaire du lexique-grammaire TAG exprimant ce nœud et ses éventuels actants (exemple fig. 7.21).

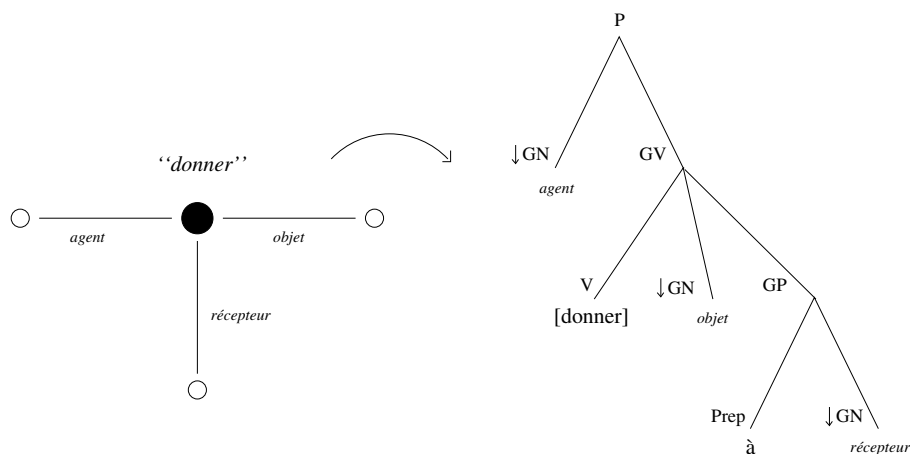


FIG. 7.21: Sélection d'un arbre TAG élémentaire pour un nœud prédicat du graphe

À chaque étape de la première passe de hiérarchisation du réseau, le programme transforme donc un nœud, le nœud « courant », en arbre syntaxique élémentaire, puis appelle récursivement la fonction de hiérarchisation sur les nœuds qui se trouvent eux-mêmes au bout des relations casuelles du nœud courant.

Le processus PROLOG essaie ainsi un par un, au nœud  $n$ , tous les arbres élémentaires du lexique TAG correspondant à l'entrée  $n$ , jusqu'à ce qu'il en trouve un lui permettant d'aboutir avec succès à un arbre de couverture (partielle) du graphe : à chacune de ces sélections, il essaie d'appliquer récursivement le même traitement aux nœuds situés directement en aval de  $n$  :  $n_1^1, n_2^1, \dots, n_m^1$ . La génération réussie de l'arbre au niveau  $n$  dépend alors de la génération réussie des arbres lexicalisant  $n_1^1, n_2^1, \dots, n_m^1$ , puis de leur unification réussie sur les nœuds à substituer dans l'arbre de  $n$ . Si cette unification ne peut aboutir pour tous les  $n_i^1$ , alors on essaie, s'il y en a, un autre arbre élémentaire lexicalisant  $n$  — et ainsi de suite jusqu'à épuisement de la liste. Ce processus est illustré sur la figure 7.22.

Ainsi lorsqu'il tente de lexicaliser le nœud “voir” (fig.7.22, opération 1), le programme tente-t-il tout d'abord le premier arbre élémentaire qui lui est fourni par le lexique TAG à l'entrée “voir” : l'arbre représenté fig.7.16.a. C'est lorsqu'il échoue à unifier la partie du graphe représentant l'objet de “voir” (“manger” et ses actants) à un nœud de substitution étiqueté GN, qu'il abandonne finalement cet arbre élémentaire pour essayer — avec plus de succès — l'arbre de la figure 7.16.b.

Certains actants, lorsque leur type de relation casuelle avec le prédicat n'est pas inclus dans l'une des représentations de l'entrée lexicale de celui-ci, sont également traités dans cette phase de premier parcours, mais avec une méthode différente : ils sont intégrés par substitution à un arbre auxiliaire non-terminal, du type de celui qui représente les compléments d'instrument (fig.7.19), et l'arbre dérivé est lui-même adjoint à l'arbre de la phrase en cours au nœud courant (qui représente le prédicat).

Cette première passe de hiérarchisation permet ainsi d'aboutir, à partir du premier prédicat du graphe, à une phrase qui est déjà au moins un squelette du message final.



Dans la mesure du possible, l'ordre de topicalité du message est conservé lors de la génération. On en voit un exemple lorsque l'on compare les réponses fournies par PVI à la séquence CHAT/MANGER/OISEAU/GENTIL et à la séquence OISEAU/GENTIL/MANGER/CHAT, respectivement :

```
> test(chat.manger.oiseau.gentil.nil);
Séquence : chat.manger.oiseau.gentil.nil
Graphe   : manger(agent(chat).objet(oiseau).nil).gentil(qualifie(oiseau)\
au).nil)
Phrase   : Le chat mange le gentil oiseau. (2417 ms)

> test(oiseau.gentil.manger.chat.nil);
Séquence : oiseau.gentil.manger.chat.nil
Graphe   : gentil(qualifie(oiseau).nil).manger(agent(chat).objet(oiseau)\
au).nil)
Phrase   : L'oiseau que le chat mange est gentil. (2317 ms)
```

Dans les deux cas, l'analyse aboutit au graphe représenté fig.7.23 (car les « contraintes » sémantiques conservées dans le dictionnaire d'icônes interdisent que ce soit l'oiseau qui mange le chat). Mais dans le premier cas, la topicalité du message porte d'abord sur l'action de *manger* (le prédicat MANGER apparaissait avant le prédicat GENTIL dans la séquence d'entrée), alors que dans le second cas, elle porte d'abord sur l'état de *gentil*-lesse de l'oiseau.

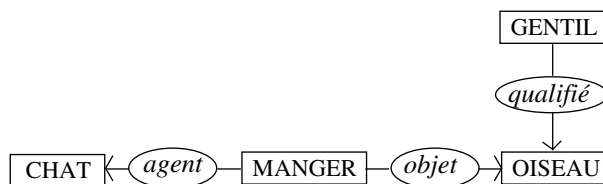


FIG. 7.23: Graphe conceptuel résultant de l'analyse de OISEAU/GENTIL/MANGER/CHAT et de CHAT/MANGER/OISEAU/GENTIL

À la génération, le premier parcours de hiérarchisation du réseau va donc aboutir, pour le premier cas, à « *Le chat mange l'oiseau* », et pour le second cas, à « *l'oiseau est gentil* ». Les prédicats restant sont alors exprimés dans les deux cas par adjonction à la phrase déjà formée ; adjonction d'un adjectif épithète dans le premier cas, adjonction d'une subordonnée relative dans le second cas. Ce qui aboutit finalement aux phrases : « *le chat mange le gentil oiseau* », et « *l'oiseau que le chat mange est gentil* ».

#### 7.4.2.3 Génération de plusieurs arbres

Lorsque le programme a déjà engendré une phrase correspondant au premier parcours du graphe, il essaye, on l'a vu, d'exprimer l'information restante par des adjonctions dans la même phrase. Il dispose en outre d'une autre stratégie, qui est de commencer une nouvelle phrase pour le nouveau prédicat à exprimer.

En fait, comme le langage PROLOG est non-déterministe, toutes les solutions possibles sont systématiquement calculées. Ainsi un graphe comme [VOIR : *agent*=MOI, *objet*=CHAT], [GENTIL : *qualifié*=CHAT] peut-il s'exprimer par la phrase « *je vois le gentil chat* », ou, secondairement, par « *je vois le chat ; le chat est gentil* »<sup>8</sup>. Bien entendu, l'interface ne peut présenter, elle, qu'une seule sortie à la fois, et c'est donc dans ce cas la première phrase qui sera, seule, affichée et transmise à la synthèse vocale.

Dans certains cas, le choix n'existe pas et le système doit générer son message en plusieurs phrases. Ainsi lorsqu'il faut exprimer deux propositions de même niveau, dont aucune n'est subordonnée à l'autre, et qui ont un actant en commun. Dans un tel cas, on ne dispose, d'une part, ni de la possibilité d'exprimer l'une de ces propositions sous la forme d'une subordonnée relative, ni d'autre part de la possibilité de coordonner des groupes verbaux. Ainsi l'analyse de MOI/MANGER/BOIRE aboutit-elle nécessairement à « *je mange ; je bois* ».

Lorsque le système crée un arbre correspondant à une nouvelle phrase, alors qu'il en a déjà engendré au moins une pour exprimer le même graphe, il s'assure également que le groupe nominal sujet de cette nouvelle phrase n'a pas déjà été formulé tel quel dans l'une des phrases précédentes. Si c'est le cas, alors dans la nouvelle phrase, ce groupe nominal est transformé en pronom sujet.

### 7.4.3 La linéarisation de l'arbre

La phase la plus triviale de la génération, enfin, est de transformer l'arbre, ou la forêt d'arbres syntaxiques, en une chaîne de caractères qui puisse être affichée à l'écran et envoyée au module de synthèse vocale.

#### 7.4.3.1 Recueil des nœuds terminaux

Un parcours en profondeur de l'arbre permet de construire la liste des nœuds terminaux. Ceux-ci sont des éléments lexicaux du type (Étiquette de catégorie, (Entrée lexicale, (Identificateur, (Liste de traits morphosyntaxiques)))

#### 7.4.3.2 Les entrées du lexique morphologique

Grâce à ces informations, on peut aller récupérer dans le lexique morphologique la forme de surface qui correspond à la déclinaison adéquate de l'entrée lexicale. Voici un exemple de déclinaisons d'un lemme telles qu'elles se présentent dans le lexique des formes morphologiques :

```
lexical(<Adj,lex_gentil>,"gentil",adj.mas.sing.nil) ->
```

---

<sup>8</sup>Il n'y a pas dans ce système de nuance entre la ponctuation « ; » et la ponctuation « . ». Le point-virgule dans cet exemple sépare bel et bien deux phrases différentes ; son choix est en fait imposé par les caractéristiques de certains systèmes de synthèse vocale placés en aval, et qui considèreraient le point comme une marque de fin de séquence à synthétiser, oubliant ainsi la deuxième phrase.

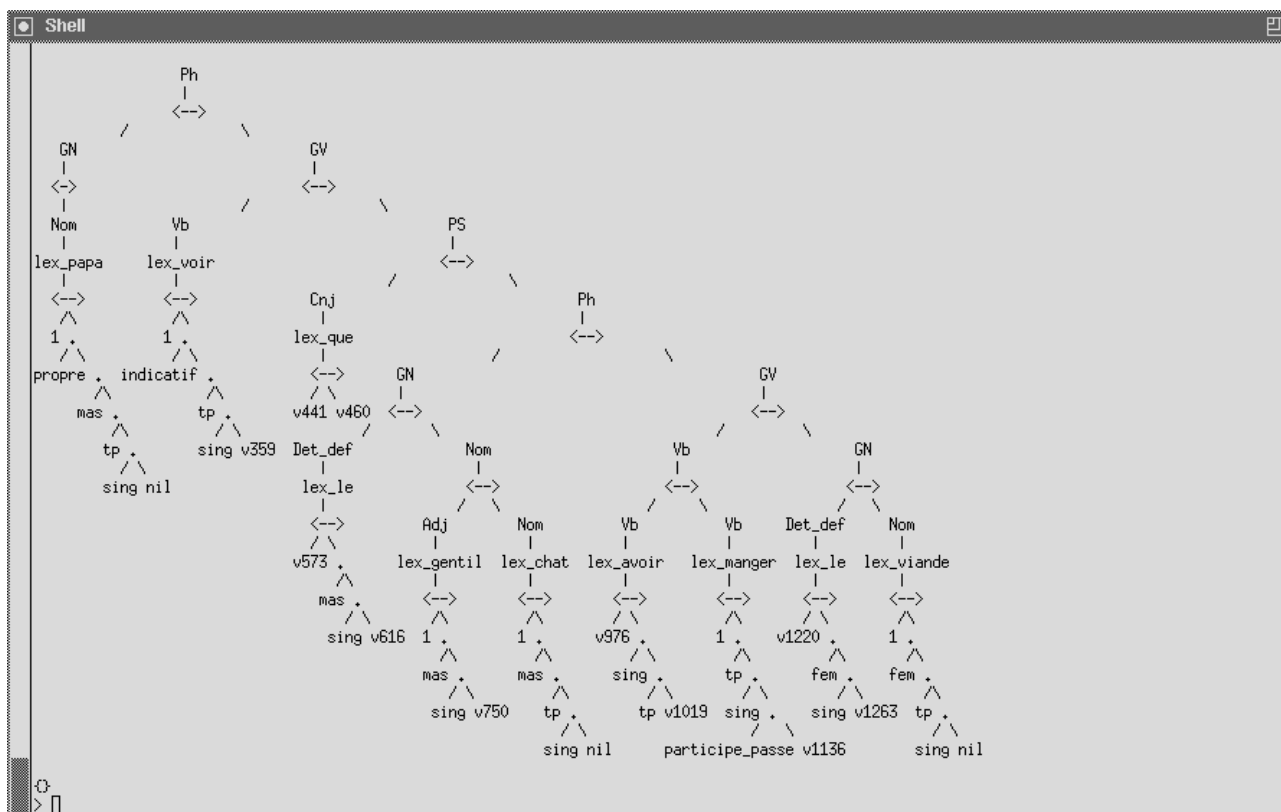


FIG. 7.24: Arbre engendré par le module de génération pour le graphe fig.7.20

```
lexical(<Adj,lex_gentil>,"gentille",adj.fem.sing.nil) ->;
lexical(<Adj,lex_gentil>,"gentilles",adj.fem.plur.nil) ->;
lexical(<Adj,lex_gentil>,"gentils",adj.mas.plur.nil) ->;
```

### 7.4.3.3 Élision, majuscules, ponctuation

Enfin, de dernières retouches « cosmétiques » sont apportées à la concaténation des formes de surface trouvées dans le lexique morphologique :

- Les formes de certains pronoms, prépositions ou conjonctions sont éventuellement modifiées pour former élision devant voyelles;
- Les phrases sont terminées par des point-virgules (cf. note p. 235) et commençées par des majuscules.

## 7.5 Évaluation et discussion

### 7.5.1 Banc d'essai

Une première évaluation du système a été menée suivant le principe du banc d'essai, c'est-à-dire « en laboratoire » : un ensemble de 200 séquences d'icônes,

correspondant dans leur structure à des expressions produites spontanément en situation réelle<sup>9</sup>, a été soumis à PVI pour interprétation et génération. Concrètement, c'est un fichier contenant la liste des séquences d'identificateurs qui a été fourni en entrée au système. Le but de ce banc d'essai était de tester les performances du programme de traduction sur des entrées formatées, indépendamment de l'interface.

Les résultats fournis par le programme ont été classés en quatre catégories :

- I Bonne interprétation, bonne génération ;
- II Bonne interprétation, génération mauvaise ou maladroite ;
- III Interprétation incomplète ou maladroite ;
- IV Interprétation fautive ou dépourvue de sens.

Les résultats du classement ont été les suivants :

- Catégorie I : 147 séquences ;
- Catégorie II : 15 séquences ;
- Catégorie III : 15 séquences ;
- Catégorie IV : 18 séquences.

Ceci signifie que l'ensemble des phrases correctement traitées représente 73,5% du total. Ou encore, si l'on décide de compter comme « acceptables » les séquences qui ont été soit correctement interprétées et générées, soit à la rigueur correctement interprétées mais imparfaitement générées (c'est-à-dire en groupant les catégories I et II), que le *taux d'acceptabilité* de PVI, sur ce banc de tests, est de 80,5%.

Ce score sur banc d'essai n'a toutefois absolument pas la valeur d'un taux d'acceptabilité par l'utilisateur, qu'il revient à ce dernier d'évaluer.

## 7.5.2 **Évaluation sur site**

L'évaluation sur site a été conduite par des orthophonistes et des ergothérapeutes au CRRF de Kerpape de décembre 1995 à mai 1996. Un prototype, livré avec une version de test de l'interface, contenant 152 icônes, a été testé par chacun des quatre utilisateurs pendant une période d'un mois. Un retour d'expérience plus qualitatif a pu à cette occasion être recueilli, et a enseigné en premier lieu qu'il restait encore beaucoup à faire pour transformer PVI en un produit acceptable.

Il a tout d'abord été souligné qu'un taux d'erreur donné pouvait paraître beaucoup moins acceptable en conditions réelles d'utilisation que sur banc d'essai. Il faut garder à l'esprit qu'une personne souffrant de handicaps moteurs, qui accède aux icônes par le truchement d'un dispositif de désignation à défilement colonne/ligne, peut passer plusieurs minutes à composer une séquence complète. Il est compréhensible dans ces conditions que toute réponse insatisfaisante soit ressentie par lui comme extrêmement frustrante. Des améliorations ergonomiques ont été suggérées ;

---

<sup>9</sup>On s'est inspiré de séquences fournies par les corpus du centre de Kerpape, en adaptant éventuellement le vocabulaire pour se replacer dans l'ensemble couvert par le lexique d'icônes (e.g. *Maman acheter moi classeur* devient *Maman donner moi livre*).

nous avons ainsi en fin de compte, dans la dernière version du système, remplacé les messages d'erreur initialement programmés par de simples transcriptions linéaires de la séquence d'entrée brute. Ce type de résultat, s'il ramène finalement à celui que donnerait un simple système de désignation sans traitement intelligent, est en tout cas très largement mieux toléré par les utilisateurs que le « message incompris » des premières versions.

Pour ce qui est du module linguistique proprement dit, un certain nombre d'erreurs sont apparues fréquemment, essentiellement dues au module d'analyse (sur 34 occurrences d'erreurs pendant les évaluations, 3 seulement étaient attribuables au module de génération, et encore ont-elles pu être corrigées sans difficulté).

En définitive, il reste en effet dans PVI un problème central, qui est celui de l'équilibre (fragile) que la conception du lexique doit permettre d'installer entre la précision d'interprétation et le pouvoir expressif. Les contraintes imposées à l'interprétation par les restrictions de sélection (modélisées ici par les traits sémantiques extrinsèques), peuvent se révéler dans certains contextes trop faibles — et conduire à des affectations d'actants inappropriées —, mais dans d'autres contextes trop fortes — et empêcher l'utilisateur de dire quoi que ce soit d'autre que ce qui a été prévu et prescrit dans un schéma actanciel étroit, toute autre interprétation étant rejetée, quelle que soit la manière dont la séquence est tournée. Toutes les erreurs d'analyse sémantique relevées résultent au fond d'un déséquilibre de ce type.

Ce problème de maniement délicat des unités primitives lexicales nous conduit à évoquer le problème plus vaste de la complexité croissante du lexique. Ajouter du vocabulaire à PVI est une tâche plutôt simple tant qu'il s'agit de personnes, d'animaux familiers, de fruits et légumes . . . bref de rajouter des membres à des taxèmes déjà existants. Mais lorsqu'il s'agit de définir de nouveaux prédicats — disons de nouveaux « verbes » pour simplifier —, cette tâche implique de prévoir et de définir les différentes interactions sémantiques possibles de ces verbes avec les autres icônes déjà présentes dans le lexique. Ceci signifie définir des traits de sélection au niveau des verbes que l'on ajoute, mais également retoucher le lexique déjà existant en rajoutant des traits intrinsèques pour toutes les icônes susceptibles d'interagir de manière particulière avec ces verbes. Or ces nouveaux traits peuvent eux-mêmes modifier les interactions de ces icônes avec d'autres verbes déjà existant, etc. En somme, chaque ajout d'un nouveau concept au lexique demanderait dans l'idéal un nouveau cycle complet de contrôle de cohérence et un nouveau re-réglage global.

## 7.6 Conclusion

Les principes d'analyse sémantique définis dans le cadre du projet PVI ont fait la preuve de leur intérêt et de leur validité sur un domaine miniature, ce qui est un résultat relativement positif dans le cadre d'un projet de recherche prospective. Les problèmes identifiés pendant le développement et confirmés par l'évaluation, en revanche, rendent inenvisageable le passage immédiat à une phase de développement plus « industriel » et permettent d'identifier les points sur lesquels devraient porter des efforts de robustification.