

Modélisations linguistiques

Nous souhaitons maintenant mettre en œuvre le formalisme décrit dans le chapitre précédent pour modéliser divers phénomènes linguistiques. Cet exercice constitue essentiellement une mise à l'épreuve des outils proposés plus haut face à une tâche de modélisation réelle dans le domaine de la linguistique formelle. Nous nous appuyons sur des descriptions existantes et matérialisons les concepts sur lesquels elles s'appuient au moyen de formules logiques, que nous utiliserons pour décrire le langage abstrait des énoncés valides. Notre modélisation permettra également de reconstruire la forme phonologique de ces énoncés, ainsi qu'une représentation de leur sens.

Nous produirons ainsi une grammaire munie de contraintes, et deux linéarisations associées, en syntaxe et en sémantique. Cette grammaire couvrira initialement (section 4.2) un modèle simple de la phrase. Nous étendrons ensuite ce modèle en y incluant un traitement de l'accord en genre, nombre, et personne, afin d'illustrer comment l'emploi de propriétés lexicales et de contraintes logiques se substitue à la multiplication de symboles non-terminaux, ou à l'emploi de structures de traits (section 4.3). Nous inclurons ensuite dans notre grammaire des propositions subordonnées, complétives et relatives (section 4.4). L'emploi des pronoms relatifs dans ces dernières entraîne usuellement un traitement des dépendances à longue distance, incluant des contraintes dites d'îlot : nous traiterons ces phénomènes par le biais de contraintes logiques intégrant des expressions régulières. En outre, le phénomène de mouvement associé à l'antéposition du pronom sera traité au moyen de requêtes logiques. Enfin, nous élargirons la grammaire par l'ajout de subordonnées infinitives dépendant de verbes dits à contrôle (section 4.5). Outre leur traitement sur le plan sémantique, nous proposerons pour ces derniers plusieurs linéarisations synchrones en syntaxe, soulignant les différences et les similarités dans la construction de propositions subordonnées complexes entre l'anglais, l'allemand et le néerlandais. Ces linéarisations parallèles illustreront un intérêt de la séparation explicite entre la structure d'un énoncé et sa réalisation, ainsi qu'un exemple d'emploi du lambda-calcul pour produire des réalisations

échappant à une analyse hors-contexte directe (dans le cas de l'ordre des mots en néerlandais).

4.1 Langages abstrait et concrets

Nous présentons maintenant la structure du langage abstrait décrit par notre grammaire, ainsi que celle des deux langages concrets évoqués plus haut (forme phonologique et représentation sémantique). Comme décrit dans les chapitres précédents, ces derniers seront des langages de lambda-termes construits sur une signature appropriée, représentant respectivement des séquences de mots et des formules logiques.

Langage abstrait Comme dans l'exemple du chapitre 3, notre langage abstrait sera un ensemble d'arbres dénotés par des termes, dont les nœuds internes sont construits au moyen d'un symbole unique \bullet muni d'une arité fixée, et dont les feuilles seront des éléments du lexique. Les arêtes seront comme auparavant nommées, chaque étiquette d'arête correspondant à un entier de $[\rho(\bullet)]$. En particulier, la notion de tête syntagmatique sera représentée par une étiquette *tête*; les autres étiquettes d'arêtes utilisées initialement étant *sujet*, *objet* et *déterm*, dénotant respectivement la relation entre un syntagme et son sujet, son objet et son déterminant.

Le langage que décriront ces arbres coïncide essentiellement avec la notion de syntaxe profonde d'une langue décrite par Chomsky [1957] (*deep structure*). Notons que le regroupement des syntagmes dans une structure abstraite ne correspond pas toujours à celui qui apparaît dans sa forme phonologique, bien qu'ils soient en pratique souvent similaires. Une exception notable à cette corrélation s'observe dans l'antéposition des pronoms relatifs et interrogatifs, susceptibles de traverser un grand nombre de subordonnées enchâssées; une autre exception apparaissant dans nos modélisations sera l'ordre des mots dans les dépendances croisées en série, en néerlandais.

Les feuilles de nos structures abstraites seront des entrées lexicales, représentées par un mot associé à un ensemble de propriétés par un lexique, tout comme dans l'exemple du chapitre précédent. La table 4.1 présente quelques entrées possibles pour notre grammaire initiale. Parmi les propriétés qu'elle contient, nous distinguons des parties du discours (*verbe*, *nom*, *nom propre*, *pronom*, *déterminant*), dont l'usage sera similaire aux propriétés *booléen*, *entier* et *opérateur* dans l'exemple du chapitre précédent : ces propriétés seront surtout utilisées comme des symboles non-terminaux dans la grammaire d'approximation, restreignant la catégorie syntaxique des feuilles pouvant apparaître à certains endroits de la structure abstraite. D'autres propriétés (*transitif*, *intransitif*), n'apparaîtront que dans les contraintes de bonne formation, ou dans le cadre de réalisations conditionnelles lors de la linéarisation, guidant l'usage

ou l'interprétation des phrases qui les contiennent de manière plus subtile. Elles incluront principalement des propriétés morphosyntaxiques ou liées à la sous-catégorisation (valence, restrictions de sélection, *etc.*).

un	déterminant
chien	nom
François	nom propre
il	pronom
regarde	verbe, transitif
marche	verbe, intransitif

TABLE 4.1 – Exemples d'entrées lexicales pour le langage abstrait

Tout au long de ce chapitre, le langage abstrait et son lexique seront progressivement étendus afin d'enrichir la grammaire. Nous introduirons en particulier de nouvelles propriétés lexicales et étiquettes d'arête afin de modéliser les phénomènes nouvellement couverts par la grammaire.

Forme de surface La forme phonologique d'un énoncé, désignée dans la suite par le terme de *forme de surface*, est, formellement, un mot sur un alphabet Σ constitué par l'ensemble des mots du lexique. Le langage concret que nous construirons sera formé de lambda-termes, représentant des formes de surface de la manière évoquée dans la section 2.2.4. Ces lambda-termes s'appuieront sur un ensemble de constantes de type $* \rightarrow *$, correspondant aux mots du lexique.

Pour faciliter la lecture de ces termes, nous omettrons systématiquement l'opération de concaténation ainsi que les annotations de typage, pour ne conserver qu'une représentation directe du mot résultant. Ainsi, la concaténation $(. a (. b c)) =_{\beta} \lambda x.a (b (c x))$ de trois termes a , b et c représentant des chaînes sera simplement notée $a b c$, sans ambiguïté supplémentaire en raison de l'associativité de la concaténation.

Sémantique Pour représenter le sens des énoncés, nous nous appuierons tout au long de ce chapitre sur la sémantique par la théorie des modèles initialement proposée par Montague. Dans ce cadre, le sens d'une phrase déclarative est dénoté par une formule logique, et le monde dans lequel cet énoncé se situe est associé à un modèle mathématique. Ainsi, la notion de vérité d'un énoncé coïncide avec la satisfaction de la formule logique représentant son sens, par le modèle correspondant au monde auquel il appartient, aux termes d'une interprétation fixée à l'avance. Nous adopterons plusieurs conventions issues de cette tradition, en particulier en matière de traitement des quantifications par passage de continuation. Nous reviendrons à l'issue de ce chapitre sur le choix de cet outil, notamment dans le but de simplifier nos règles de linéarisation sémantiques.

Concrètement, nos réalisations en sémantique seront des lambda-termes représentant des formules logiques, suivant la méthode présentée dans le dernier paragraphe de la section 2.2.4. Afin de distinguer visuellement ces formules sémantiques de celles employées pour contraindre la grammaire et guider la linéarisation, nous emploierons une police spécifique pour les connecteurs logiques et quantificateurs sémantiques : $\neg, \wedge, \vee, \Rightarrow, \Leftrightarrow, \exists, \forall$. En outre, pour faciliter la lecture des lambda-termes sémantiques, nous omettrons l'abstraction lors d'une quantification, en notant par exemple $\exists x.M$ au lieu de $\exists \lambda x.M$; nous emploierons également les connecteurs comme des opérateurs infixes, en leur attribuant leurs priorités usuelles : ainsi, le terme $M \vee \neg N \wedge O$ abrégera le terme $\vee M (\wedge (\neg N) O)$, par exemple.

Outre les constantes correspondant aux connecteurs logiques et quantificateurs, nos règles de linéarisation sémantiques pourront inclure des constantes représentant les prédicats du lexique (verbes, noms, *etc.*), tandis que les déterminants et pronoms se verront associer des termes d'ordre supérieur assurant la quantification. La nature exacte de ces constantes sera détaillée lors de la construction de la linéarisation sémantique.

Les types atomiques sur lesquels s'appuient nos lambda-termes sémantiques sont e et t , correspondant respectivement aux éléments (les objets du modèle représentant le monde) et aux propositions (valeurs de vérité des énoncés). Par convention, les abstractions portant sur un type atomique (usuellement e) emploieront pour variable une lettre minuscule (x, y, z, \dots) ; tandis que celles portant sur des termes d'ordre supérieur (par exemple de type $e \rightarrow t$) emploieront une majuscule (P, Q, \dots). Ainsi, nous pourrions omettre les annotations de type de la plupart des termes, leur typage étant implicite par convention (par exemple dans $\lambda P.\lambda Q.\lambda x.P x \wedge Q x$).

Pour finir, l'approche que nous décrivons ici associe aux énoncés une sémantique extensionnelle : nous avons opté pour ce modèle pour préserver la simplicité de nos exemples sémantiques. Toutefois, linéariser nos structures abstraites vers une représentation sémantique intensionnelle (permettant d'exprimer les notions de croyance et de mondes possibles) ou tout autre modèle demeure également possible, aussi longtemps que les formules correspondantes peuvent être représentées par le biais du lambda-calcul simplement typé.

4.2 Grammaire initiale

Ayant spécifié les types d'objets que nous manipulerons, nous proposons maintenant une grammaire constituant un modèle simple de la phrase en français. Cette grammaire sera par la suite enrichie progressivement au fil des prochaines sections.

En suivant la méthode exposée tout au long du chapitre 3, nous décrivons d'abord le langage abstrait, modélisant la syntaxe profonde, par le biais d'une

grammaire régulière d'approximation, que nous compléterons ensuite par des contraintes logiques de bonne formation. Nous proposerons enfin des règles de linéarisation en surface et en sémantique pour les productions de cette grammaire.

Grammaire d'approximation Notre grammaire support emploiera deux symboles non-terminaux, P et A , qui correspondent respectivement aux propositions et à leurs arguments; le symbole P tenant lieu d'axiome. Elle se compose de quatre productions, données par la figure 4.1.

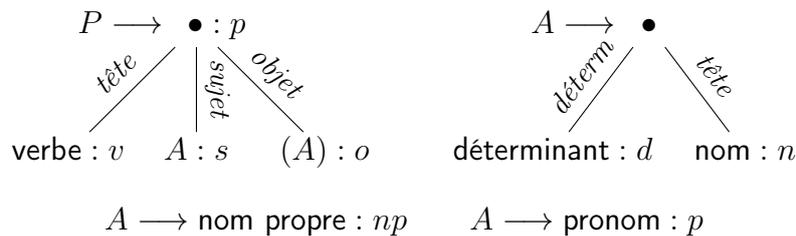


FIGURE 4.1 – Grammaire d'approximation initiale

La première production (en haut à gauche) permet de construire une proposition, dont la tête syntaxique (étiquetée v pour la suite) est une entrée lexicale ayant la propriété **verbe**. Elle est munie d'un argument sujet (étiqueté s), ainsi que d'un argument objet optionnel (o). Les autres productions permettent de construire un argument comme un syntagme nominal, pouvant être composé d'une paire déterminant/nom (le nom étant dans ce cas considéré comme la tête syntaxique), d'un nom propre, ou bien d'un pronom.

Contraintes logiques Comme suggéré dans le chapitre 3, la grammaire d'approximation précédente, complétée par un lexique tel que celui suggéré par la table 4.1, ne décrit pas exactement le langage des structures abstraites que nous souhaitons considérer comme valides. Elle permet en effet de construire des phrases mal formées, ne tenant pas compte, par exemple, de la transitivité d'un verbe. La figure 4.2 illustre ce problème de sur-génération : les deux structures abstraites qu'elle dépeint sont sanctionnées par la grammaire support, mais celle de droite comprend un usage transitif du verbe **marche**, ce qui ne devrait pas être permis par le lexique.

Nous raffinons maintenant le langage abstrait esquissé par la grammaire d'approximation précédente, par le biais de contraintes utilisant le langage logique décrit section 3.3. Nous nous contenterons initialement de garantir le respect de la transitivité des verbes, en associant dans la première production l'existence d'un argument objet optionnel (o) avec la propriété lexicale de transitivité du verbe qui lui correspond (v). Cette contrainte est traduite par la formule suivante : $\text{avec}(o) \Leftrightarrow \text{transitif}(v)$.

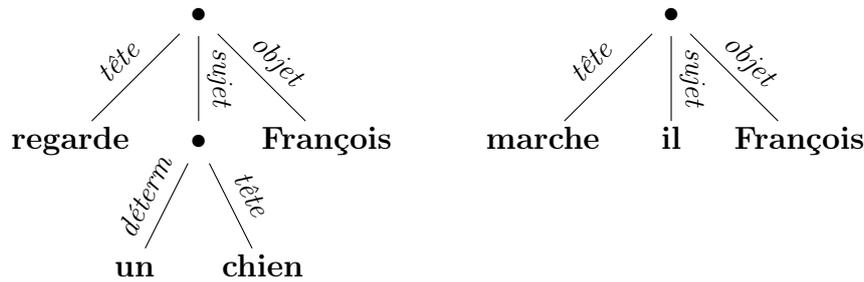


FIGURE 4.2 – Deux structures abstraites non contraintes

La structure de droite de la figure 4.2 est ainsi correctement exclue, tout en préservant de la validité de la structure de gauche. Cette dernière sera utilisée dans la suite pour exemplifier le résultat des linéarisations, en surface et en sémantique.

Linéarisations Nous donnons maintenant, pour chacune des productions de la grammaire, des règles de linéarisation construisant la forme de surface des énoncés, ainsi que leur sémantique.

Forme du surface Pour l’instant, les règles de linéarisation produisant la forme de surface demeurent très simples ; elles ne contiennent pas de réalisations multiples ou de conditions. Nous rappelons les quatre productions de notre grammaire, en précisant à droite de chaque production la règle de linéarisation qui lui est associée :

1. $P \rightarrow \bullet : p$

 $\text{verbe} : v \quad A : s \quad (A) : o$ $s v o$
2. $A \rightarrow \bullet$

 $\text{déterminant} : d \quad \text{nom} : n$ $d n$
3. $A \rightarrow \text{nom propre} : np$ np
4. $A \rightarrow \text{pronom} : p$ p

Cette linéarisation produit simplement le résultat de la concaténation des entrées lexicales de la phrase en suivant l’ordre des mots usuel en français : la première règle construit une proposition suivant l’ordre SVO, la seconde place le déterminant avant le nom, et les deux dernières reprennent simplement la forme de surface de l’entrée lexicale qui réécrit le symbole A .

Ainsi, dans le cas de la structure abstraite valide donnée par la figure 4.2 (à gauche), la réalisation associée est la séquence de mots attendue : « un

chien regarde François ». La seconde production (instanciée pour construire l'argument *sujet*) combine les mots associés aux entrées lexicales de ses variables d et n pour produire la chaîne **un chien**, et la troisième production produit l'argument *objet*, en recopiant la réalisation de son unique variable np (ici, **François**) ; enfin, la première production concatène les réalisations de ses trois arguments s , v et o (dans cet ordre) pour produire la réalisation associée à la phrase.

Remarquons que la variable o dans la première règle correspond à un nœud optionnel : nous considérons ici par simplicité que le lambda-terme associé aux feuilles étiquetées \perp est la constante ε dénotant le mot vide. Une manière plus explicite d'aboutir au même résultat pourrait être d'employer une linéarisation conditionnelle comme suit :

$$1. \quad \begin{array}{c} P \longrightarrow \bullet : p \\ \begin{array}{ccc} \text{tête} & & \text{objet} \\ \swarrow & & \searrow \\ \text{verbe} : v & A : s & (A) : o \end{array} \\ \text{subject} \end{array} \quad \left\{ \begin{array}{l} \text{avec}(o) \longrightarrow s v o \\ \text{sans}(o) \longrightarrow s v \end{array} \right\}$$

Les résultats produits par ces deux variantes de la première règle de linéarisation sont identiques. Par la suite, nous adopterons fréquemment pour des raisons pratiques cette convention d'attribuer une valeur « par défaut » aux arguments optionnels absents, sans spécifier systématiquement des linéarisations alternatives pour chaque cas.

Sémantique Nous décrivons maintenant la linéarisation sémantique de nos structures abstraites, en commençant par détailler les réalisations associées à leurs entrées lexicales.

Chaque entrée du lexique est associée à un lambda-terme sur la signature logique évoquée précédemment (*cf.* page 73). Dans la plupart des cas, ce lambda-terme se réduit à une constante, dont le type dépend de la catégorie syntaxique du mot en question ; d'autres entrées (en particulier des grammèmes comme les déterminants) auront pour réalisations des lambda-termes complexes. La table 4.2 donne quelques exemples de réalisations associées à des entrées lexicales.

François	<i>François</i> ^{<i>e</i>}	regarde	<i>regarde</i> ^{<i>e</i>→<i>e</i>→<i>t</i>}
chien	<i>chien</i> ^{<i>e</i>→<i>t</i>}	un	$\lambda P. \exists x. P x$
marche	<i>marche</i> ^{<i>e</i>→<i>t</i>}		

TABLE 4.2 – Lambda-termes sémantiques associés aux entrées lexicales

Les noms propres sont ainsi simplement associés à des constantes de type e , correspondant à l'entité qu'ils dénotent. Les constantes associées aux noms communs sont des prédicats de type $e \rightarrow t$, le terme « **livre x** » s'interprétant comme vrai lorsque l'élément désigné par x est un livre dans le modèle. Le

Afin de permettre le traitement de la quantification, la règle de linéarisation associée à la première production abstrait l'objet éventuel du verbe, et applique au terme résultant le terme associé à son objet, puis procède de même pour son sujet. Cette règle fait usage du langage de macros détaillé dans la section 3.5.1, pour factoriser le mode d'application du sujet et de l'objet, indépendamment de la transitivité du verbe.

Subsidiairement, remarquons que ce choix de construction n'est pas entièrement innocent. Considérons la règle de linéarisation alternative suivante :

$$1. \quad \begin{array}{c} P \longrightarrow \bullet : p \\ \text{tête} \swarrow \quad \downarrow \text{sujet} \quad \searrow \text{objet} \\ \text{verbe} : v \quad A : s \quad (A) : o \end{array} \quad s \lambda x. \left\{ \begin{array}{l} \text{transitif}(v) \longrightarrow o \lambda y.v x y \\ \text{sinon} \longrightarrow v x \end{array} \right\}$$

Fonctionnellement, cette règle est équivalente à la précédente. Elle serait cependant moins aisée à enrichir si nous souhaitions, par exemple, tenir compte des ambiguïtés de portée dans la quantification, en permettant également de placer la quantification de l'objet en amont de celle du sujet. À titre d'exemple, le choix de quantifier sur le sujet d'abord et sur l'objet ensuite force l'interprétation d'une phrase telle que « Un service secret a conçu chaque méthode de chiffrement connue. » par « Un unique service secret est à l'origine de toutes les méthodes de chiffrement connues. » ; l'autre lecture possible de cette phrase (« Toute méthode de chiffrement connue peut être associée au service secret particulier qui l'a mise au point. »), qui constitue une affirmation plus faible, est exclue par la règle de linéarisation naïve donnée plus haut.

Afin de permettre les deux lectures, nous pourrions proposer une règle de linéarisation factorisant le noyau verbal dans la première version de notre règle par une variable nv , et produisant indifféremment les deux modes de quantification. Cette règle est plus éloignée dans sa structure de la version alternative que nous avons écartée, et se formule comme suit :

$$1. \quad \begin{array}{c} P \longrightarrow \bullet : p \\ \text{tête} \swarrow \quad \downarrow \text{sujet} \quad \searrow \text{objet} \\ \text{verbe} : v \quad A : s \quad (A) : o \end{array} \quad \left\{ \begin{array}{l} s \lambda x.o \lambda y.nv \\ o \lambda y.s \lambda x.nv \end{array} \right\} \quad \text{où } nv = \left\{ \begin{array}{l} \text{transitif}(v) \longrightarrow v x y \\ \text{sinon} \longrightarrow v x \end{array} \right\}$$

Retournant aux règles de linéarisation que nous avons choisi de proposer, la seconde règle construit le sens d'un groupe nominal en appliquant la quantification apportée par son déterminant (d) au terme $\lambda x.n x \wedge P x$, qui vérifie que l'élément sur lequel porte la quantification (identifié avec x par β -réduction) satisfait bien la propriété donnée par son nom commun ($n x$), ainsi que la propriété supplémentaire P – où P est une propriété abstraite, qui sera déterminée par le contexte du groupe nominal ; suivant l'interprétation usuelle.

Les deux dernières règles de linéarisation construisent également des formules de type $(e \rightarrow t) \rightarrow t$, soit en appliquant directement la propriété abstraite P donnée par le contexte à la constante correspondant à l'entité nommée en propre, soit en employant directement la réalisation associée à un pronom. Nous ne détaillons pas ici cette réalisation (complexe), qui doit tenir compte du contexte de l'énoncé et traiter le problème de l'anaphore.

Ainsi, nous pouvons détailler la réalisation sémantique de la structure valide donnée par la figure 4.2 de manière ascendante. Le symbole non-terminal correspondant au groupe nominal sujet se voit associer, selon la seconde règle de linéarisation, le terme $\lambda P. \exists x. \mathbf{chien} \ x \wedge P \ x$ (après β -réduction). Le groupe nominal objet est, pour sa part, réalisé comme $\lambda P. P \ \mathbf{Fran\c{c}ois}$. Pour obtenir la réalisation associée à l'ensemble de la structure, nous remplaçons respectivement dans la règle de linéarisation associée à la première production les variables s , o et v par les réalisations associées aux arguments *sujet* et *objet* et par celle associée à l'entrée lexicale du verbe (**regarde** ^{$e \rightarrow e \rightarrow t$}). Le verbe **regarde** ayant la propriété transitif, le lambda-terme résultant est le suivant :

$$(\lambda P. \exists x. \mathbf{chien} \ x \wedge P \ x) \ \lambda x. (\lambda P. P \ \mathbf{Fran\c{c}ois}) \ \lambda y. \mathbf{regarde} \ x \ y$$

Équivalent, par β -réduction, à : $\exists x. \mathbf{chien} \ x \wedge \mathbf{regarde} \ x \ \mathbf{Fran\c{c}ois}$.

4.3 Traitement de l'accord

Nous souhaitons maintenant enrichir progressivement cette grammaire, en commençant par inclure un traitement de l'accord en genre, nombre et personne. Ceci nous permettra de restreindre davantage le langage résultant, en garantissant l'accord dans une proposition (sujet/verbe) ainsi qu'au sein d'un groupe nominal (déterminant/nom).

Cette extension ne requiert pas de modifier la structure générale des productions de la grammaire : nous ajoutons seulement au lexique un ensemble de propriétés traduisant le genre, le nombre et la personne associés à certaines entrées, et complétons l'ensemble des contraintes logiques de la grammaire pour forcer l'accord entre les entrées d'un même syntagme. Le processus de linéarisation, pour sa part, demeure inchangé.

Alternativement, nous aurions pu traiter l'accord de manière asymétrique, en attribuant des propriétés lexicales d'accord uniquement aux noms et pronoms, et en incluant dans la linéarisation des règles morphologiques permettant de former, par exemple, le pluriel d'un déterminant ou d'un verbe. Les langages concrets issus de ces deux approches seraient identiques, en surface comme en sémantique. Nous avons ici opté pour l'emploi d'une linéarisation directe, sans règles morphologiques.

Propriétés lexicales Nous complétons nos entrées lexicales à l'aide d'un ensemble de propriétés caractérisant le genre (**masculin**, **féminin**), le nombre

(singulier, pluriel) et la personne (1^{re} personne, 2^e personne, 3^e personne). Dans le cas où un mot du lexique appartient à une catégorie syntaxique portant une marque d'accord, mais que sa forme est ambiguë, plusieurs entrées lexicales lui sont associées. Chacune de celles-ci correspond à l'une des interprétations possibles du mot, et elles ne diffèrent que par leurs listes de propriétés. Quelques exemples d'entrées lexicales mises à jour avec ces informations sont données par la table 4.3.

il	pronom, 3 ^e personne, masculin, singulier
elles	pronom, 3 ^e personne, féminin, pluriel
un	déterminant, masculin, singulier
chien	nom, masculin, singulier
souris	nom, féminin, singulier
souris	nom, féminin, pluriel
François	nom propre, masculin, singulier
regarde	verbe, 1 ^{re} personne, singulier
regarde	verbe, 3 ^e personne, singulier

TABLE 4.3 – Extrait de lexique incluant des propriétés d'accord

Remarquons que les entrées lexicales multiples (telles que **souris** ou **regarde**) sont distinctes dans les structures profondes, mais que les linéarisations en surface se restreignent au mot tel qu'il apparaît dans la colonne de gauche. Ainsi, le mot « souris » tel qu'il apparaît dans la réalisation d'une phrase n'est pas marqué en nombre, et correspond à deux structures abstraites distinctes (l'une au singulier et l'autre au pluriel) dont l'une est sans doute mal formée.

L'emploi d'entrées lexicales multiples dans ces cas permet de simplifier les contraintes logiques liées à l'accord, par rapport à l'emploi d'une unique entrée lexicale sous-spécifiée. Supposons en effet que l'entrée lexicale **souris** ait été sous-spécifiée, et soit compatible avec des déterminants ou adjectifs singuliers ou pluriels : dans ce cas, la validité d'une structure abstraite dans une grammaire plus riche (incluant, par exemple, des adjectifs) aurait dépendu des relations entre tous les composants d'un syntagme nominal. Par exemple, la structure abstraite décrite par la séquence de feuilles « **les souris verte** » est invalide ; mais ni le déterminant, ni l'adjectif ne sont incompatibles avec la tête, puisque le nombre de celle-ci est sous-spécifié : il conviendrait alors de comparer les marques d'accord entre tous les éléments du syntagme, requérant des contraintes plus élaborées que celles que nous donnons dans la suite (mais néanmoins formulables dans notre langage logique). La forme des entrées apparaissant dans notre lexique actuel permet d'éviter ces problèmes.

Prédicats et relations additionnels L'un des principaux intérêts de l'emploi d'un langage logique pour la contrainte grammaticale est son extensibilité :

nous pouvons combiner les primitives à notre disposition (opérateurs, quantificateurs, prédicats liés aux propriétés lexicales et relations de domination) pour construire des prédicats et relations plus riches, qui traduisent directement des concepts linguistiques de plus en plus complexes. Comme évoqué dans le chapitre 3, nous employons l'opérateur \triangleq pour définir un nouveau prédicat (à gauche) à partir d'une formule complexe (à droite).

Ainsi nous pouvons, par exemple, construire une relation qui traduit le respect des contraintes de sous-catégorisation imposées par notre grammaire :

$$\begin{aligned} \text{sous-catégorisation}(\text{verbe}, \text{sujet}, \text{objet}) &\triangleq \text{avec}(\text{objet}) \Rightarrow \text{transitif}(\text{verbe}) \\ &\wedge \text{sans}(\text{objet}) \Rightarrow \text{intransitif}(\text{verbe}) \end{aligned}$$

La définition de cette relation peut être envisagée comme la mise en œuvre (implémentation) du concept linguistique de sous-catégorisation d'un verbe dans notre grammaire, sous la forme d'une relation entre trois nœuds *verbe*, *sujet* et *objet*. L'objectif est que, par la suite, toute amélioration du traitement des contraintes de sous-catégorisation dans notre grammaire passe par un changement ponctuel dans cette définition, plutôt qu'une revue exhaustive des contraintes sur les productions liées aux syntagmes verbaux.

Nous introduisons maintenant une série de prédicats liés à l'accord, permettant de parler du genre, du nombre et de la personne d'un syntagme. Un syntagme possède ainsi le genre et le nombre de sa tête. Celle-ci peut-être localisée par l'expression régulière *tête** sur les chemins présents dans la structure abstraite : une entrée lexicale est ainsi sa propre tête, ainsi que celle de tous les syntagmes la dominant par l'intermédiaire d'une séquence d'arêtes étiquetées *tête*. Cette dernière catégorie se limite pour l'instant aux paires déterminant/nom produites par la seconde production, mais comprendra par la suite des syntagmes nominaux plus riches, incluant des modificateurs. Formellement, les définitions ainsi introduites s'appuient sur des relations construites à partir d'expressions régulières (voir définition 3.3), et se formulent comme suit :

$$\begin{aligned} g_masculin(x) &\triangleq \exists t.\text{masculin}(t) \wedge \text{tête}^*(x, t) \\ g_féminin(x) &\triangleq \exists t.\text{féminin}(t) \wedge \text{tête}^*(x, t) \\ n_singulier(x) &\triangleq \exists t.\text{singulier}(t) \wedge \text{tête}^*(x, t) \\ n_pluriel(x) &\triangleq \exists t.\text{pluriel}(t) \wedge \text{tête}^*(x, t) \end{aligned}$$

Par ailleurs, un syntagme est considéré comme étant de la troisième personne si et seulement si il ne se réduit pas à une entrée lexicale à la première ou deuxième personne ; cette dernière condition repose sur l'hypothèse qu'un pronom est le seul élément de son syntagme nominal (il ne peut pas être déterminé ou modifié) ; nous abandonnons pour cette raison l'emploi de l'expression *tête**. Par suite, les syntagmes construits autour d'un nom seront systématiquement traités comme étant de la troisième personne du point de vue de l'accord. Réciproquement, aucune définition logique supplémentaire n'est requise pour

caractériser un syntagme à la première ou deuxième personne : les propriétés lexicales 1^{re} personne et 2^e personne des pronoms suffisent à cette tâche. La formule correspondant à la notion de troisième personne est donc donnée par :

$$3^e \text{ personne}(x) \triangleq \neg(1^e \text{ personne}(x) \vee 2^e \text{ personne}(x))$$

Nous pouvons maintenant définir simplement les relations traduisant l'accord en genre, en nombre et en personne :

$$\begin{aligned} \text{accord_genre}(x, y) &\triangleq g_masculin(x) \wedge g_masculin(y) \\ &\quad \vee g_féminin(x) \wedge g_féminin(y) \\ \text{accord_nombre}(x, y) &\triangleq n_singulier(x) \wedge n_singulier(y) \\ &\quad \vee n_pluriel(x) \wedge n_pluriel(y) \\ \text{accord_personne}(x, y) &\triangleq 1^e \text{ personne}(x) \wedge 1^e \text{ personne}(y) \\ &\quad \vee 2^e \text{ personne}(x) \wedge 2^e \text{ personne}(y) \\ &\quad \vee 3^e \text{ personne}(x) \wedge 3^e \text{ personne}(y) \end{aligned}$$

Mise à jour des contraintes logiques En nous appuyant sur les définitions précédentes, nous pouvons maintenant mettre à jour les contraintes de la grammaire pour tenir compte de l'accord. Pour rappel, la figure 4.1 résume les productions de la grammaire actuelle.

Nous remplaçons la contrainte précédente sur la première production (portant sur la transitivité) par la formule atomique *sous-catégorisation*(v, s, o). La grammaire résultante est exactement équivalente en termes de langage généré. La seule différence réside dans le fait que la contrainte modifiée traduit maintenant explicitement son utilité : elle représente exactement les contraintes de sous-catégorisation mises en œuvre dans la grammaire, ce qui permettra de faciliter de futures extensions.

Par suite, nous ajoutons deux contraintes liées à l'accord. La première s'applique à la première production, et garantit l'accord en nombre et en personne du verbe avec son sujet : $\text{accord_nombre}(s, v) \wedge \text{accord_personne}(s, v)$. La seconde s'applique à la deuxième production, qui construit un argument comme une paire déterminant/nom, et garantit l'accord en genre et en nombre entre ces deux éléments : $\text{accord_genre}(d, n) \wedge \text{accord_nombre}(d, n)$.

Le langage abstrait résultant de l'ajout de ces contraintes est plus restreint que celui de la grammaire initiale, éliminant nombre de structures mal formées. L'exclusion de ces structures du langage abstrait entraîne celle de leurs linéarisations dans le langage concret, améliorant la précision de notre grammaire.

4.4 Traitement des subordinées

Nous souhaitons maintenant inclure dans notre grammaire un traitement des propositions subordinées. Ce traitement inclura des propositions complé-

tives sujet et objet, ainsi que la modification d'un syntagme nominal par une proposition relative, en tenant compte de l'accord entre antécédent et pronom relatif. Un pronom relatif peut aussi bien tenir lieu de sujet que d'objet dans sa proposition¹ ; la linéarisation en surface devra donc permettre l'antéposition du pronom par rapport à la relative. Enfin, un pronom relatif objet peut apparaître dans une proposition subordonnée à la relative à laquelle il est lié ; cette relation de subordination pouvant être indirecte, de sorte que plusieurs complétives successives séparent l'antécédent du pronom relatif. Ce dernier cas de figure n'est possible que dans certaines configurations, dépendant des contraintes dites d'îlot étudiées par Ross [1967] ; et entraîne des dépendances syntaxiques entre des composants arbitrairement lointains, à la fois dans la réalisation de la phrase et dans sa représentation abstraite. Ces dernières contraintes sont qualifiées de dépendances à longue distance, et font l'objet de traitements très divers dans la littérature (*e.g.* Chomsky [1977], Kaplan et Zaenen [1995], ...). Nous présenterons dans cette section une approche rappelant la seconde, proposée dans le cadre des grammaires lexicales-fonctionnelles.

Concrètement, un exemple de phrase que nous souhaitons pouvoir traiter est : « Le livre que je tiens affirme que notre monde est plat. » ; la structure abstraite associée est donnée par la figure 4.3. L'argument objet du verbe « affirme » dans la proposition principale y est réécrit comme une proposition subordonnée complétive ; et son sujet est, comme précédemment, un groupe nominal. Ce dernier est modifié par une proposition subordonnée relative, dans laquelle l'un des arguments est un pronom relatif. Remarquons que l'ajout de modificateurs aux groupes nominaux requiert l'introduction d'une nouvelle étiquette d'arête dans notre langage abstrait (*mod*).

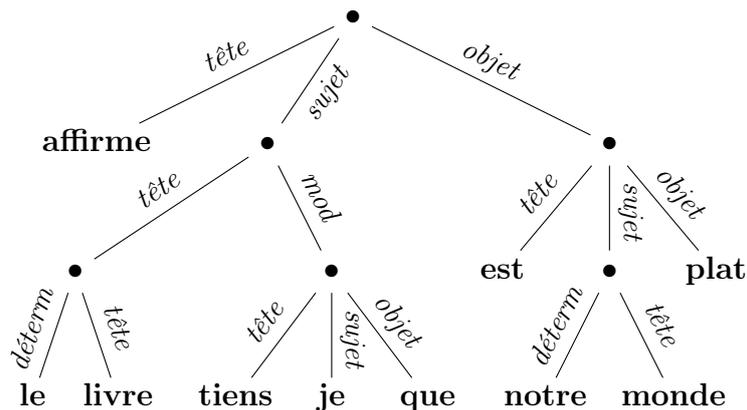


FIGURE 4.3 – Exemple de structure abstraite contenant des subordinées

1. Par exemple, le pronom relatif **qui/que** est sujet du verbe **lire** dans « le livre qui est lu », alors qu'il est son objet dans « le livre que Paul lit ».

Structure grammaticale Afin de permettre la construction de structures abstraites semblables à celle proposée figure 4.3, nous ajoutons maintenant deux autres productions aux quatre de la grammaire d’approximation initiale. Les règles ainsi introduites sont données par la figure 4.4; nous nous référons à elles dans la suite comme à la cinquième et sixième production de la grammaire support.



FIGURE 4.4 – Productions supplémentaires pour l’ajout de subordinées

La cinquième production, à gauche, permet de réécrire un argument verbal (A) comme une proposition (P); elle permet l’inclusion de propositions complétives (sujet et objet) dans la grammaire (« que notre monde est plat »). Observons que cette production rend notre grammaire récursive, en lui permettant d’imbriquer plusieurs complétives (subordonnées les unes aux autres). La sixième production, à droite, permet d’ajouter un modificateur à un argument, sous la forme d’une proposition; cette règle traduit l’ajout de propositions relatives (« que je tiens ») qualifiant le reste du groupe nominal (« le livre »). Dans le syntagme nominal résultant, le nœud dominé par l’arête *tête* correspond à l’antécédent, et celui dominé par *mod* à la relative.

Une fois encore, l’ensemble des structures abstraites générées par cette grammaire contient de nombreux éléments invalides requérant d’être filtrés. Nous décrirons dans la suite les contraintes de sous-catégorisation associées aux complétives et celles portant sur le placement des pronoms relatifs. Celles-ci devront cependant s’appuyer sur des prédicats et relations nouvellement définis, ainsi que sur des propriétés lexicales additionnelles que nous introduisons maintenant.

Propriétés lexicales Les propriétés supplémentaires venant enrichir notre lexique sont *relatif*, *nominatif*, *accusatif*, *complétive objet*, *objet nominal*, *complétive sujet* et *sujet nominal*. Les premières d’entre elles concernent les pronoms : la propriété *relatif* dénote un pronom relatif, et les propriétés *nominatif* et *accusatif* dénotent les fonctions syntaxiques qu’il peut occuper (respectivement sujet et objet, héritage des cas morphologiques). Les autres propriétés apportent des informations sur la sous-catégorisation des verbes : *complétive objet*, *objet nominal*, *complétive sujet* et *sujet nominal* dénotent respectivement qu’un verbe peut avoir pour objet une proposition complétive ou un syntagme nominal, et de même pour son sujet. Des exemples d’entrées lexicales illustrant l’usage de ces propriétés sont fournies par les Tables 4.4 et 4.5.

que	pronom, relatif, accusatif, masculin, singulier
que	pronom, relatif, accusatif, féminin, singulier
que	pronom, relatif, accusatif, masculin, pluriel
que	pronom, relatif, accusatif, féminin, pluriel
qui	pronom, relatif, nominatif, masculin, singulier
qui	...
lequel	pronom, relatif, nominatif, masculin, singulier
lequel	pronom, relatif, accusatif, masculin, singulier
laquelle	pronom, relatif, nominatif, féminin, singulier
laquelle	...
lesquels	...
lesquelles	...

TABLE 4.4 – Entrées lexicales décrivant des pronoms relatifs

Remarquons que certains pronoms relatifs (**qui**, **que**) ne portent pas de marque d'accord en genre et en nombre, tandis que d'autres (**le/la/lesquelles**) sont partiellement indifférents à la fonction syntaxique dans laquelle ils sont utilisées (sujet ou objet direct). Le lexique sur lequel nous nous appuyons développe toute la combinatoire associée à ces pronoms relatifs. Le développement d'une grammaire à large couverture requerrait l'emploi d'une représentation compacte pour ces multiples entrées. Le cas échéant, il serait également possible d'intégrer des règles morphologiques à la linéarisation en surface des phrases, comme évoqué précédemment. Ceci permettrait de construire la forme appropriée pour chaque pronom relatif dynamiquement au moment de la réalisation, par opposition au lexique statique, pré-calculé, dont nous disposons actuellement.

affirme	verbe, 3 ^e personne, singulier, transitif, sujet nominal, complétive objet, objet nominal
regarde₁	verbe, 3 ^e personne, singulier, transitif, sujet nominal, objet nominal
regarde₂	verbe, 3 ^e personne, singulier, transitif, complétive sujet, sujet nominal, objet nominal

TABLE 4.5 – Entrées lexicales décrivant des verbes acceptant des complétives

Nous distinguons en outre les différents usages d'un verbe, afin de permettre un traitement valide en sémantique. Le verbe « regarder » possède ainsi deux sens : le premier comme synonyme de voir, requérant un sujet nominal ; et le second correspondant à l'emploi familier comme synonyme de concerner, qui admet plusieurs usages (« La date de mon départ me regarde. » ou « Que je décide de partir ou non me regarde. »).

Maintenant que nous avons enrichi notre collection de propriétés lexicales,

nous devons également étendre notre vocabulaire logique pour pouvoir formuler les contraintes de grammaticalité portant sur les relatives. Dans ce but, nous explicitons tout d'abord la nature de ces contraintes linguistiques, avant de décrire la mise en œuvre des notions associées dans notre vocabulaire logique.

Fonctions syntaxiques des pronoms relatifs La formation d'une relative obéit à un ensemble de contraintes précis : le pronom relatif antéposé peut avoir fonction de sujet ou d'objet dans la proposition, mais il peut également correspondre à l'argument d'une proposition complétive qui lui est subordonnée. Cette possibilité de relier une relative à son pronom par le biais d'un chemin complexe (dit *chemin wh*, en raison de la forme des pronoms relatifs et interrogatifs anglais) fait toutefois l'objet de restrictions importantes, connues sous le nom de *contraintes d'îlot* (*island constraints*). Considérons ainsi les exemples suivants :

- 1 Le camion qui accélère ...
- 2 Les livres que tu collectionnes ...
- 3 Les gens que Luc affirme que je fréquente ...
- 3a Les gens que Marie pense que Luc affirme que je fréquente ...
- 3b Les gens que tu dis que Marie pense que Luc affirme que je fréquente
- 4 Luc affirme que le camion accélère.
- 4a *Le camion qui Luc affirme qu'accélère ...
- 5 Que des enfants lisent ce livre inquiète leurs parents.
- 5a *Ce livre, lequel que des enfants lisent inquiète leurs parents ...
- 6 L'homme qui a vu l'homme qui a vu l'ours ...
- 6a L'ours que l'homme qui a vu l'homme qui a vu ...

Les exemples 1 et 2 sont des groupes nominaux modifiés par une relative, dans laquelle le pronom relatif a fonction de sujet et d'objet respectivement. L'exemple 3 illustre la possibilité pour un pronom relatif d'occuper une fonction d'objet dans une complétive subordonnée à la proposition relative à laquelle il correspond ; les exemples 3a et 3b montrant que ce pouvoir n'est pas limité par la « profondeur » de la complétive : le pronom relatif peut être l'objet d'une complétive subordonnée à une complétive subordonnée à la relative, et ainsi de suite.

En revanche, les exemples 4 à 6 soulignent trois limitations imposées par les contraintes d'îlot : un pronom relatif ne peut correspondre au sujet d'une complétive objet (4a), ni à l'argument d'une complétive sujet (5a) ; pas plus qu'il ne peut provenir d'une autre relative enchâssée. L'ensemble des contraintes d'îlot conduisant au rejet de ces énoncés a été, comme mentionné précédemment, implémenté avec concision dans la tradition des grammaires lexicales-fonctionnelles. L'analyse retenue est basée sur la notion d'*incertitude fonctionnelle* (*functional uncertainty*), qui permet au pronom d'être relié à son

antécédent par un chemin régulier dans une structure arborescente (la structure fonctionnelle, ou f-structure) qui représente l'énoncé à un niveau abstrait, et présente de nombreuses similarités avec notre notion de structure abstraite. Aussi, ces contraintes de régularité sur le chemin traversé peuvent se traduire directement dans notre formalisme : par une expression régulière sur les étiquettes d'arêtes, reliant le pronom relatif à la proposition qui lui correspond. L'expression *sujet | objet** désigne ainsi soit le sujet d'une relative ; soit son objet, ou l'objet de sa subordinée complétive objet, ou ainsi de suite. Cette expression recouvre les rôles syntaxiques autorisés pour un pronom relatif, incluant les relatives des exemples 1 à 3 tout en excluant celles des exemples 4 et 5.

Observons enfin que les structures que nous cherchons à exclure de la grammaire ne sont pas exactement représentées par les phrases d'exemple ci-dessus : ces dernières sont au plus une réalisation plausible des structures abstraites que nous souhaitons rejeter. Il est cependant plus simple de raisonner sur ces exemples que sur les structures abstraites qui les sous-tendent. En d'autres termes, nos contraintes filtrent seulement des structures abstraites jugées incorrectes : l'élimination des énoncés agrammaticaux (comme 4a et 5a) est une conséquence du rejet de certaines structures, dont les réalisations en surface auraient été ces énoncés jugés incorrects.

Vocabulaire logique Ayant établi la nature des contraintes que nous souhaitons imposer à nos structures, nous pouvons inclure les notions intermédiaires utiles à leur formulation dans notre vocabulaire logique.

Tout d'abord, nous souhaitons pouvoir identifier par un prédicat une proposition et un syntagme nominal dans une structure abstraite. Selon notre grammaire support, une proposition est une structure dont la tête syntaxique (à distance 1) est un verbe. Un syntagme nominal a pour sa part un nom commun, un nom propre ou un pronom comme tête syntaxique, située à une distance arbitraire dans la structure (puisque'un syntagme peut se réduire à un pronom, où être composé d'un nom commun accompagné d'un déterminant et de modificateurs). Nous pouvons utiliser à cet effet les relations basées sur les expressions régulières, l'expression *tête** caractérisant la notion de tête syntaxique. La définition des prédicats traduisant ces deux notions dans notre grammaire est alors transparente :

$$\begin{aligned} \text{proposition}(x) &\triangleq \exists v. \text{verbe}(v) \wedge \text{tête}(x, v) \\ \text{synt_nominal}(x) &\triangleq \exists n. (\text{nom}(n) \vee \text{nom propre}(n) \vee \text{pronom}(n)) \wedge \text{tête}^*(x, n) \end{aligned}$$

Nous formulons également une nouvelle relation, reliant une proposition relative à toute position que son pronom associé est susceptible d'occuper, en respectant les contraintes d'ilot présentées plus haut :

$$\text{chemin_wh}(x, y) \triangleq \text{sujet}(x, y) \vee \text{objet}^+(x, y)$$

Afin de réaliser l'antéposition du pronom relatif lors de la linéarisation en surface, et d'assurer la concordance entre une proposition relative et son pronom associé, nous définissons deux prédicats liés à la notion d'extraction. D'une part, un élément est dit extrait si et seulement si il s'agit d'un pronom relatif (ce qui est vrai dans le cadre de notre grammaire actuelle – ce prédicat pourrait être enrichi par la suite pour inclure, par exemple, les pronoms interrogatifs). D'autre part, nous caractérisons l'ensemble des propositions contenant une extraction, comme l'ensemble des propositions contenant un élément extrait au terme d'un chemin d'extraction valide.

$$\begin{aligned} \text{extrait}(x) &\triangleq \text{pronom}(x) \wedge \text{relatif}(x) \\ \text{prop_ext}(x) &\triangleq \text{proposition}(x) \wedge \exists p. \text{extrait}(p) \wedge \text{chemin_wh}(x, p) \end{aligned}$$

Remarquons que, dans la définition du prédicat *prop_ext*, la première clause (*proposition(x)*) est en fait redondante (compte tenu de notre grammaire support, le nœud *x* correspond nécessairement à une proposition en raison de l'existence d'un chemin *wh* de *x* à *p*). Elle traduit néanmoins explicitement le fait qu'une « proposition contenant une extraction » est, implicitement, une proposition : une éventuelle redéfinition de la nature d'un chemin *wh* (par exemple, à la lumière de nouvelles observations linguistiques) est ainsi moins susceptible de mettre en péril la cohérence de cette définition.

Nous pouvons maintenant nous appuyer sur ces notions pour étendre notre modélisation logique. Nous définissons ainsi une relative comme une proposition contenant un pronom relatif extrait et modifiant un syntagme nominal :

$$\text{relative}(x) \triangleq \text{prop_ext}(x) \wedge \exists s. \text{synt_nominal}(s) \wedge \text{mod}(s, x)$$

Pour conclure, nous pouvons désormais définir dans notre langage logique la relation d'antécédence : l'antécédent *x* d'un pronom relatif *y* est un groupe nominal ; et tous deux sont reliés par l'intermédiaire d'une proposition relative qui modifie l'antécédent, et qui contient le pronom relatif (au terme d'un chemin *wh* valide). La définition est la suivante (pour rappel, l'opérateur \uparrow est issu de la définition 3.3) :

$$\begin{aligned} \text{antécédent}(x, y) &\triangleq \text{synt_nominal}(x) \wedge \text{pronom}(y) \wedge \text{relatif}(y) \\ &\wedge \exists r. \text{relative}(r) \wedge \text{tête}^* \uparrow \text{mod}(x, r) \wedge \text{chemin_wh}(r, y) \end{aligned}$$

La dernière définition complétant notre vocabulaire logique vise à remettre à jour notre définition précédente de la notion de sous-catégorisation, pour tenir compte des propriétés lexicales des verbes acceptant des sujets ou objets

nominaux ou sous la forme de propositions complétives :

$$\begin{aligned}
\text{sous-catégorisation}(v, s, o) &\triangleq \text{avec}(o) \Rightarrow \text{transitif}(v) \\
&\wedge \text{sans}(o) \Rightarrow \text{intransitif}(v) \\
&\wedge \text{proposition}(o) \Rightarrow \text{complétive objet}(v) \\
&\wedge \text{synt_nominal}(o) \Rightarrow \text{objet nominal}(v) \\
&\wedge \text{proposition}(s) \Rightarrow \text{complétive sujet}(v) \\
&\wedge \text{synt_nominal}(s) \Rightarrow \text{sujet nominal}(v)
\end{aligned}$$

Cette définition vient remplacer celle proposée précédemment pour la relation *sous-catégorisation*.

Contraintes grammaticales Nous disposons maintenant de tous les outils nécessaires pour actualiser et compléter les contraintes logiques portant sur la grammaire support.

La première production conserve sa contrainte *sous-catégorisation*(v, s, o), mais profite de la version actualisée de cette relation, restreignant l’emploi de complétives ou de syntagmes nominaux comme sujet et objet en fonction des propriétés lexicales du verbe.

La quatrième production ($A \rightarrow \text{pronom} : p$) requiert maintenant un traitement des cas (pronoms sujets et objets), fourni par la contrainte suivante (la variable liée p' désignant la proposition dont le pronom p est un argument) :

$$\begin{aligned}
&\text{nominatif}(p) \Rightarrow \exists p'. \text{sujet}(p', p) \\
\wedge &\text{accusatif}(p) \Rightarrow \exists p'. \text{objet}(p', p)
\end{aligned}$$

Cette double condition évite de sanctionner des énoncés employant des pronoms relatifs dans des rôles syntaxiques incorrects, comme dans « *le mécanisme qui j’ai réparé » (pronom sujet employé comme objet) ou « *la machine que tombe en panne régulièrement » (le cas inverse).

De plus, une structure abstraite valide doit requérir une correspondance unique entre propositions et pronoms relatifs : chaque proposition doit être associée à un unique pronom, et réciproquement. Nous contraignons ainsi un pronom relatif à être lié à une unique relative², par le biais d’un chemin *wh* valide :

$$\text{relatif}(p) \Rightarrow \exists! r. \text{relative}(r) \wedge \text{chemin_wh}(r, p)$$

Cette contrainte exclut l’emploi d’un pronom relatif hors du contexte d’une relative, éliminant des phrases telles que « *Que j’ai rencontré. » en forçant l’existence d’une relative r . L’unicité de r est techniquement redondante, au vu des définitions de *relative* et *chemin_wh* : tout nœud constituant un candidat acceptable pour r doit être dominé par une arête *mod*, excluant la possibilité

2. La condition d’unicité dans une quantification existentielle, dénotée par $\exists!$, est définissable en logique du premier ordre comme suit : $\exists! x. P(x) \stackrel{\text{def}}{\Leftrightarrow} \exists x. P(x) \wedge \forall x'. P(x') \Rightarrow x' = x$.

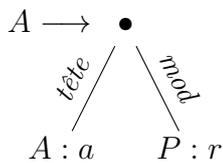
d'être dominé par *objet* et donc celle qu'il existe un autre candidat valable pour r . Cette condition d'unicité fait néanmoins sens dans le cadre de cette contrainte : chaque pronom doit pouvoir être associé à une unique proposition relative.

Enfin, nous imposons l'accord en genre et en nombre d'un pronom avec son antécédent. Remarquons que la définition actuelle d'un antécédent limite pour l'instant l'application de cette règle d'accord aux pronoms relatifs, mais que cette contrainte sur les pronoms resterait valide dans la perspective d'une définition plus large de la notion d'antécédent.

$$\forall a. \text{antécédent}(a, p) \Rightarrow \text{accord_genre}(a, p) \wedge \text{accord_nombre}(a, p)$$

Cette règle d'accord élimine des énoncés où le pronom relatif et son antécédent sont incompatibles, tels que « *cette voiture, lesquels revient de chez le garagiste, ... ».

Pour finir, la sixième production permet la modification d'un argument par une relative, et porte deux contraintes : d'une part, une telle modification est généralement interdite sur les pronoms ; d'autre part, chaque relative doit contenir un unique pronom relatif au terme d'un chemin *wh* valide. Ces deux contraintes sont formulées comme suit (nous rappelons en passant la production correspondante) :



1. $\neg(\text{pronom}(a) \wedge \text{relatif}(p))$
2. $\exists! p. \text{pronom}(p) \wedge \text{relatif}(p) \wedge \text{chemin_wh}(r, p)$

Ces dernières contraintes excluent d'une part la modification d'un pronom relatif (telle que « *l'employé qui que la société va licencier est parti ») ; et d'autre part l'absence de pronom relatif dans une relative (« *les chaises elles sont sur les bureaux ») ou sa multiplicité (« *l'homme qui que regarde dans le miroir »).

Linéarisations Nous décrivons maintenant les deux linéarisations associées à notre grammaire actualisée, permettant l'antéposition des pronoms relatifs et l'ajout de conjonctions de subordination en surface, et incluant un traitement des événements en sémantique.

Forme de surface La principale difficulté dans la mise au point de règles de linéarisation pour cette version de la grammaire est le traitement de l'antéposition des pronoms relatifs : leur position dans chaîne constituant la réalisation de surface est en effet arbitrairement distante de leur position dans la structure abstraite. Toutefois, ces pronoms peuvent être commodément insérés lors de la linéarisation : au niveau de la racine de la relative à laquelle ils se rattachent, la réalisation du pronom peut être insérée entre celle de sa

relative (à droite) et celle du syntagme nominal auquel sa relative est rattachée (à gauche). Nous utilisons à cet effet, le mécanisme de requêtes logiques introduit dans la section 3.4.3 : la variable \mathbf{p} présente dans la sixième règle de linéarisation ci-dessous dénote ainsi la réalisation attachée à toute position p satisfaisant la pré-condition $extrait(p) \wedge chemin_wh(r, p)$. Le pronom extrait lui-même (situé à la position p) n'est pas réalisé localement en raison de la règle de linéarisation associée à la production qui le domine (la première). Celle-ci associe la chaîne vide ε à un argument formé par un pronom extrait, et préserve la réalisation de l'argument dans le cas contraire. Deux variables (su et ob) sont introduites à cet effet : elles peuvent être envisagées comme une abstraction de la notion de réalisation de l'argument (sujet ou un objet). Observons que ces règles de linéarisation entraînent que la réalisation d'une phrase incorpore exactement une fois chaque pronom, en raison des contraintes de grammaticalité : chaque relative (associée à une requête logique) correspond à exactement un pronom extrait (réalisé localement par ε), et réciproquement.

Les règles de linéarisation ajoutées ou modifiées sont les suivantes :

1.
$$\begin{array}{c}
 P \longrightarrow \bullet : p \\
 \begin{array}{l}
 \text{tête} \swarrow \quad \downarrow \text{sujet} \quad \searrow \text{objet} \\
 \text{verbe} : v \quad A : s \quad (A) : o
 \end{array}
 \end{array}$$
 où $su = \left\{ \begin{array}{l} extrait(s) \longrightarrow \varepsilon \\ \text{sinon} \longrightarrow s \end{array} \right\}$
 et $ob = \left\{ \begin{array}{l} extrait(o) \longrightarrow \varepsilon \\ \text{sinon} \longrightarrow o \end{array} \right\}$
5. $A \longrightarrow P : c$ **que** c
6.
$$\begin{array}{c}
 A \longrightarrow \bullet \\
 \begin{array}{l}
 \text{tête} \swarrow \quad \searrow \text{mod} \\
 A : a \quad P : r
 \end{array}
 \end{array}$$
 $extrait(p) \wedge chemin_wh(r, p) \longrightarrow a \mathbf{p} r$

La cinquième règle de linéarisation ajoute explicitement la conjonction de subordination « que » pour introduire une complétive ; cette règle simple suffit à permettre l'analyse de complétives introduites par « que ». Un traitement plus en profondeur des complétives pourrait inclure une lexicalisation des conjonctions de subordination (permettant un traitement approprié en sémantique), et une complétion de la structure abstraite par un nœud supplémentaire, dominé par une arête portant une étiquette *conjonction*.

Linéarisation alternative Les requêtes logiques nous fournissent un moyen de décrire simplement le mouvement du pronom relatif. Elles ne sont cependant pas requises pour rendre compte de ce type de phénomènes dans notre formalisme : l'emploi de lambda-termes plus complexes et de types variés (par contraste avec les réalisations actuelles, qui sont uniformément des

chaînes de type $* \rightarrow *$) permet de construire séparément la réalisation locale et l'élément extrait d'une proposition, et de reconstituer l'énoncé par la suite. Spécifiquement, nous pouvons ici construire la réalisation d'une proposition contenant un pronom extrait comme une paire de chaînes, contenant à gauche la réalisation locale de la proposition (où le pronom est absent), et à droite le pronom extrait. La règle de linéarisation combinant la relative et son syntagme nominal peut alors concaténer ces chaînes dans l'ordre approprié pour obtenir l'antéposition du pronom relatif.

En suivant les conventions de notation utilisées jusque là, nous notons (s_1, s_2) dans la règle de linéarisation ci-dessous la paire formée par les chaînes s_1 et s_2 ; et dénotons par $\pi_1(p)$ et $\pi_2(p)$ les projections gauche et droite (respectivement) de la paire p . Ces opérations de construction et projection pouvant être représentées par le lambda-calcul simplement typé (comme illustré dans la section 2.2.4), ces notations constituent simplement une abréviation commode pour les lambda-termes correspondants, et ne requièrent pas d'augmenter le pouvoir d'expression des règles de linéarisation données jusque là.

Les règles de linéarisation suivantes produisent alors les mêmes réalisations que les précédentes, sans employer de requêtes logiques :

$$\begin{array}{l}
 1. \quad \begin{array}{c} P \longrightarrow \bullet : p \\ \text{tête} \swarrow \quad \downarrow \quad \searrow \text{objet} \\ \text{verbe} : v \quad A : s \quad (A) : o \end{array} \\
 5. \quad A \longrightarrow P : c \\
 6. \quad \begin{array}{c} A \longrightarrow \bullet \\ \text{tête} \swarrow \quad \searrow \text{mod} \\ A : a \quad P : r \end{array}
 \end{array}
 \quad \begin{array}{l}
 \left\{ \begin{array}{l} \text{prop_ext}(p) \longrightarrow (su \ v \ ob, pr) \\ \text{sinon} \longrightarrow s \ v \ o \end{array} \right\} \\
 \text{où } su = \left\{ \begin{array}{l} \text{extrait}(s) \longrightarrow \varepsilon \\ \text{sinon} \longrightarrow s \end{array} \right\} \\
 \text{et } ob = \left\{ \begin{array}{l} \text{prop_ext}(o) \longrightarrow \pi_1(o) \\ \text{extrait}(o) \longrightarrow \varepsilon \\ \text{sinon} \longrightarrow o \end{array} \right\} \\
 \text{et } pr = \left\{ \begin{array}{l} \text{prop_ext}(o) \longrightarrow \pi_2(o) \\ \text{extrait}(o) \longrightarrow o \\ \text{extrait}(s) \longrightarrow s \end{array} \right\} \\
 \left\{ \begin{array}{l} \text{prop_ext}(c) \longrightarrow (\mathbf{que} \ \pi_1(c), \pi_2(c)) \\ \text{sinon} \longrightarrow \mathbf{que} \ c \end{array} \right\} \\
 a \ \pi_2(r) \ \pi_1(r)
 \end{array}$$

Le fonctionnement de ces règles nous paraît cependant moins intuitif que celui des précédentes, dans lesquelles le mécanisme de requête logique permet d'éviter les problèmes de typage, dus à la possibilité pour la réalisation d'un nœud d'être, au choix, une chaîne (de type $* \rightarrow *$) ou une paire de chaînes (ayant le type $(\sigma \rightarrow \sigma \rightarrow \sigma) \rightarrow \sigma$, avec $\sigma = * \rightarrow *$). Une solution alternative résolvant cette ambiguïté pourrait être d'associer à tous les nœuds de la structure abstraite une paire de chaînes, incluant à toute réalisation un

composant extrait (éventuellement vide). Ce choix de modélisation, évoquant l'hypothèse qu'une phrase simple (sans mouvement) contient un composant déplacé « vide », nous paraît cependant difficile à justifier linguistiquement.

Sémantique Notre linéarisation en sémantique pour le traitement des subordinées complétives passe par l'emploi d'une sémantique événementielle, à la manière de Davidson [1967]. L'argument d'un verbe sous-catégorisant pour une complétive est un évènement, dénoté par une variable e de type e , ce même évènement étant également un argument supplémentaire du verbe de la complétive. Par exemple, la sémantique associée à une phrase telle que « Luc croit que le chat reviendra. » est qu'il existe un évènement e que constitue le retour du chat, et que c'est en cet évènement que Luc croit. Accessoirement, l'analyse de cette phrase affirme qu'il existe également un évènement de croyance (dont Luc est l'acteur), dont la continuation n'est pas utilisée. La formule logique associée est $\exists x. \mathbf{chat}(x) \wedge \exists e_2. \exists e_1. \mathbf{croit}(e_1, \mathbf{Luc}, e_2) \wedge \mathbf{reviendra}(e_2, x)$, où le premier argument d'un prédicat verbal est l'évènement décrit par ce verbe, et les suivants correspondent respectivement à son sujet et à son objet. Ce traitement des complétives n'est pas entièrement satisfaisant par certains aspects (en particulier, il ne permet pas en l'état un traitement approprié des problèmes liés à l'intension), mais constitue néanmoins une base suffisante pour démontrer le fonctionnement de notre formalisme, et permettre dans la suite un traitement adéquat des verbes à contrôle.

Les nouvelles règles de linéarisation sémantique associées aux productions de la grammaire sont les suivantes :

- $$\left\{ \begin{array}{l} prop_ext(p) \longrightarrow ph \\ sinon \longrightarrow ph \Omega^e \end{array} \right\}$$
- où $ph = \lambda r. \lambda C. su \lambda x. ob \lambda y. \exists e. (C e) \wedge nv$
1.
$$\begin{array}{c} P \longrightarrow \bullet : p \\ \begin{array}{ccc} \text{tête} & & \text{objet} \\ \diagdown & & \diagup \\ \text{verbe} : v & A : s & (A) : o \end{array} \\ \text{subject} \end{array}$$

$$\begin{array}{l} où su = \left\{ \begin{array}{l} extrait(s) \longrightarrow s r \\ sinon \longrightarrow s \end{array} \right\} \\ et ob = \left\{ \begin{array}{l} prop_ext(o) \longrightarrow o r \\ extrait(o) \longrightarrow o r \\ sinon \longrightarrow o \end{array} \right\} \\ et nv = \left\{ \begin{array}{l} transitif(v) \longrightarrow v e x y \\ sinon \longrightarrow v e x \end{array} \right\} \end{array}$$
 2.
$$\begin{array}{c} A \longrightarrow \bullet \\ \begin{array}{cc} \text{déterm} & \text{tête} \\ \diagdown & \diagup \\ \text{déterminant} : d & \text{nom} : n \end{array} \end{array}$$

$$\lambda P. d \lambda x. n x \wedge P x$$
 3.
$$A \longrightarrow \text{nom propre} : np \qquad \lambda P. P np$$

4. $A \longrightarrow \text{pronom} : p$ $\left\{ \begin{array}{l} \text{extrait}(p) \longrightarrow \lambda r. \lambda P. P r \\ \text{sinon} \longrightarrow p \end{array} \right\}$
5. $A \longrightarrow P : c$ $\left\{ \begin{array}{l} \text{prop_ext}(c) \longrightarrow c \\ \text{sinon} \longrightarrow c \end{array} \right\}$
6. $A \longrightarrow \bullet$
 $\begin{array}{c} \text{tête} \\ \swarrow \\ A : a \end{array}$ $\begin{array}{c} \text{mod} \\ \searrow \\ P : r \end{array}$ $\lambda P. a \lambda x. (r x \lambda e. \top) \wedge P x$

Observons tout d'abord la quatrième règle de linéarisation, associée aux pronoms. Celle-ci propose une linéarisation alternative en cas d'extraction (s'il s'agit d'un pronom relatif) : dans ce dernier cas, le type sémantique de la réalisation du pronom est $e \rightarrow (e \rightarrow t) \rightarrow t$, plutôt que $(e \rightarrow t) \rightarrow t$. Cet élément supplémentaire est celui auquel la propriété P est appliquée, omettant la quantification. En effet, une proposition relative n'est autre qu'un modificateur (de type $e \rightarrow t$), s'appliquant à l'élément quantifié par le groupe nominal auquel elle se rattache. Cet élément, qui est manquant dans le contexte de la relative, correspond à la variable r de type e qui est abstraite ici.

Ceci observé, la règle de linéarisation associée à la première production est la plus complexe. Afin d'en simplifier la compréhension, nous avons fait usage du langage de macros (décrit section 3.5.1) pour en isoler les constituants variables (au travers des variables su , ob et nv), et factoriser la majeure partie du terme entre deux réalisations alternatives (grâce à la variable ph). Les composants sujet et objet de la phrase, de type $(e \rightarrow t) \rightarrow t$, sont abstraits au moyen des variables su et ob respectivement : dans le cas général, ces variables dénotent exactement la réalisation associée au sujet ou à l'objet dans la structure abstraite. Toutefois, dans le cas où l'un des deux est (ou contient) un pronom relatif extrait, l'élément manquant à celui-ci est fourni sous la forme d'une variable libre r , qui dénote l'élément auquel se réfère le pronom relatif. Observons que les contraintes d'îlot formulées plus haut interdisent à une complétive sujet de contenir une extraction dans une phrase valide : la réalisation alternative $\text{prop_ext}(s) \longrightarrow s r$ de su est donc inutile, et a été exclue de cette règle de linéarisation. Enfin, le noyau verbal, représenté par la variable nv , est formé par l'application du prédicat verbal aux variables libres dénotant ses arguments : la variable e représente l'évènement décrit par le verbe, x est son sujet et, si la sous-catégorisation du verbe le réclame, y est son objet.

Ces composants sont ensuite combinés pour former la variable ph dénotant le sens de la proposition : une proposition affirme l'existence de l'évènement e apparaissant dans nv , et permet une continuation sur celui-ci au moyen de la variable C . Le sujet et l'objet sont ensuite ajoutés au moyen de la construction usuelle par montée de type, et la variable C est ensuite abstraite. Pour terminer la variable r susceptible d'apparaître dans su ou ob (dans le contexte d'une relative) est à son tour abstraite.

Une fois la variable ph formée, deux cas sont envisageables : soit la proposition contient une extraction, auquel cas son type doit être $e \rightarrow (e \rightarrow t) \rightarrow t$ (le premier argument correspond à l'élément dénoté par le pronom relatif, et le second à la continuation de l'évènement), soit ce n'est pas le cas, auquel cas son seul argument possible est la continuation de l'évènement : nous fournissons alors un élément sémantique vide (Ω^e) en guise de variable r .

Observons finalement que dans ce dernier cas, la variable libre r n'apparaît ni dans su , ni dans ob ; la constante Ω n'apparaît donc jamais dans la réalisation d'une phrase valide. Cette propriété, tout comme le bon typage des expressions su et ob n'est pas garantie par le formalisme : elle est une conséquence (prouvable) des contraintes de validité associées à la structure abstraite, et des règles de linéarisation fournies. La mise au point d'un système d'annotation de types plus poussé, incluant des conditions logiques (tel qu'évoqué lors de la description du système de typage de notre langage de macro, section 3.5.3), permettrait potentiellement de vérifier ce type de propriétés automatiquement.

Les règles de linéarisation associées à la seconde et à la troisième production demeurent inchangées. La cinquième production, permettant de réécrire un argument comme une complétive, distingue le cas où celle-ci contient une extraction (le type de la variable c est alors $e \rightarrow (e \rightarrow t) \rightarrow t$) de celui où ce n'est pas le cas (auquel cas le type de c est $(e \rightarrow t) \rightarrow t$). Dans les deux cas, la réalisation associée à l'argument est celle de la complétive qui le compose (c), sans modification. Enfin, la sixième règle de linéarisation modifie un argument a au moyen d'une relative r : ce traitement s'effectue en appliquant a (qui a le type d'un quantificateur) au prédicat appliquant l'élément x (qui sera identifié avec l'élément quantifié par a) à la relative r . Cet élément est ainsi identifié avec la variable r qui apparaissait dans la première règle de linéarisation dans le contexte d'une relative. La réalisation de la relative est ensuite complétée par une continuation vide (substituant la valeur de vérité constante \top à la continuation de l'évènement $C e$). Enfin, une nouvelle continuation $P x$ est ajoutée à la réalisation du syntagme nominal modifié, afin de préserver son type, selon la construction usuelle. Observons que cette règle suppose que la relative est rattachée à un groupe nominal formé (incluant un déterminant), ce qui n'est pas toujours justifié linguistiquement.

4.5 Traitement du contrôle et linéarisations synchrones

Nous souhaitons maintenant inclure à notre traitement des subordinées des propositions infinitives, introduites par des verbes à contrôle (qualifiés de verbes caténatifs en anglais) ; ces constructions incluent des énoncés tels que « Thomas souhaite partir. ». Sur le plan syntaxique, la proposition infinitive

qui sert d'argument objet à un tel verbe est dénuée de sujet explicite. Du point de vue sémantique, l'argument correspondant à son sujet est l'un des arguments du verbe contrôleur, qui est dupliqué au profit de la subordonnée. Ainsi, dans l'exemple précédent, Thomas est à la fois l'acteur du souhait et celui du départ, bien qu'il n'apparaisse qu'une fois dans la forme de surface.

L'argument dupliqué par le verbe contrôleur n'est pas nécessairement son sujet ; il peut également s'agir d'un autre objet pour lequel le verbe sous-catégorise, comme dans « Thomas demande à Paul de partir. », où Paul est à la fois le destinataire de la demande et l'acteur du départ. L'interprétation sémantique que nous souhaitons donner à ce phénomène est que la proposition contrôlée (la subordonnée infinitive) attend un argument (correspondant à son sujet) et qu'une propriété lexicale du verbe contrôleur (liée à sa sous-catégorisation), « décide » lequel de ses arguments (sujet ou second objet) est dupliqué au bénéfice de sa subordonnée.

Outre cette analyse en sémantique, nous désirons proposer de multiples linéarisations en surface, vers plusieurs langues européennes. En effet, des problèmes uniques liés à l'ordre des mots apparaissent dans des constructions impliquant plusieurs subordonnées infinitives imbriquées par des verbes à contrôle. Ainsi, dans des exemples tels que « Jean ordonne à Marie de demander à Paul de transmettre à Luc de . . . », chaque verbe (à l'exception du dernier dans l'ordre linéaire) est un verbe à contrôle, doté de deux objets, dont l'un est une subordonnée infinitive, et l'autre un syntagme nominal (plus une éventuelle préposition). Des langues comme le français et l'anglais privilégient dans ces cas un ordre séquentiel, alternant objets nominaux et subordonnées ($s_1 v_1 o_1 v_2 o_2 v_3 o_3 \dots$). L'allemand, en revanche, rejette le verbe d'une proposition subordonnée en dernière position, produisant une construction parenthésée (adoptant de manière canonique la forme $s_1 v_1 o_1 o_2 o_3 \dots v_3 v_2$) ; un certain degré de liberté est toutefois permis dans l'ordre des arguments des subordonnées, requérant un traitement spécifique tel que proposé dans le chapitre 5. Enfin, le néerlandais présente dans ces phrases un phénomène de dépendances croisées en série ($s_1 v_1 o_1 o_2 o_3 \dots v_2 v_3 \dots$). Ce dernier ordre échappe à l'analyse par une grammaire hors-contexte, comme argumenté par Shieber [1985] : bien que la structure abstraite de ce type de construction puisse être exprimée par un langage régulier d'arbres, une linéarisation naïve (qui se contente de concaténer les réalisations des différents sous-nœuds dans un certain ordre) ne peut pas produire la réalisation attendue. Nous emploierons pour le néerlandais des paires de chaînes, décrites et manipulées par le lambda-calcul, afin de surmonter cette difficulté.

Structure abstraite du phénomène de contrôle Afin de modéliser la structure des verbes à contrôle objet, qui sous-catégorisent pour deux compléments distincts, nous introduisons une nouvelle étiquette d'arête obj_2 dans nos structures abstraites. Celles-ci seront partagées entre les trois langues cibles,

qui ne se distingueront que lors du processus de linéarisation. Ce choix de modélisation est quelque peu arbitraire, dans la mesure où la sous-catégorisation des verbes à contrôle peut varier subtilement d’une langue à l’autre : il pourrait être souhaitable de distinguer, par exemple, un objet second accusatif d’un objet datif, indirect [cf. Butt *et al.*, 1999, p. 48-51]. Nous nous satisfaisons ici de cette approximation dans la mesure où notre but est d’illustrer l’emploi du formalisme décrit dans le chapitre 3, sans avoir l’ambition de construire une grammaire multilingue à large couverture.

La figure 4.5 détaille la structure abstraite d’un syntagme nominal modifié par une relative contenant des verbes à contrôle imbriqués. La réalisation en français de cet énoncé partiel (qui devrait, selon notre grammaire, être employé comme argument dans une structure plus large) serait la chaîne « un livre que Jean laisse Marie aider Anne à lire ». Bien que contestable en français en raison de l’extraction à travers un complément datif, cette phrase se linéarise correctement dans les langues cibles de la grammaire. Observons que le sujet d’une proposition contrôlée est absent de la structure profonde : il doit en effet être déduit en sémantique de la nature du verbe contrôleur et de ses arguments (c’est Marie qui apporte l’aide, et Anne qui effectue la lecture). Observons également que la linéarisation en sémantique de la relative à droite doit correctement abstraire l’objet du verbe **lire**, et le faire correspondre à l’antécédent (**un livre**) ; et que, comme précédemment, le pronom relatif **que** correspondant doit être antéposé en surface.

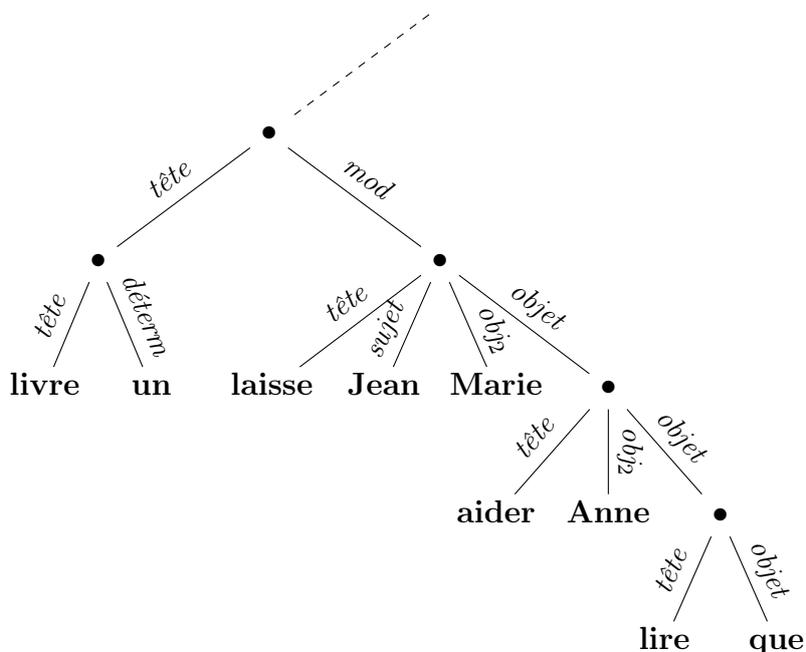


FIGURE 4.5 – Structure d’un syntagme incluant des proposition contrôlées

Mise à jour du lexique Afin de contraindre les structures abstraites de manière appropriée, nous intégrons à notre lexique des verbes à contrôle (sujet et objet) ainsi que la forme infinitive des verbes existants. Les propriétés associées à ces entrées sont respectivement **contrôle sujet**, **contrôle objet** et **infinitif**. Notons que, comme pour les noms dont les propriétés d'accord sont sous-spécifiées, un même verbe peut apparaître plusieurs fois dans le lexique, avec ou sans les propriétés **contrôle sujet** et **contrôle objet**. Tout comme la structure abstraite des énoncés, les entrées lexicales seront communes aux trois langues cibles. Chaque entrée disposera cependant, outre de sa réalisation en sémantique, de trois réalisations distinctes en surface, correspondant à chacune des linéarisations synchrones.

La table 4.6 propose un ensemble d'entrées lexicales liées au contrôle, ainsi que les réalisations qui leur sont attachées.

Entrée	Propriétés	en	de	nl
vouloir	verbe, infinitif, transitif, contrôle sujet	want	willen	wilen
aider	verbe, infinitif, transitif, contrôle objet	help	helpen	helfen
laisse	verbe, transitif, contrôle objet	lets	lässt	laat
lire	verbe, infinitif, transitif, objet nominal, complétive objet	read	lesen	lezen

TABLE 4.6 – Entrées lexicales liées au contrôles et réalisations

Vocabulaire logique Nous introduisons deux nouveaux prédicats permettant de spécifier des contraintes de grammaticalité sur les verbes à contrôle. Le premier caractérise l'ensemble des verbes à contrôle en regroupant les propriétés lexicales **contrôle sujet** et **contrôle objet**; le second dénote une proposition contrôlée (tenant lieu d'argument objet pour un verbe contrôleur).

$$\begin{aligned} \text{contrôleur}(x) &\triangleq \text{contrôle sujet}(x) \vee \text{contrôle objet}(x) \\ \text{contrôlée}(x) &\triangleq \exists v. \text{contrôleur}(v) \wedge \text{tête} \uparrow \text{objet}(v, x) \end{aligned}$$

De plus, les linéarisations vers l'allemand et le néerlandais requièrent un ordre des mots différent entre propositions indépendantes et subordonnées (le verbe étant, pour ces dernières, rejeté à la fin). Nous obtiendrons les réalisations attendues en incluant des conditions appropriées dans les règles de linéarisation, au moyen des prédicats *indépendante* et *subordonnée*, définis comme suit : dans notre grammaire, une proposition est indépendante si elle n'est dominée par aucune autre proposition, et subordonnée dans le cas contraire.

$$\begin{aligned} \text{indépendante}(x) &\triangleq \text{proposition}(x) \wedge \nexists p. \text{proposition}(p) \wedge \text{dom}(p, x) \\ \text{subordonnée}(x) &\triangleq \text{proposition}(x) \wedge \neg \text{indépendante}(x) \end{aligned}$$

Grammaire support et contraintes logiques Nous sommes maintenant en mesure d’actualiser les productions de notre grammaire, ainsi que les contraintes logiques associées, pour inclure les phénomènes de contrôle.

La principale modification porte sur la première production, permettant l’ajout d’un argument objet supplémentaire (optionnel, étiqueté *o2*) et rendant l’argument sujet optionnel (celui-ci étant absent dans le cas d’une infinitive contrôlée). La nouvelle version de la première production est illustrée par la figure 4.6.

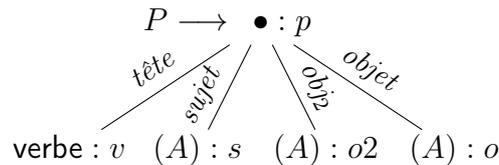


FIGURE 4.6 – Première production actualisée pour le contrôle

Par suite, nous modifions les contraintes logiques portant sur cette production, en incluant tout d’abord deux contraintes supplémentaires portant sur les propositions infinitives. D’une part, dans le cadre de notre grammaire, les propositions infinitives coïncident avec les propositions subordonnées contrôlées :

$$\text{contrôlée}(p) \Leftrightarrow \text{infinitif}(v)$$

D’autre part, en l’absence de sujet, la contrainte d’accord sujet-verbe est trivialement satisfaite. Nous conditionnons donc son application à l’existence d’un sujet de la manière suivante :

$$\text{avec}(s) \Rightarrow \text{accord_nombre}(s, v) \wedge \text{accord_personne}(s, v)$$

Enfin, nous modifions le prédicat *sous-catégorisation* pour traiter le cas des verbes contrôleurs, en tenant compte du second objet, de l’optionnalité du sujet et des complétives infinitives :

- Un sujet est absent si et seulement si son verbe est à l’infinitif.
- Un objet second est l’argument d’un verbe à contrôle objet, et doit être nominal.
- Un verbe peut sous-catégoriser pour une complétive objet parce qu’il est un verbe à contrôle, ou parce qu’il possède la propriété lexicale spécifique **complétive objet**.

La définition actualisée du prédicat résultant (d’arité 4) est donnée ci-dessous, chaque ligne correspondant à une « règle » de sous-catégorisation. Les trois conditions spécifiées ci-dessus sont implémentées respectivement par la première, la troisième et la quatrième ligne, le reste de la définition demeurant inchangé. En outre, la contrainte de bonne formation imposée précédemment selon laquelle les arguments de la première production doivent satisfaire le prédicat *sous-catégorisation* est maintenue.

$$\begin{aligned}
\text{sous-catégorisation}(v, s, o, o2) &\triangleq \text{sans}(s) \Leftrightarrow \text{infinitif}(v) \\
&\wedge \text{avec}(o) \Rightarrow \text{transitif}(v) \wedge \text{sans}(o) \Rightarrow \text{intransitif}(v) \\
&\wedge \text{avec}(o2) \Leftrightarrow \text{contrôle objet}(v) \Leftrightarrow \text{synt_nominal}(o2) \\
&\wedge \text{proposition}(o) \Rightarrow \text{complétive objet}(v) \vee \text{contrôleur}(v) \\
&\wedge \text{synt_nominal}(o) \Rightarrow \text{objet nominal}(v) \\
&\wedge \text{proposition}(s) \Rightarrow \text{complétive sujet}(v) \\
&\wedge \text{synt_nominal}(s) \Rightarrow \text{sujet nominal}(v)
\end{aligned}$$

Linéarisation en sémantique Pour la plupart des productions, les règles de linéarisation sémantique restent inchangées, hormis l’avant dernière, qui doit tenir compte de contraintes de typage supplémentaire : un argument formé par une complétive peut en effet avoir le type $e \rightarrow e \rightarrow (e \rightarrow t) \rightarrow t$: le second e manquant à gauche représente alors, comme précédemment, l’élément associé à un pronom relatif ; et le premier correspond au sujet d’une proposition infinitive contrôlée.

Ce mécanisme s’explique en examinant la règle de linéarisation associée à la première production, donnée figure 4.7. Celle-ci est passablement complexe, mais le langage de macros adopté dans le chapitre 3 permet de gérer cette complexité, en cloisonnant les différents composants de la linéarisation afin de les considérer indépendamment.

Remarquons tout d’abord que cette règle de linéarisation conserve le même schéma que précédemment, son terme principal étant factorisé par la variable ph . Ce terme abstrait la continuation C de l’évènement dénoté par la proposition, puis emploie la construction habituelle par montée de type pour traiter la quantification de chacun de ses trois arguments, su , ob et $o2$. Observons que l’objet second $o2$ est traité avant l’objet ob . Enfin, le terme ph quantifie sur l’évènement e associée à la proposition, applique sa continuation, puis incorpore la sémantique du noyau verbal nv . Ce dernier applique simplement le terme représentant le sens du verbe à ses arguments, en fonction de ses contraintes de sous-catégorisation. Remarquons que, sémantiquement, tout verbe possède au moins un argument en plus de son évènement, même s’il s’agit d’un infinitif dénué de sujet en surface.

Examinons maintenant chacun des arguments (su , ob et $o2$) séparément. Par défaut, l’argument sujet su est simplement la réalisation s associée au sous-terme correspondant. Comme précédemment, il peut également s’agir d’un pronom (relatif) extrait, auquel cas la réalisation emploie la variable libre r qui conserve le même usage que précédemment. Enfin, il est possible que la proposition soit contrôlée, auquel cas l’argument sujet du verbe est déterminé (et quantifié) par la proposition parente ; dans ce dernier cas, la variable libre s' est employée, d’une manière similaire à la variable r pour l’élément désigné par un pronom relatif.

$$\begin{array}{c}
 P \longrightarrow \bullet : p \\
 \begin{array}{ccc}
 \text{tête} & & \text{objet} \\
 \swarrow & & \searrow \\
 \text{verbe} : v & (A) : s & (A) : o2 \quad (A) : o \\
 \swarrow & \text{sujet} & \searrow
 \end{array}
 \end{array}$$

$$\left\{ \begin{array}{l}
 \text{contrôlée}(p) \longrightarrow \left\{ \begin{array}{l} \text{prop_ext}(p) \longrightarrow \lambda s'.\lambda r.ph \\ \text{sinon} \longrightarrow \lambda s'.(\lambda r.ph) \Omega^e \end{array} \right\} \\
 \text{sinon} \longrightarrow \left\{ \begin{array}{l} \text{prop_ext}(p) \longrightarrow \lambda r.(\lambda s'.ph) \Omega^e \\ \text{sinon} \longrightarrow (\lambda r.\lambda s'.ph) \Omega^e \Omega^e \end{array} \right\}
 \end{array} \right\}$$

où $ph = \lambda C.su \lambda x.o2 \lambda z.ob \lambda y.\exists e.(C e) \wedge nv$

$$\text{où } su = \left\{ \begin{array}{l} \text{contrôlée}(p) \longrightarrow \lambda P.P \ s' \\ \text{extrait}(s) \longrightarrow s \ r \\ \text{sinon} \longrightarrow s \end{array} \right\}$$

$$\text{et } ob = \left\{ \begin{array}{l} \text{extrait}(o) \longrightarrow o \ r \\ \text{prop_ext}(o) \longrightarrow \left\{ \begin{array}{l} \text{contrôle objet}(v) \longrightarrow o \ x \ r \\ \text{contrôle sujet}(v) \longrightarrow o \ z \ r \\ \text{sinon} \longrightarrow o \ r \end{array} \right\} \\ \text{sinon} \longrightarrow \left\{ \begin{array}{l} \text{contrôle objet}(v) \longrightarrow o \ x \\ \text{contrôle sujet}(v) \longrightarrow o \ z \\ \text{sinon} \longrightarrow o \end{array} \right\} \end{array} \right\}$$

$$\text{et } nv = \left\{ \begin{array}{l} \text{contrôle objet}(v) \longrightarrow v \ e \ x \ y \ z \\ \text{sinon} \longrightarrow \left\{ \begin{array}{l} \text{transitif}(v) \longrightarrow v \ e \ x \ y \\ \text{sinon} \longrightarrow v \ e \ x \end{array} \right\} \end{array} \right\}$$

FIGURE 4.7 – Linéarisation sémantique associée à la première production

Considérons maintenant l'argument objet *ob* : il fait l'objet du même traitement que précédemment pour l'extraction, distinguant les cas *extrait(o)* et *prop_ext(o)* où l'objet requiert un argument supplémentaire *r* se rapportant au pronom relatif. Cependant, un argument supplémentaire est inséré dans le cas où *v* est un verbe contrôleur : celui-ci doit en effet dupliquer son sujet (*x*) ou son objet second (*z*) au profit de sa complétive infinitive, selon le type de contrôle exercé par le verbe.

Enfin, le dernier argument *o2* est employé directement : nous ne traitons ici pas la possibilité pour ce dernier de faire l'objet d'une extraction. Le traitement de ce cas par une extension de la grammaire actuelle ne présente pas de difficulté particulière, hormis sur un point délicat : la correspondance supposée jusqu'à présent entre la fonction profonde d'un pronom relatif (*sujet, objet*) et sa forme dans le lexique (**nominatif, accusatif**) constitue une approximation qui atteint ici ses limites. Pour certains verbes, le pronom requerrait en français l'ajout de la préposition « à » suggérant un datif, alors que la forme accusative simple est préférée ailleurs (en particulier pour les verbes de perception comme voir/entendre).

Enfin, considérons le contexte dans lequel le terme *ph* apparaît : celui-ci tient compte des deux éléments susceptibles d'être manquants. Si la proposition *p* est contrôlée, alors la variable libre *s'*, correspondant à son sujet manquant, doit être abstraite dans la réalisation de la proposition. En outre, comme précédemment, la réalisation doit également abstraire *r* lorsqu'une extraction est en jeu. La réalisation d'une proposition a donc le type $e^k \rightarrow (e \rightarrow t) \rightarrow t$, où $k \in \{0, 1, 2\}$: si $k = 0$, la proposition est complète ; si $k = 1$, elle fait l'objet soit d'un contrôle, soit d'une extraction, et la variable correspondante est abstraite ; enfin, si $k = 2$, la proposition est contrôlée et fait également l'objet d'une extraction, et les deux variables correspondantes sont laissées abstraites. Observons dans ce cas que l'ordre de ces abstractions (*s'* puis *r*) correspond à l'ordre d'application des arguments dans la construction de *ob* (traitant d'abord l'argument lié au contrôle, puis celui des extractions).

Les autres règles de linéarisation demeurent inchangées, à l'exception de la cinquième, qui doit tenir compte du type de son unique variable *c*. Nous distinguons pour cela quatre cas, liés à la présence ou à l'absence d'une abstraction due au contrôle ou au pronom relatif. Seul le type de la variable *c* varie d'un cas à l'autre, le terme lui-même demeurant inchangé.

- | | | |
|----|--|---|
| 2. | $ \begin{array}{c} A \longrightarrow \bullet \\ \begin{array}{cc} \text{déterm} & \text{tête} \\ \swarrow & \searrow \\ \text{déterminant} : d & \text{nom} : n \end{array} \end{array} $ | $\lambda P.d \lambda x.n x \wedge P x$ |
| 3. | $A \longrightarrow \text{nom propre} : np$ | $\lambda P.P np$ |
| 4. | $A \longrightarrow \text{pronom} : p$ | $ \left\{ \begin{array}{ll} \text{extrait}(p) & \longrightarrow \lambda r.\lambda P.P r \\ \text{sinon} & \longrightarrow p \end{array} \right\} $ |

$$\begin{array}{l}
 5. \quad A \longrightarrow P : c \quad \left\{ \begin{array}{l} \text{contrôlée}(c) \longrightarrow \left\{ \begin{array}{l} \text{prop_ext}(c) \longrightarrow c \\ \text{sinon} \longrightarrow c \end{array} \right\} \\ \text{sinon} \longrightarrow \left\{ \begin{array}{l} \text{prop_ext}(c) \longrightarrow c \\ \text{sinon} \longrightarrow c \end{array} \right\} \end{array} \right\} \\
 6. \quad \begin{array}{c} A \longrightarrow \bullet \\ \text{tête} \swarrow \quad \searrow \text{mod} \\ A : a \quad P : r \end{array} \quad \lambda P.a \lambda x.(r \ x \ \lambda e.\top) \wedge P \ x
 \end{array}$$

Linéarisations synchrones vers la forme de surface Nous décrivons maintenant trois ensembles de règles de linéarisation, afin de produire la forme de surface attendue pour nos structures abstraites modélisant le contrôle en anglais, allemand et néerlandais, en respectant les contraintes d'ordre des mots mentionnées précédemment.

La plupart des règles de linéarisation sont communes aux trois langues et demeurent inchangées : les productions 2, 3 et 4 conservent les règles de linéarisation précédentes, qui concatènent simplement les entrées lexicales dans l'ordre usuel. La sixième règle de linéarisation, rappelée ci-dessous est également inchangée, employant une requête logique pour antéposer le pronom à la relative :

$$6. \quad \begin{array}{c} A \longrightarrow \bullet \\ \text{tête} \swarrow \quad \searrow \text{mod} \\ A : a \quad P : r \end{array} \quad \text{extrait}(p) \wedge \text{chemin_wh}(r, p) \longrightarrow a \ \mathbf{p} \ r$$

La cinquième production, qui introduisait manuellement la conjonction de subordination **que**, doit être modifiée : les propositions infinitives (contrôlées) ne requièrent pas de conjonction. En outre, la conjonction employée dépend de la langue cible (**that**, **dass** ou **dat** respectivement) :

$$5. \quad A \longrightarrow P : c \quad \left\{ \begin{array}{l} \text{contrôlée}(c) \longrightarrow c \\ \text{sinon} \longrightarrow \mathbf{that/dass/dat} \ c \end{array} \right\}$$

Enfin, la première production est celle qui construit les propositions, et doit répondre au problème de l'ordre des mots. Nous détaillons maintenant la règle de linéarisation associée pour chacune des trois langues cibles. Afin d'explicitier les réalisations attendues pour chacun des cas, la figure 4.8 détaille la forme de surface résultant d'une même structure abstraite en anglais, allemand et néerlandais (en reprenant l'exemple de structure abstraite donné par la figure 4.5).

Les mots obtenus lors de la réalisation des feuilles de la structure abstraite sont soulignés, et regroupés selon la proposition dont ils dépendent immédiatement. La flèche en pointillés illustre le mouvement du pronom extrait (correspondant au pronom relatif **que** en français).

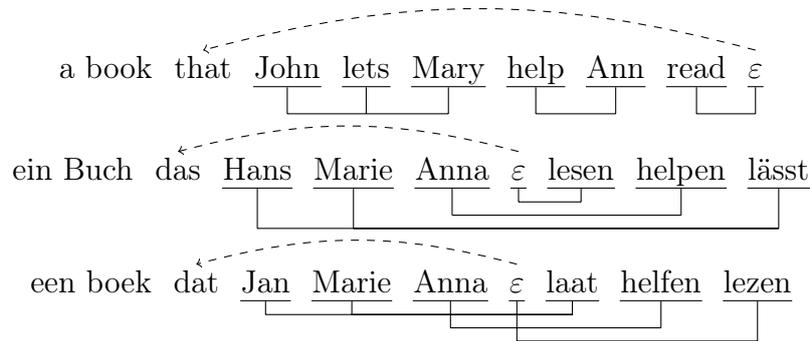
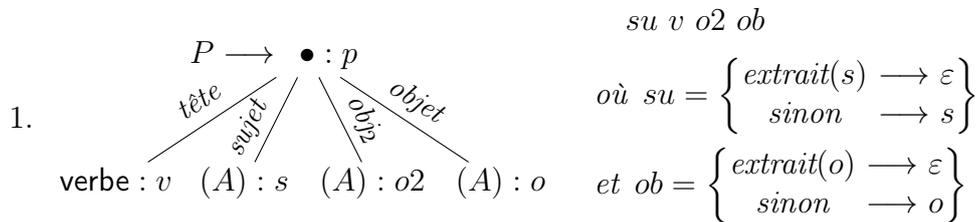


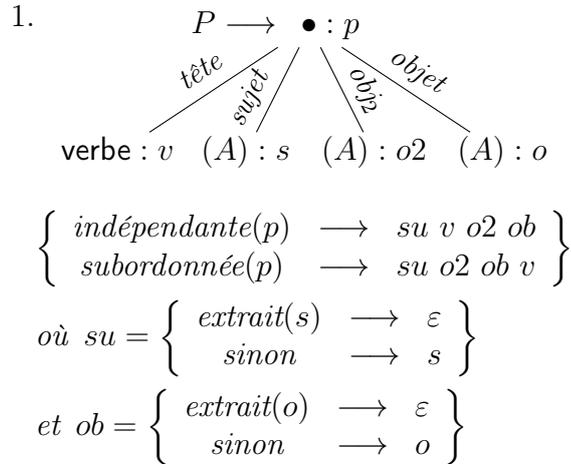
FIGURE 4.8 – Réalisations multilingues et ordre des mots

Anglais La règle de linéarisation pour l’anglais est similaire à celle adoptée jusqu’ici. L’ordre canonique place le second complément d’un verbe contrôleur en première position, suivi de la subordonnée infinitive tenant lieu d’objet. La règle de linéarisation résultante est la suivante :



Allemand Pour notre grammaire, dans le cadre d’une proposition indépendante, l’ordre des mots allemand est identique à celui de l’anglais ; plaçant le verbe en deuxième position, à la suite de son sujet. En revanche, l’allemand impose de rejeter le verbe à la fin dans le contexte d’une subordonnée : ainsi, l’ordre des composants dans une proposition telle que « Er gewann das Rennen. » (« Il a gagné la course. ») devient « Ich weiss, dass er das Rennen gewann. » (« Je sais qu’il a gagné la course. »).

L’analyse du contexte dans lequel une proposition est réalisée (subordonnée ou indépendante) s’effectue au moyen d’une condition de réalisation simple, sélectionnant l’ordre des mots approprié :



Observons que cette règle ne produit que l'ordre canonique pour les arguments du verbe : des réalisations multiples permettraient d'inclure les autres ordres possibles (autorisant, par exemple, à mettre l'accent sur l'objet du verbe en le plaçant en position initiale). D'autres phénomènes plus complexes, autorisant le mélange entre des arguments appartenant à plusieurs propositions distinctes, seront évoqués et traités dans le prochain chapitre.

Néerlandais L'ordre des mots en néerlandais est en général similaire à celui de l'allemand, à l'exception des constructions impliquant plusieurs verbes à contrôle imbriqués. Dans ce cas, la séquence formée par les verbes rejetés à la fin est inversée : les verbes de chaque subordonnée apparaissent dans l'ordre où leurs propositions ont été introduites (et non dans l'ordre inverse).

Ce phénomène, connu sous le nom de dépendances croisées en série, ne permet pas à une règle de linéarisation simple (choisissant simplement un ordre pour les composants de chaque production) de produire le résultat attendu. Une analyse commode de ce phénomène consiste à produire non pas une chaîne, mais une paire de chaînes en guise de réalisation : le premier élément de la paire contient les arguments des subordonnées, concaténés par ordre d'apparition ; tandis que le second accumule simultanément les verbes, dans le même ordre. La proposition contrôleuse initiale peut alors produire la réalisation attendue en concaténant directement les deux éléments de la paire.

Nous détaillons maintenant la règle de linéarisation que nous associons aux propositions en néerlandais, donnée par la figure 4.9.

Les deux premières alternatives de réalisation, en l'absence de contrôle, sont identiques à celles données pour l'allemand : l'ordre SVO prévalant dans une proposition indépendante, et SOV dans une subordonnée. Par suite, une proposition subordonnée contrôlée a pour réalisation une paire de chaînes, dont le composant gauche est formé par ses arguments objets placés dans l'ordre usuel (avec une éventuelle complétive en dernier), et le composant droit contient le verbe. Une proposition contrôlée étant une infinitive (et donc dénuée

4.6 Conclusion

Dans ce chapitre, nous avons mis en pratique le formalisme décrit dans le chapitre précédent, afin de concevoir et de faire évoluer une grammaire couvrant des phénomènes linguistiques divers en syntaxe et en sémantique, parmi lesquels l'accord, le mouvement *wh*, le contrôle et l'ordre des mots. Dans une large mesure, les outils de description proposés se sont avérés adéquats dans le cadre d'un travail de conception grammaticale portant sur des phénomènes non-triviaux. Ils permettent la mise en œuvre des concepts linguistiques abordés, et la concision des descriptions résultantes contribue à la maintenabilité de la grammaire résultante, de même que le nommage des formules logiques (utilisant l'opérateur \triangleq) et de certains lambda-termes apparaissant dans les règles de linéarisation (au moyen de notre langage de macros). L'emploi de requêtes logiques facilite en outre la description des phénomènes liés au mouvement.

Une observation intéressante soulevée par l'emploi de ce formalisme est l'existence de nombreuses similarités entre la conception d'une grammaire enrichie et de ses linéarisations, et la programmation en général :

- la relative indépendance entre les différents composants d'une grammaire enrichie (et les éventuelles linéarisations associées) évoque la décomposition d'un programme en modules ;
- le nommage des différents composants d'une grammaire (propriétés lexicales, prédicats et relations ajoutés au vocabulaire logique, étiquettes décorant les productions et variables apparaissant dans les règles de linéarisation) bénéficie de l'intuition fournie par les règles en usage pour le choix des identifiants dans un programme ;
- bien que le formalisme soit conçu pour faciliter l'implémentation et la compréhension de phénomènes linguistiques, les formules logiques et règles de linéarisation les plus complexes bénéficient grandement de l'ajout de commentaires, sous la forme d'annotation expliquant divers choix de mise en œuvre ;
- certains concepts, tels que l'opposition entre une implémentation statique (plus simple) ou dynamique (plus correcte et adaptable) resurgissent, par exemple dans notre modélisation de l'accord, où toutes les informations morphologiques sont pré-compilées dans le lexique (*cf.* table 4.4), plutôt que calculées dynamiquement d'après un ensemble de règles de linéarisation morphologiques ;
- l'exercice d'une discipline stricte vis-à-vis des types donnés aux réalisations des non-terminaux (par les règles de linéarisation associées aux productions) contribue grandement à garantir la cohérence sémantique des réalisations – à l'inverse, certaines réalisations imprévues (et non souhaitables) peuvent être exclues en vérifiant les types associés à certaines règles de linéarisation (c'est particulièrement le cas pour la linéarisation sémantique du contrôle (figure 4.7), où une proposi-

- tion complétive peut avoir quatre types différents en fonction de son contexte) ;
- certaines erreurs susceptibles d’apparaître lors de la modification d’une grammaire (par exemple, lors de la mise à jour de la définition d’un prédicat logique à la lumière de nouvelles informations linguistiques) peuvent être prévenues par l’emploi d’assertions dans les contraintes logiques : en spécifiant plusieurs contraintes partiellement ou totalement redondantes compte tenu des autres contraintes et de la grammaire support, le langage abstrait peut devenir vide de tout énoncé si une modification invalide de manière inattendue une supposition antérieure. Par exemple, notre définition du prédicat *prop_ext(x)* (p. 89) précise, inutilement, que *x* doit être une proposition ; mais tout changement dans la définition de la relation *chemin_wh* qui y inclurait l’arête *mod* risquerait d’entraîner, de façon inattendue, qu’un syntagme nominal soit considéré comme une proposition contenant une extraction, introduisant un « bug » dans la grammaire. D’autres exemples incluent la contrainte d’unicité portant sur la proposition relative associée à chaque pronom relatif (p. 90) ; ou, négativement, notre définition d’une proposition indépendante (p. 99, qui ne doit être dominée par aucun *nœud* dans la structure abstraite, mais devrait en toute rigueur n’être dominée par aucune *proposition*) ; ou encore celle d’un syntagme à la troisième personne (p. 83, qui suppose, peut-être trop arbitrairement, qu’un pronom personnel ne pourra jamais être modifié ou déterminé dans la structure abstraite) ;
 - le langage de macros décrit section 3.5.1 dans les règles de linéarisation permet de factoriser certaines constructions, mais soulève les mêmes question de lisibilité que la factorisation de fragments de code : un compromis doit être établi entre la répétition d’un lambda-terme dont une petite partie change d’un copier-coller à l’autre, et une généralisation excessive compliquant inutilement une règle de linéarisation ;
 - enfin, le développement d’une grammaire dans le but de sanctionner ou d’exclure certains exemples ou contre-exemples linguistiques, tels que ceux proposés pour expliciter les contraintes d’îlot (p. 87), ressemble de très près à la méthodologie du développement piloté par les tests (*Test-Driven Development*) en ingénierie logicielle.

Limites et discussion À l’usage, toutefois, certaines limitations dans les capacités du formalisme deviennent apparentes. Certaines redondances dans les règles de linéarisation ne peuvent pas être factorisées par notre langage de macros, en raison de son manque de granularité sur les conditions : les formules logiques apparaissant dans les conditions ne peuvent qu’être fusionnées deux à deux. Par conséquent des termes qui ne diffèrent que par l’emploi d’une variable dans une pré-condition, comme *su* et *ob* dans les modélisations impliquant le

mouvement (p. 92), ne peuvent être factorisés ; il s’agit toutefois d’un problème assez mineur.

Une autre source de lourdeur, due à la discipline de typage choisie pour nos lambda-termes, est la présence de règles de linéarisation telles que celle de la cinquième production en sémantique page 103 : dans tous les cas, la réalisation attachée à la production est la même (c). Cependant, son type peut varier en fonction du contexte, conduisant à employer la variable c avec des types différents sous différentes pré-conditions. Le fait de forcer cette distinction aide à garantir la cohérence de notre réalisation sémantique (en rendant explicites les suppositions faites sur le type des non-terminaux), mais oblige dans ces cas à produire des règles de linéarisation en apparence chargées, alors que leur réalisation est dans tous les cas égale à celle de leur unique argument. L’emploi d’un système de typage plus riche, tel qu’évoqué à la fin du chapitre précédent (voir page 3.5.3), pourrait résoudre ce problème et faciliter l’apport d’une solution au précédent.

Les règles de linéarisation en sémantique pourraient également bénéficier d’une simplification supplémentaire : elles sont actuellement l’élément le plus complexe apparaissant dans nos modélisations (figure 4.7), en grande partie à cause du traitement usuel des quantifications dans la sémantique de Montague. L’emploi de constructions à montée de type, ou plus particulièrement de continuations, répond à des besoins de compositionnalité issus d’autres traditions grammaticales. Il pourrait être opportun de les éliminer là où c’est possible, en mettant à contribution notre mécanisme de requêtes logiques. La réalisation d’un syntagme ne rendrait ainsi compte de sa quantification qu’une fois celui-ci complètement formé. Une relative perte de compositionnalité s’ensuivrait, en ce sens qu’il ne serait plus possible de déterminer complètement le sens d’un fragment partiel de la structure abstraite ; cependant, il peut sembler étrange que le sens d’un syntagme nominal incorpore une continuation se rapportant au sens du verbe à venir.

Pour terminer, le compte rendu qui est fait ici du phénomène de contrôle et de l’ordre des mots résultant est quelque peu superficiel. D’une part, contrairement à ce que nous avons supposé, la propriété de contrôle n’est pas complètement trans-linguistique : certains énoncés n’ont pas d’équivalent direct dans une langue voisine (par exemple, « John wants Mary to stay. » ne peut pas se traduire littéralement en français par une infinitive contrôlée), et nos arguments *objet* et *obj₂* ne tiennent pas compte des prépositions qui sont requises par certains verbes pour leurs arguments (infinitif avec ou sans *to* en anglais, objets indirects construits avec *à* en français, *etc.*). Enfin, les exemples illustrant les différences dans l’ordre des mots sont basés sur des constructions complexes, et peuvent par conséquent sembler artificiels, bien qu’ils transcrivent des phénomènes clairement attestés.

Certains de ces derniers problèmes semblent être inhérents au concept de grammaire synchrone multilingue. D’autres pourraient être couverts par notre

grammaire en énumérant et traitant toutes les constructions attestées (élargissement du lexique et de ses informations de valence ainsi que de la relation *sous-catégorisation()*, traitement spécifique des verbes de perception, *etc.*), et n'apporteraient pas grand chose de plus au propos de ce chapitre. En revanche, certains phénomènes d'ordre des mots posent des problèmes plus sérieux. C'est en particulier le cas de l'allemand, qui permet le mélange des arguments entre plusieurs propositions complétives imbriquées, un phénomène connu sous le nom de *scrambling*, que notre grammaire actuelle ne couvre pas. L'ensemble des réalisations résultantes, énumérant tous les ordres possibles entre les arguments des subordinées, ne peut être directement obtenu par les mécanismes de linéarisation utilisés jusqu'ici, et pose un problème de fond.

Plus généralement, d'autres langues offrant un grand degré de liberté dans l'ordre des mots requièrent une énumération exhaustive des ordres possibles, qui est peu commode à exprimer par nos règles de linéarisation. Afin de répondre à ces difficultés, nous introduisons dans le prochain chapitre un mécanisme de représentation compacte, dénotant des phrases modulo permutation de certains mots ou groupes de mots, et étudions ses propriétés algorithmiques. Les termes qui en résultent, représentables par le lambda-calcul, constituent une solution possible aux problèmes d'ordre des mots posés par des phénomènes comme le *scrambling*.