

PROJET EUREKA GENELEX

Rapport sur la COUCHE SYNTAXIQUE

Consortium GENELEX

Version 4.0

1 Décembre 1993

TABLE DES MATIERES

A - Préliminaires 1

1 : Modèle générique, modèle "théorie-accueillant" 3

1.1 : Cadre lexicographique et formalisme d'enregistrement 3

1.2 : Un modèle générique "théorie-accueillant" 4

1.3 : Expression de descriptions différentes dans le formalisme
GENELEX 5

1.4 : Fusion cohérente de différentes descriptions 6

1.4.1 : Faits linguistiques complémentaires 6

1.4.2 : Faits linguistiques concurrents mais non contradictoires 7

1.4.3 : Faits linguistiques concurrents et contradictoires 9

2 : Homogénéité de codage 10

3 : Articulation Lexique - Grammaire 11

3.1 : Compatibilité du lexique et de la grammaire 11

3.1.1 : Approche théorique et formalisme 11

3.1.2 : Compatibilité minimale requise par le modèle GENELEX
14

3.2 : Renvois du lexique vers la grammaire 15

4 : Autonomie de la Syntaxe 17

B - Analyse des Unités Syntaxiques Simples 19

I : Articulation avec la Couche Morphologique 21

II : Les Unités Syntaxiques (Usyn) 23

1 : Définition générale 23

2 : Point de vue atomiste 24

3 : Point de vue syntaxiste 26

4 : Critères d'éclatement des Usyn 27

III : Descriptions 29

IV : Constructions 30

V : Self 31

1 : Définition générale 31

2 : Self intervenant dans sa construction 31

3 : Self appelé par un élément externe à sa construction 33

VI : Positions 34

1 : Approche générale et justification 34

1.1 : L'exemple de la complémentation verbale 34

1.2 : Emergence de la notion de Position 34

1.3 : Cas de la détermination gauche 36

2 : Application aux autres catégories 37

2.1 : Les Noms 37

2.2 : Les Adjectifs 38

2.3 : Les Prépositions 39

2.4 : Les Déterminants 39

2.5 : Les Adverbes 40

3 : Définition de la Position 40

3.1 : Définition formelle 41

3.2 : Positions et Distributions 42

3.2.1 : Occupants de Position 42

3.2.2 : Partage des Positions 43

3.3 : Positions et Fonctions 44

3.4 : Positions et Rôles Thématiques 45

4 : Positions et Constructions 46

4.1 : Optionnalité 46

4.1.1 : Optionnalités simples 46

4.1.2 : Optionnalités distantes ou interdépendantes 47

4.2 : Actualisation des Positions 49

4.2.1 : Phénomènes linguistiques 49

4.2.2 : Indicateur de répétabilité sur les Positions 51

4.3 : Linéarité 52

4.3.1 : Linéarité canonique 52

4.3.2 : Linéarité et transformations 53

4.3.3 : Linéarité et optionnalité 53

4.4 : Phénomènes de surface 54

4.5 : Solidarité des Positions 54

4.6 : Réalisations de Positions inter-conditionnées 55

4.7 : Conditions entre Self et les Positions 56

VII : Syntagmes 57

1 : Définition générale 57

2 : Etiquette Syntagmatique 57

3 : Contraintes par Traits 58

4 : Contraintes de structure 59

4.1 : Le Trait de sous-catégorisation syntaxique 59

4.2 : La réécriture arborescente 59

4.2.1 : Description 59

4.2.2 : Exemples d'utilisation 60

4.3 : Réécriture partielle de Syntagmes 64

VIII : Les Traits 65

1 : Types, sous-types et valeurs de Traits 65

1.1 : Traits lexicaux 65

1.1.1 : Les Introdecteurs (Prep, Conj, ProRel, ProIntrog et Introd) 65

1.1.2 : Lex 67

1.1.3 : Cooccurrence de traits lexicaux 68

1.1.4 : Cas particulier des Agglutinés 68

1.2 : Traits morphologiques 69

1.3 : Traits morpho-syntaxiques 69

1.3.1 : SsCatMorph 69

1.3.2 : Aux 70

1.3.3 : Pronominal 71

1.3.4 : Accord 72

1.4 : Traits syntaxiques 72

1.4.1 : SsCatSynt 72

1.4.2 : Tournure 73

1.4.3 : Neg 74

1.4.4 : Passif 74

1.5 : Traits syntaxico-sémantiques 75

1.5.1 : Coref 75

1.6 : Traits sémantiques 78

1.6.1 : Aspect du verbe 78

1.6.2 : Conditions Dénotationnelles (Traits Bin) 78

1.6.3 : Classe Sémantique (Traits Libre) 79

2 : Contraintes d'intégrité sur les Traits 80

2.1 : Règles générales d'utilisation des Traits 80

2.2 : Sous-types de Traits autorisés selon l'étiquette syntagmatique
81

2.3 : Valeurs de Traits de sous-catégorie autorisées selon l'étiquette
syntagmatique 85

2.4 : Conjonctions possibles de Traits 88

IX : Transformations 90

1 : Transformations entre Usyn 91

1.1 : Transformations entre Usyn provenant de la même Um 91

1.2 : Transformations entre Usyn provenant d'Um différentes 91

1.3 : Calculatoire : le Mode de Transformation 92

2 : Transformations entre Descriptions 93

2.1 : Emploi et contexte d'utilisation 93

2.1.1 : Exemple de la passivation 94

2.1.2 : Exemple de restructuration de complétive 95

2.2 : Héritage des positions : observations 95

2.3 : Calculatoire : le Mode de Transformation 96

2.3.1 : Explicitation du processus 97

2.3.2 : Exemplification du processus 98

3 : Transformations entre Syntagmes occupant la même position 99

C - Analyse des Unités Syntaxiques Composées 101

I : Introduction 103

1 : Enregistrement des composés 103

2 : Dictionnaires électroniques, dictionnaires éditoriaux 103

3 : Couverture des composés dans GENELEX 104

II : Caractéristiques des composés syntaxiques 106

1 : Syntagmes bien formés 106

1.1 : Lexicalisation 106

1.2 : Flexion et accord 107

1.3 : Alternatives de réalisation 107

1.4 : Effacement 108

1.5 : Modification 109

1.6 : Déplacement 109

1.7 : Transformations 110

1.8 : Coordination 111

2 : Figement 111

2.1 : Figement total ou partiel 111

2.2 : Constituants solidaires 112

3 : Intérieur et extérieur du composé 112

3.1 : Assimilation à une catégorie terminale 112

3.2 : Distinction intérieur / extérieur d'un composé 114

3.3 : Frontière entre extérieur et intérieur du composé 114

4 : Héritage des propriétés syntaxiques des composants 115

4.1 : Héritage des constructions 116

4.2 : Héritage des caractéristiques propres au composant 117

5 : Verbes supports 118

III : Principes de la représentation formelle 119

1 : Axes de représentation 119

2 : Vision atomiste vs vision syntaxiste 120

IV : Listes de composants 121

1 : Type et choix des composants 121

2 : Alternatives de composants 122

3 : Référence aux composants 123

V : Mode de Composition (MdC) 124

1 : Intérêt 124

2 : Inhibition, héritage et filtrage 124

2.1 : Inhibition, héritage et filtrage des Positions 124

2.2 : Filtrage de Self 125

2.3 : Exemplification 125

3 : Appels des composants pour lexicalisation 125

3.1 : Notion d'appel 125

3.2 : Tête du composé syntaxique 126

3.3 : Lexicalisation 127

3.4 : Compatibilité entre les catégories du syntagme appelant et du composant appelé 127

3.5 : Unification 128

VI : Structure interne du composé 129

1 : Choix de la catégorie 129

2 : Syntagmes de Structure (Syntagme_NT_S) 130

3 : Insertions 131

4 : Solidarité 132

5 : Traits restrictifs 132

5.1 : Traits lexicaux 132

5.2 : Traits morphologiques 133

5.3 : Traits Accord et Coref 133

5.4 : Traits sémantiques 134

6 : Conditions 135

7 : Transformations 136

VII : Construction 137

D - Annexe aux Unités Syntaxiques Composées 139

exemple 1 : *prendre son pied* 141

exemple 2 : *être à l'aise / être à son aise* 142

exemple 3 : *(fil (de fer)) barbelé* 143

exemple 4 : *poulet (à la (mode)) basquaise* 144

exemple 5 : *moules marinières / (à la (mode)) marinière* 145

exemple 6 : *mettre en ŷuvre* 146

exemple 7 : *à ses risques et périls* 147

exemple 8 : *tirer les vers du nez* 148

exemple 9 : *au détriment de / à son détriment* 149

exemple 10 : *mettre en garde > être mis en garde par* 152

exemple 11 : *au cas où / en cas de / en ce cas* 153

exemple 12 : *carte bleue* 156

exemple 13 : *abattre carte ma"tresse* 157

exemple 14 : *tarte à la crème* 159

exemple 15 : *avoir/éprouver/vouer admiration pour/à* 161

exemple 16 : *boîte aux lettres / boîte à lettres* 164

exemple 17 : *la plupart de* 165

exemple 18 : *en conséquence* 165

E - Bibliographie 167

F - Manuel de l'Utilisateur 171

1 : Usyn 173

1.1 : Généralités 173

1.2 : Usyn simple 173

1.3 : Usyn composée 173

2 : Descriptions 174

3 : Self 174

3.1 : Self simple 174

3.1.1 : Self intervenant dans la Construction (IntervConst) 174

3.1.2 : Self appelé par un élément externe à sa Construction (ComportAppele) 176

3.2 : Self composé 176

4 : Constructions 177

4.1 : Définition générale 177

4.2 : Squelettes de Constructions 179

5 : Positions 179

6 : Insertions 180

7 : Conditions 180

7.1 : Propriétés générales 180

7.2 : Conditions sur l'optionnalité des Positions 181

7.3 : Conditions sur les réalisations de Position 182

8 : Syntagmes 183

9 : Traits 184

9.1 : Traits lexicaux 185

9.1.1 : Famille des traits Lex 185

9.1.2 : Famille des traits RefLex 187

9.2 : Traits morphologiques 188

9.3 : Traits morpho-syntaxiques 189

9.4 : Traits syntaxiques 191

9.5. : Traits syntaxico-sémantiques 192

9.6 : Traits sémantiques 192

10 : Transformations 193

10.1 : TransfUsyn et TransfDescription 193

10.2 : TransfUsyn 194

10.3 : TransfDescription 195

10.4 : TransfPosition 195

10.5 : TransfSyntagme 195

11 : Modifs 196

11.1 : ModifDescription 197

11.2 : ModifConstruction 197

11.3 : ModifPosition 197

11.4 : ModifSyntagme 198

11.5 : ModifIntervConst 198

12 : R_Compose et Composition 199

13 : Mode de Composition 200

13.1 : MdC 200

13.2 : HeritePosition 200

13.3 : FiltreSelf 201

14 : Structure (Syntagme_NT_S) 201

15 : Appellation, commentaire, exemple 202

G - Schémas Entités-Relations 203

1 : Unité Syntaxique, Description et Self 205

2 : Construction, Position_C et Optionnalite 206

3 : Syntagme_T et Syntagme_NT_C 207

4 : Condition, CheminPosition, CheminSyntagme 208

5 : Composition, MdC et ModifIntervConst 209

6 : Syntagme_NT_S, Position_S et Insertion 210

7 : ModifConstruction, TransfPosition et ModifPosition 211

8 : TransfSyntagme, ModifSyntagme_T et ModifSyntagme_NT 212

9 : TransfUsyn, TransfDescription et ModifDescription 213

H - DTD SGML 215

I : Introduction - Traduction du Modèle Conceptuel 217

II : DTD GENELEX commentée 218

1 : DTD *genelex.dtd* 218

2 : DTD *syntaxe.dtd* 220

3 : Entités *morpho.ent* 241

4 : Entités *syntaxe.ent* 242

5 : Entités *custom.ent* 243

6 : Contraintes *syntaxe.ctr* 244

III : Exemples de données balisées 249

A - PRELIMINAIRES

Ce rapport présente le travail réalisé par l'ensemble des partenaires du consortium GENELEX France pour la définition d'un modèle décrivant la partie syntaxique d'un dictionnaire.

Il fait suite au document de présentation comparable, produit par le même consortium, et intitulé "COUCHE MORPHOLOGIQUE Version 3.0". L'introduction générale et les déclarations préliminaires que l'on peut lire dans ce premier rapport restent valables ici. Les notions et les choix conceptuels détaillés dans ce même rapport morphologique ne sont pas réprécisés ici mais il y est fait parfois référence.

1 : Modèle générique, modèle "théorie-accueillant"

1.1 : Cadre lexicographique et formalisme d'enregistrement

Lors de la constitution d'un fonds lexicographique, il est une tâche qu'on isole clairement, celle du lexicographe qui enregistre pour chaque entrée du lexique les informations pertinentes.

Pour chacune des entrées, le lexicographe procède systématiquement à un certain nombre de tests, suivant une démarche heuristique, note les caractéristiques que lui permettent de dégager les critères donnés par une théorie linguistique, prend des décisions sur l'acceptabilité des constructions rencontrées, et traduit, dans un formalisme descriptif, les informations pertinentes. C'est assez dire que ce travail ne peut être mené sans qu'aient été définis en amont,

- d'une part, la théorie linguistique qui définit les notions, les grilles de critères déterminant la description de l'observable, parmi lesquels on peut citer :

- la définition de l'unité lexicale,
- l'identification, la structuration et les frontières du syntagme,
- le rattachement du groupe prépositionnel,
- une prise de position sur les notions de complément essentiel et de modifieur, qui peuvent être considérées comme des notions à distinguer ou pas. Dans le cas où le parti pris est celui de la distinction, alors il faut spécifier les critères définitoires.

- d'autre part, un formalisme d'enregistrement, langage formel de codage des informations qui détermine leur représentation (définition des objets formels manipulés, vocabulaire de description, règles de manipulation sur les objets, contraintes d'intégrité et de cohérence).

Dans la pratique éditoriale, les manuels de rédaction (confidentiels) remplissent ces fonctions et donnent, par le biais des consignes, l'ordre séquentiel et le contenu des tests à faire, ainsi que les modes de codage en respectant le cadre théorique retenu (qui, lui, donne la définition des notions).

Le formalisme de codage se limite aux exigences de la loi du genre : la lexicographie traditionnelle n'a recours au formalisme que pour structurer les textes (impression papier) et répondre aux besoins de la lecture par un humain. Il n'en va pas de même pour les dictionnaires de machine : le lecteur humain s'accommode très bien par exemple de références implicites, qu'il peut élucider grâce à sa connaissance du monde ou à sa connaissance plus ou moins consciente des structures de la langue, de la structuration des champs lexicaux. Le dictionnaire de machine ne doit contenir que des informations explicites enregistrées dans un formalisme non-ambigu. Dans le cas d'un dictionnaire applicatif, l'application détermine le formalisme descriptif, l'école de référence détermine les critères linguistiques retenus.

Quel que soit le type de dictionnaire dont il est question, trois compétences (distinctes mais néanmoins cumulables) sont toujours requises : celle du linguiste, celle du formalisateur et celle du lexicographe. Une instance de dictionnaire, éditorial ou machine, sera le produit de ces trois compétences.

1.2 : Un modèle générique "théorie-accueillant"

GENELEX doit représenter les diverses descriptions existantes, compte tenu de ce qu'elles dépendront du modèle théorique, du degré de finesse et des critères discriminants retenus par le lexicographe, et ce, quelle que soit son école. Ceci aura des répercussions :

- au niveau morphologique,
- au niveau du comportement syntaxique des unités lexicales
- au niveau sémantique
- au niveau des inter-relations entre les différentes informations codées.

Ces diverses descriptions apparaîtront soit dans plusieurs instances de dictionnaires, soit dans une seule et même instance construite par fusion de toutes les autres.

En fait, il faut distinguer un formalisme descriptif qui soit "théorie-accueillant" et une instance de dictionnaire qui, elle, utilise un formalisme dans une théorie donnée. Le pari de généralité sur les dictionnaires ne peut donc être tenu qu'au niveau du formalisme descriptif, pont entre les différentes théories linguistiques.

Le modèle GENELEX est le formalisme descriptif, et c'est en tant que tel qu'il prétend à la généralité.

Un formalisme descriptif est générique si et seulement si il permet l'enregistrement des faits linguistiques dégagés par les différentes théories.

1.3 : Expression de descriptions différentes dans le formalisme GENELEX

Soient différents dictionnaires sources, aussi différents en nature qu'en fonction que le DNT (Dictionnaire de Notre Temps - Hachette) et une table du Ladl. On veut traduire les informations contenues par chacun de ces dictionnaires dans le formalisme GENELEX.

Afin de permettre la lecture des exemples qui suivent, nous donnons ici une rapide description du formalisme adopté dans GENELEX. Au niveau de la couche syntaxique, l'entrée est l'Unité

Syntaxique (notée **Usyn**). Une Usyn représente l'un des comportements syntaxiques du mot décrit. Elle comporte une "description de base" (notée **db**). Celle-ci regroupe la "construction de base" (notée **cb**) et les particularités de l'entrée dans cette construction (portées par **Self**). Une construction est une liste de **Positions** pouvant s'instancier par des **Syntagmes**. On peut considérer que l'unité elle-même (décrite par Self) fait ou ne fait pas partie de cette construction, ce qui se traduit par l'indication ou la non-indication du point d'insertion de Self dans la construction. Par commodité, on distingue dans les exemples les différentes positions en leur attribuant un numéro d'ordre (P0, P1, P2, etc.), à l'exclusion de la position qui se réalise Self.

Format DNT (Dictionnaire Hachette de Notre Temps)

aimer V. tr. dir.

On déduit de l'étiquette transitif, que l'entrée *aimer* a la description syntaxique suivante :

Format GENELEX

aimer

db : SELF cb

SELF : IntervConst : V

cb : P0 SELF (P1)

P0 : SN

P1 : SN

Format Ladl

aimer T7

On déduit des critères d'entrée dans la table 7 que l'entrée *aimer* a la description syntaxique suivante :

Format GENELEX

aimer

db : SELF cb

SELF : IntervConst : V

cb : P0 SELF (P1)

P0 : SN

P1 : SN

P[SsCatSynt : COMPLETIVE]

[Mode : SUBJONCTIF]

1.4 : Fusion cohérente de différentes descriptions

Il faut distinguer différents cas de figure qui appellent différentes solutions.

1.4.1 : Faits linguistiques complémentaires

Les informations enregistrées dans les dictionnaires concernés sont différentes mais complémentaires. L'union de ces informations suffit à leur intégration.

Format DNT (Dictionnaire Hachette de Notre Temps)

aimer V. tr. dir.

Format Ladl

aimer T7

Format GENELEX

aimer

db : SELF cb

SELF : IntervConst : V

cb : P0 SELF (P1)

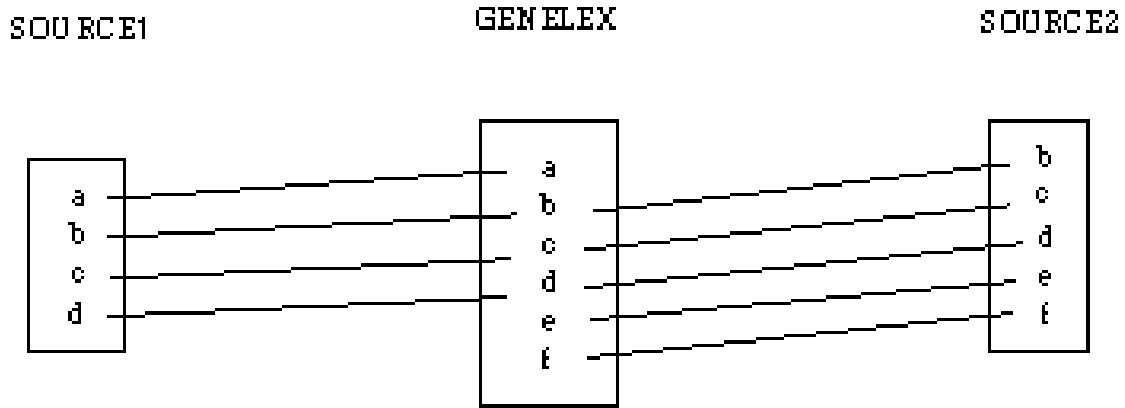
P0 : SN

P1 : SN

P[SsCatSynt : COMPLETIVE]

[Mode : SUBJONCTIF]

Un premier dictionnaire indique pour un verbe qu'il est transitif ou intransitif, et un autre dictionnaire précise les constructions régies par le verbe. Dans ce cas, l'information contenue dans le premier dictionnaire se laisse compléter sans conflit par l'information contenue dans le second. On a donc :

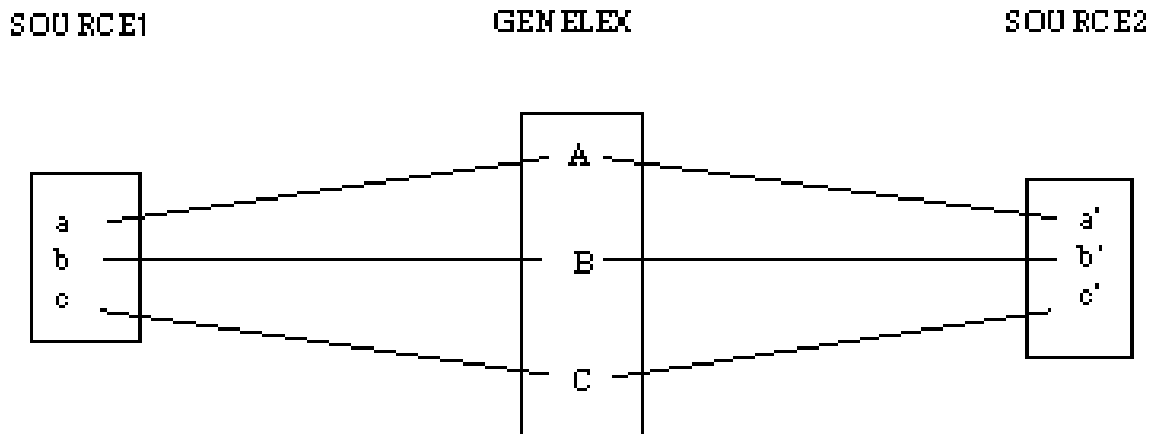


1.4.2 : Faits linguistiques concurrents mais non contradictoires

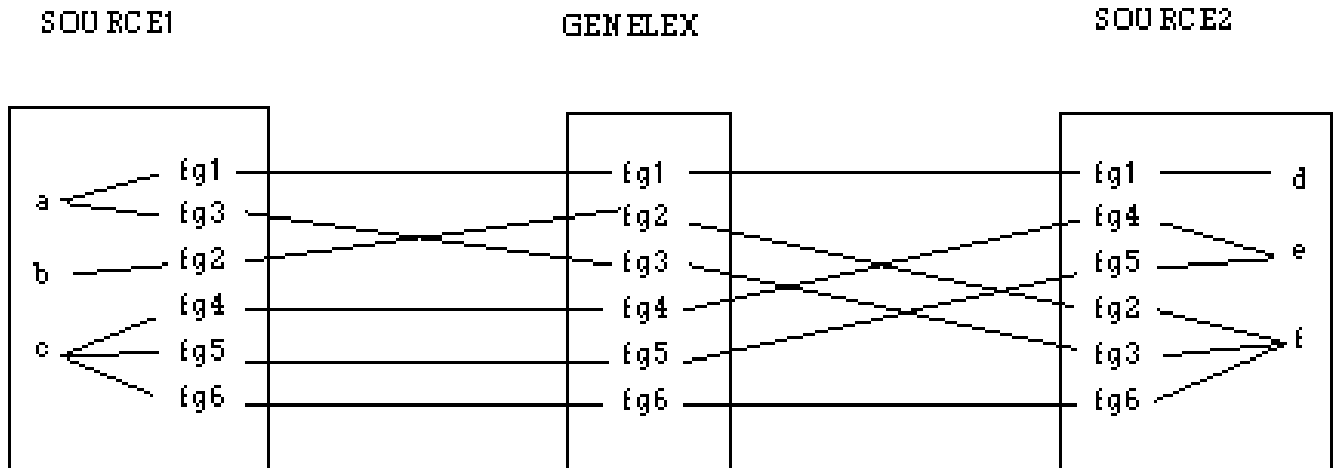
Les dictionnaires traitent des mêmes faits en termes différents. Les différentes terminologies traduisent des divisions théoriques qui tiennent à l'angle d'approche des objets linguistiques, aux types de définitions opératoires autorisées, plus qu'à l'identification des objets eux-mêmes.

Ces divergences peuvent en fait se résorber si l'on parvient :

- a) à définir un métalangage pour décrire l'ensemble des phénomènes, c'est-à-dire trouver un système d'équivalence entre les notions.



- b) à identifier les faits granulaires constitutifs des notions indépendamment de leur regroupement d'origine, c'est-à-dire trouver un système d'équivalence entre les éléments constitutifs des notions.



On traite l'ensemble des informations dont on peut disposer afin d'en extraire les "informations élémentaires" (faits granulaires constitutifs) qui sont mises à plat dans un premier temps puis exprimées dans le format générique.

Format Ladi

aimer

Table 7 : aimer à Vinf

Table 32h : qqun aimer qqun

Table 12, etc...

sujet Nhum +

sujet V1 W +

Compl complétive pc z +

Compl inf V0 W +

Compl Pronom ppv le +

Compl le fait que P +

Compl Nhum +

Compl N-hum +

Format IBM

aimer

(VERB IF TRAN (INFV TR) (COPLOBJ (A P)) AUXA (QUEOBJ SUBJ) INFA
INFD)

Format GENELEX

aimer

db : SELF cb

SELF : IntervConst : V[Aux:AVOIR]

cb : P0 SELF (P1)

P0 : SN

PRO[Lex:quelqu'un]

P1 : SN

P[Introd:le fait que]

[SsCatSynt:COMPLETIVE]

P[Mode:INFINITIF]

P[Prep:à][Mode:INFINITIF]

P[SsCatSynt:COMPLETIVE]

[Mode:SUBJONCTIF]

PRO[Lex:le]

PRO[Lex:quelqu'un]

PRO[Lex:quelque chose]

1.4.3 : Faits linguistiques concurrents et contradictoires

Dans certains cas, en l'absence de solution englobante, les conflits théoriques sont irréductibles, et les informations linguistiques dégagées dans une théorie sont clairement contradictoires avec d'autres. Or, sauf si l'on tolère des incohérences de codage (et quelle serait alors la fiabilité du dictionnaire ?), il est clair que des faits contradictoires ne pourront cohabiter dans une même instance de dictionnaire.

Ainsi, pour un verbe neutre,

Ex : Jean casse la branche

la branche casse

on ne pourra pas dire en même temps que *la branche casse* correspond à la description de base et à une description transformée de la même Usyn "casser".

Ex : (1) Jean casse la branche

->(2) la branche casse

(1) est une des actualisations possibles de la description de base (db) de "casser".

Elle peut avoir pour transformée une actualisation (2) de sa description transformée (dt) :

Dans le cas où l'on considère qu'il s'agit de deux comportements syntaxiques distincts :

(1) Jean casse la branche

est une des actualisations possibles d'une db de "casser".

la branche casse

est une des actualisations possibles d'une autre db de "casser".

De même, pour un verbe de mouvement, on ne pourra pas dire dans un même dictionnaire que tel complément de lieu est à la fois complément essentiel et modifieur.

Ex : A Paris, Jean a marché de la place Vendôme jusqu'à la Défense

Suivant l'option théorique choisie, on traite les compléments de lieu de différentes manières :

- exclusivement comme des compléments essentiels
- exclusivement comme des modifieurs
- suivant les verbes

-soit comme compléments essentiels (par exemple verbes de mouvement)

-soit comme modifieurs (autres verbes).

La fusion de ces dictionnaires sera impossible, sauf à privilégier un point de vue aux dépens des autres. Dans ce cas et pour garantir la généralité, on choisira l'option théorique qui entraîne le moins de perte d'information et qui permette, par calcul, l'extraction d'un dictionnaire applicatif dans une autre théorie. Encore faut-il pouvoir faire dans tous les cas la preuve de l'existence d'une telle théorie.

2 : Homogénéité de codage

Les objectifs de cohérence et donc de (ré)utilisabilité ne pourront être atteints sans que soient précisés les choix faits quant aux descriptions des phénomènes linguistiques : choix non seulement théoriques, mais aussi de codage dans le formalisme proposé par GENELEX. On citera pour exemple la nécessité de préciser globalement - et non au coup par coup - les niveaux de codage de la structuration des syntagmes décrits par des listes de positions ou des transformations (entre Usyn, descriptions, syntagmes occupants de positions).

La stratégie lexicographique devra préciser par ailleurs les critères permettant de déterminer quelles informations il est nécessaire de coder dans le lexique.

Il est également crucial de se tenir à une même stratégie garantissant une cohérence entre les différentes couches (morphologique, syntaxique et sémantique).

Il ressort clairement de ce qui précède qu'il est une étape incontournable et préalable à tout travail lexicographique lors de la mise en œuvre du modèle GENELEX : la définition d'une stratégie de codage.

3 : Articulation Lexique - Grammaire

Lexique et grammaire sont les deux dépositaires des connaissances descriptives de la langue. On attend de ces deux entités qu'elles se complètent dans une description exhaustive (et utilisable pour le traitement automatique en ce qui nous concerne). Cela n'est pas sans répercussions sur l'approche globale nécessaire à la modélisation du lexique.

Le parti pris dans GENELEX est de coder dans un format aussi "neutre" que possible toutes les informations nécessaires, indépendamment des applications envisagées. C'est-à-dire que GENELEX, base de stockage d'informations lexicales, permet d'extraire des dictionnaires correspondant aux choix évoqués ci-dessus, et de se donner pour objectif de pouvoir représenter le plus d'informations possibles.

Rappelons néanmoins la situation classique • l'on dispose d'un lexique et d'une grammaire.

3.1 : Compatibilité du lexique et de la grammaire

3.1.1 : Approche théorique et formalisme

Considérons un point de vue théorique déterminant le découpage en unités linguistiques (constructions, rattachement de syntagmes, délimitations des frontières de syntagmes, etc.). Ce point de vue détermine tout autant la description linguistique faite par la grammaire que les informations syntaxiques contenues dans le lexique.

Lorsqu'on dispose d'un lexique et d'une grammaire, on a donc deux cas de figure :

1. Lexique et grammaire ont été construits suivant la même approche théorique
2. Lexique et grammaire ont été construits suivant des théories différentes, ce qui signifie que certains phénomènes linguistiques sont décrits différemment, ou bien identifiés dans l'un et non dans l'autre.

Indépendamment des choix théoriques devant guider le codage des informations, tant la description du lexique que celle de la grammaire supposent le choix d'un formalisme de description, qui a ses propres unités, son vocabulaire et sa structuration de l'information, et qui bien entendu a une puissance suffisante pour rendre compte de la théorie choisie.

Ici encore, face à un lexique et une grammaire, on a donc deux cas de figure :

1. Grammaire et lexique sont codés dans le même formalisme.
2. Grammaire et lexique sont codés dans des formalismes distincts.

La compatibilité entre le lexique et la grammaire peut donc être considérée sous ces deux angles, comme le montre le tableau suivant.

	<i>FORMALISMES IDENTIQUES</i>	<i>FORMALISMES DIFFERENTS</i>
<i>APPROCHES THEORIQUES IDENTIQUES</i>	<p>CAS 1</p> <p>Situation optimum :</p> <p>La "communication" entre le lexique et la grammaire est immédiate.</p> <p>Un seul souci reste éventuellement: la même finesse de description est souhaitable.</p> <p>NB. Un lexique ayant un degré de finesse de</p>	<p>CAS 2</p> <p>Situation favorable :</p> <p>La "communication" entre le lexique et la grammaire pourra se faire moyennant la conversion dans l'un des formalismes.</p> <p>Cette traduction est toujours possible car le partage de l'approche linguistique garantit la présence des mêmes unités</p>

	description trop grand sera seulement sous-exploité	linguistiques.
<p><i>APPROCHES</i></p> <p><i>THEORIQUES</i></p> <p><i>DIFFERENTES</i></p>	<p>CAS 3</p> <p>Situation passablement favorable :</p> <p>Il faut extraire du lexique un dictionnaire "applicatif" compatible avec la grammaire.</p> <p>La stratégie est la même que celle de la fusion de dictionnaires décrite plus haut.</p> <p>Risque de perte d'information</p> <p>NB. La convergence ou divergence des approches théoriques différentes conditionne la qualité des informations résultantes.</p>	<p>CAS 4</p> <p>Situation difficile :</p> <p>L'existence d'un "pont" entre lexique et grammaire n'est pas garantie.</p> <p>Il faut traduire le lexique dans le formalisme utilisé par la grammaire (si c'est possible sans une perte trop grande de l'information...)</p> <p>On se trouve alors dans le cas 3.</p> <p>Risque de perte grave d'information</p>

Là encore, le choix fait dans GENELEX tend à la "neutralité" la plus accueillante qui ait pu être élaborée. Le formalisme adopté doit être converti dans celui des applications.

N.B. : Au regard de ces problèmes, certaines théories ont donc fait le choix global d'adopter pour la description du lexique et de la grammaire non seulement la même approche théorique mais aussi le même formalisme. Les informations syntaxiques associées au lexique ont alors une granularité fine allant jusqu'à la description des combinaisons possibles des unités lexicales entre elles, comme dans les grammaires lexicales fonctionnelles (LFG).

3.1.2 : Compatibilité minimale requise par le modèle GENELEX

Pour assurer une compatibilité minimale entre le lexique et les grammaires appelées à l'utiliser, GENELEX impose, dans le codage des informations syntaxiques pour le français, le recours aux syntagmes, ce qui fera, nous l'espérons, consensus dans la communauté scientifique. Ces syntagmes seront identifiés par des étiquettes dont une liste minimale (SN, SP, SADV,...) et extensible (sauf pour les étiquettes terminales) est proposée par le modèle de la couche syntaxique. Il s'agit en fait des étiquettes les plus répandues pour désigner les structures syntagmatiques obtenues lors de l'actualisation des schémas de complémentation des unités souvent appelées "majeures".

Cette liste (minimale) d'étiquettes syntagmatiques doit garantir minimalement un "pont" théorique d'une part entre différentes théories mises en œuvre dans le modèle GENELEX, d'autre part avec différentes grammaires. On peut considérer qu'il s'agit d'un noyau dur sur lequel pourra s'appuyer toute théorie servant de base à la définition des critères de codage d'un dictionnaire au modèle GENELEX ou à la construction d'une grammaire appelée à utiliser les informations issues d'un tel dictionnaire.

La contrainte théorique que représente cette liste est contrebalancée par la possibilité de raffiner à souhait la description des syntagmes :

- grâce à la combinatoire des traits restrictifs
 - lexicaux
 - morphologiques
 - morpho-syntaxiques
 - syntaxiques
 - syntaxico-sémantiques ou sémantiques
- grâce à l'expression possible des contraintes de réécriture de

syntagmes

- grâce à la liberté de structuration linéaire ou hiérarchique des syntagmes entre eux.

On peut ainsi exprimer dans les moindres détails les caractéristiques des réalisations autorisées pour les éléments linguistiques intervenant dans la description de l'unité décrite.

Cet ensemble d'outils mis à disposition permet des descriptions à profondeur variable, suivant :

1. Le choix théorique
2. La finesse de description visée
3. L'avancement des travaux

et par là même différents modes d'appropriation du modèle par l'utilisateur.

De plus, l'utilisateur peut enrichir la liste proposée en l'augmentant d'étiquettes syntagmatiques plus précises. Mais il perdra alors en généralité.

3.2 : Renvois du lexique vers la grammaire

Un lexique qui décrit les comportements syntaxiques des entrées lexicales vient compléter les informations syntaxiques générales contenues dans une grammaire.

Ces objets sont nécessairement complémentaires pour obtenir une description satisfaisante de la langue et il existe entre ceux-ci un phénomène de vases communicants :

- plus la grammaire est riche et précise, plus le lexique peut être pauvre et concis.
- plus le lexique est riche, plus la grammaire est réduite.

La tendance actuelle est de donner un rôle croissant à la part du lexique dans la description de la syntaxe, et l'attitude "lexicaliste" poussée à l'extrême consiste d'ailleurs à coder toute l'information syntaxique dans le lexique, laissant à la "grammaire" le seul rôle de dicter les règles des combinaisons licites. On citera pour mémoire les grammaires d'arbres adjoints (TAG) et les grammaires catégorielles.

Cette complémentarité lexique grammaire implique donc que certaines informations peuvent être

enregistrées, au choix, soit dans le lexique, soit dans la grammaire.

Dans le modèle de la couche syntaxique GENELEX, on attend que soient codées dans le lexique les informations syntaxiques propres à une entrée que son appartenance à une catégorie et une sous-catégorie morphologiques ne suffit pas à prédire. On ne se prononce pas sur un niveau de précision requis.

Le principe est que les informations les plus fines doivent pouvoir être enregistrées, quitte à être "écrasées" lors d'une extraction devant générer un dictionnaire pour une application requérant des informations lexicales moins précises.

En particulier, a été implicitement reléguée à la grammaire la description d'un certain nombre de phénomènes linguistiques :

- Les phénomènes de flexions et d'accords
- La concordance des temps
- Le bouleversement de la linéarité qui peut entraîner un écart entre l'ordre canonique enregistré dans le dictionnaire et certaines réalisations de surface. Ce bouleversement peut être lié :
 - au style
 - aux structures emphatiques ou d'extraposition
 - aux structures interrogatives ou négatives
 - aux permutations dues aux phénomènes de SN lourd
 - à l'insertion de modifieurs
 - aux phénomènes de coordination
 - ...
- ...

Les renvois implicites du lexique décrit dans le modèle GENELEX vers la grammaire ne présupposent rien de la façon dont la grammaire rendra compte de ces phénomènes. De ce point de vue, on peut donc dire qu'il y a transparence de la grammaire.

4 : Autonomie de la Syntaxe

Le modèle de la couche syntaxique GENELEX autorise la prise en compte dès la syntaxe des éléments de sémantique qui ont des répercussions au niveau syntaxique et permettent de mettre en évidence des comportements syntaxiques différents. Ces informations sémantiques mènent par conséquent à des éclatements d'unités syntaxiques que la syntaxe seule ne saurait justifier sans s'appuyer implicitement sur la sémantique : ainsi l'identification de deux verbes *voler* ne repose pas uniquement sur l'identification de deux structures formelles.

On voit donc que cette incursion d'informations sémantiques dès la syntaxe permet d'identifier des constructions différentes associées à des sens différents, et, par suite, de désambiguïser sans attendre les informations plus fines du niveau sémantique. La sémantique que nous prenons le parti de pouvoir faire apparaître **explicitement** est en fait le plus souvent utilisée **implicitement** dans les descriptions syntaxiques existantes. Elle se limite aux rôles thématiques et aux conditions dénotationnelles.

Les tenants d'une autonomie totale de la syntaxe par rapport à la sémantique seront bien entendu libres de ne pas utiliser ce type d'informations.

B - ANALYSE DES UNITES SYNTAXIQUES SIMPLES

I : Articulation avec la Couche Morphologique

La couche morphologique du modèle décrit les unités lexicales d'un point de vue morphologique. On y trouve, pour chaque unité morphologique (**Um**), des informations concernant - entre autres éléments de description - la graphie et les variantes graphiques d'une même unité lexicale, sa flexion, sa catégorie grammaticale etc.. La couche morphologique permet donc de rassembler sur une seule unité morphologique les informations que des unités lexicales partagent à ce niveau, indépendamment des comportements syntaxiques ou sémantiques divergents qui pourraient leur être associés.

Pour être complète, la description des unités lexicales exige la représentation d'autres types d'informations concernant le comportement syntaxique particulier de l'entrée. On veut pouvoir enregistrer aussi précisément que nécessaire les caractéristiques singulières qui la distinguent du comportement général associé à sa catégorie grammaticale enregistrée en morphologie. Ainsi, on sait que les verbes ont un sujet et 0 à n compléments. Il faudra spécifier le nombre de ces compléments ainsi que leur nature (cf. règles de sous-catégorisations dans les HPSG). C'est à l'enregistrement de ce type d'informations qu'est dédiée la couche syntaxique de GENELEX.

Si une unité morphologique (**Um**) a une et une seule catégorie syntaxique au sens de la couche morphologique (catégorie grammaticale), elle peut avoir **un** ou **plusieurs** "comportements syntaxiques".

Ex : L'homme arrive à Paris. verbe de mouvement

L'homme arrive à comprendre. verbe modal

L'homme vole une pomme. transitif

L'oiseau vole rapidement. intransitif

Les unités syntaxiques (**Usyn**) servent à décrire ces comportements syntaxiques (attestés, autant que possible), qu'on appelle aussi couramment tournures ou emplois, et cela quelle que soit la catégorie de l'entrée.

Le modèle présenté couvre les mots simples et les unités lexicales composées. De plus, la formalisation des mots simples s'applique évidemment à la représentation de la syntaxe externe des composés.

II : Les Unités Syntaxiques (Usyn)

1 : Définition générale

L'unité syntaxique est le point d'entrée de la couche syntaxique.

Elle peut être simple ou composée. Nous nous intéressons ici plus particulièrement aux unités syntaxiques simples, les unités composées faisant l'objet de la partie C de ce document.

Une unité syntaxique (**Usyn**) décrit un - et un seul - des comportements syntaxiques d'une unité morphologique (**Um**).

Chaque unité morphologique est donc en relation avec au moins une unité syntaxique. Si une Um a plusieurs "comportements syntaxiques", alors elle est en relation avec plusieurs Usyn. Dans le modèle entité-relation, ce point est exprimé par les cardinalités. Il n'est donc pas possible de dresser une liste finie des unités syntaxiques. Elles dépendent en effet directement des Um instanciées dans le dictionnaire.

Une Usyn simple correspond à une et une seule Um.

Si l'entrée est de catégorie majeure, alors l'Usyn décrit minimalement son schéma de complémentation (complémentation du Verbe, du Nom, de l'Adjectif, de l'Adverbe). L'Usyn d'une entrée peut alors être vue comme une abstraction sur le syntagme dont l'entrée est la tête. On entend par là que l'Usyn contient à un niveau "méta" toutes les informations qui permettent de décrire son comportement dans une actualisation de surface.

L'unité syntaxique permet de représenter le comportement syntaxique des unités lexicales suivant

deux points de vue très différents:

- le point de vue "**atomiste**", focalisé sur les propriétés de l'unité lexicale indépendamment de la grammaire choisie, s'attache à décrire les éléments se trouvant dans la dépendance directe de l'unité décrite. Il s'agit d'informations concernant la syntaxe profonde, la syntaxe de surface devant être gérée par la grammaire ;
- le point de vue "**syntaxiste**", qu'on trouve dans les approches dites "lexicalistes" où l'on tend à enregistrer dans le lexique tout ou partie de la grammaire (grammaires catégorielles, TAG's, etc.) par la combinaison de syntagmes, autorise au niveau de l'Usyn décrite la référence à des structures syntaxiques dans lesquelles elle s'inscrit. Le lexicographe devra définir sa stratégie par rapport à la grammaire, pour permettre un codage homogène et cohérent du lexique.

Par souci d'homogénéité et pour permettre le contrôle de cohérence de la base lexicographique, il est recommandé de ne pas faire coexister les deux lectures du modèle dans une même instance de dictionnaire.

2 : Point de vue atomiste

La vision atomiste privilégie le point de vue de la saturation de l'unité décrite : on estime que l'unité émet des appels à saturation, c'est-à-dire qu'elle est porteuse d'un certain nombre de "trous", **les positions à saturer**, et qu'elle exprime des restrictions sur les occupants possibles sans rien dire des réalisations syntagmatiques de surface, l'organisation des syntagmes étant du ressort de la grammaire.

Les positions sont donc les informations émises par l'unité sur l'extérieur. Les positions seront occupées (actualisées) par les réalisations des Usyn appelées. Le comportement d'appelé est le fonctionnement mineur des unités qui restent par ailleurs libres d'avoir un comportement d'appelant (c'est-à-dire qu'ayant été appelées, elles peuvent à leur tour émettre des appels). Si l'unité appelante impose des restrictions aux unités qu'elle appelle, on enregistre ces restrictions lorsqu'on décrit les réalisations de ses positions : dans le formalisme, les occupants de positions sont des syntagmes que l'on peut restreindre par des traits. Dans la vision atomiste, les syntagmes tels que définis par le formalisme sont des étiquettes permettant de nommer des actualisations autorisées pour les Usyn, ils ne sont pas réécrits, alors qu'ils le seront dans la vision syntaxiste.

*Ex : un petit **groupe** de personnes*

cb : P0 (P1) (P2)

P0 : DET

P1 : SADJ

P2 : SP[Prep:de][Nombre:PLURIEL]

[SsCatSynt:DET_VIDE]

On voit sur cet exemple comment la construction de base d'une unité syntaxique décrit un fonctionnement majeur, fonctionnement de l'unité en tant qu'appelant. Les comportements majeurs des entrées seront les seuls décrits par une construction dans cette lecture du modèle. On voit aussi que le point d'insertion de l'entrée, SELF, n'est pas noté : la cb n'est pas comprise comme description d'un syntagme mais comme somme d'informations émise par l'unité décrite.

Dans la lecture atomiste du modèle, on ne dépasse donc pas, dans la construction, le niveau du syntagme dont l'entrée serait la tête. Les relations qu'une unité syntaxique entretient avec les autres unités syntaxiques, dans un cadre qui dépasse les unités qu'elle "gouverne" en sa qualité d'appelant, peuvent être précisées dans Self par un jeu de sous-catégorisations, descriptions par traits et précisions sur ses fonctions possibles. Ainsi pour les adjectifs, si l'on veut spécifier les possibilités d'emploi en attribut du sujet ou de l'objet, en épithète droit ou gauche, on donnera les fonctions suivantes :

EPITHETE_GAUCHE, EPITHETE_DROIT, ATTRIBUT_SUJET, ATTRIBUT_OBJET.

Dans les exemples de ce document, la ligne [cb : P0 SELF (P1) . . .] est utilisée pour faire apparaître de façon synthétique les informations susceptibles d'être portées par la construction et concernant son actualisation : optionnalité, contraintes sur l'organisation de surface. Le point d'insertion de Self y figurera donc systématiquement. Cette information ne sera cependant jamais pertinente pour la lecture atomiste du modèle ; cette convention a été adoptée par souci d'homogénéité de présentation.

3 : Point de vue syntaxiste

La vision syntaxiste privilégie le point de vue de l'inscription de l'unité dans un contexte d'apparition ; on fait donc apparaître la position occupée par Self (l'unité décrite) dans la construction de base, cette construction étant l'expression de la structure syntaxique dans laquelle s'insère l'unité décrite. L'unité syntaxique est donc présentée comme s'insérant dans un arbre syntaxique. Cet arbre représente un contexte syntaxique plus ou moins large suivant les spécifications et contraintes à émettre.

Les Usyn ne décrivent pas exclusivement des comportements majeurs : Self peut se trouver dans une position dont la fonction est "TETE" (comportement majeur) ou non (comportement mineur). Pour une entrée de catégorie mineure, l'Usyn décrit un contexte d'occurrence.

Dans la lecture syntaxiste du modèle, il est donc possible, pour décrire un contexte d'occurrence, de dépasser le cadre du syntagme dont l'entrée est la tête (SN pour N, SADJ pour ADJ, P ou SV pour V).

Ainsi pour les adjectifs, si l'on veut spécifier les possibilités d'emplois en attribut du sujet ou de l'objet, en épithète droit ou gauche, on pourra élargir le cadre de la description au SN, voire à la phrase, ce qui permettra alors d'émettre des contraintes sur le sujet depuis un adjectif attribut.

*Ex : Il est **intéressant** de remarquer cela.*

La description d'un syntagme par réécriture est exprimée au travers d'une liste de positions, une de ces positions pouvant avoir la fonction "TETE". Les positions décrivant la réécriture d'un syntagme peuvent elles-mêmes être occupées par un syntagme réécrit (récursivité) de manière à introduire une structure arborescente.

*Ex : un petit **groupe** de personnes*

*le **groupe** des personnes qui...*

cb : P0 P1* SELF (P2)

P0 : DET

P1 : SADJ

P2 : **SP**

SP : P0 P1

P0 : PREP[Lex:de]

P1 : **SN**

SN : P0 P1* P2 P3*

P0 : DET[SsCatMorph:DEFINI]

DET[SsCatMorph:DEMONSTRATIF]

DET[SsCatSynt:DET_VIDE]

P1 : SADJ

P2 : NOM[Nombre:PLURIEL]

P3 : SADJ

SP

P[SsCatSynt:RELATIVE]

4 : Critères d'éclatement des Usyn

On retient comme critère formel d'éclatement des unités syntaxiques le fait qu'une unité syntaxique

a une et exactement une description de base.

Les critères d'ordre linguistique et lexicographique menant à l'élaboration des descriptions syntaxiques sont du ressort du lexicographe. Nous donnons cependant ici certaines recommandations permettant de faire un bon usage du modèle.

Il s'agit en fait dans un premier temps d'adopter une description qui permettra d'avoir une bonne couverture des comportements syntaxiques en cherchant à éviter au mieux tant la redondance que les regroupements trop puissants. Nous illustrerons ce point par les exemples suivants :

Ex : (1) Pierre arrive à partir

(2) a. Pierre arrive à Paris

(2) b. Pierre arrive

(3) a. tarte aux abricots / à la banane

(3) b. tarte

- éviter au mieux la redondance (signe de modélisation conceptuelle peu satisfaisante, ce qui relève d'un même concept n'ayant pas été regroupé correctement) qui entraîne la création d'une multiplicité d'unités syntaxiques inutiles.

Ex : on choisit de regrouper :

- (2) a. et (2) b., pour *arriver* verbe de mouvement
- (3) a. et (3) b. pour *tarte* et *tarte à...*

- ne pas pécher par l'excès inverse, un modèle trop puissant ne permettant pas de gérer efficacement les restrictions sur les réalisations propres à une unité.

Ex : on choisit de dissocier les deux *arriver à* :

- l'emploi modal, soit (1)
- l'emploi de mouvement, soit (2).

Pour schématiser, il s'agit donc de déterminer quel type de représentation de l'information syntaxique on devra retenir pour éviter tant les écueils d'une description du type "ce verbe peut accepter un complément" que ceux d'une description qui mènerait à un recensement le plus exhaustif possible des constructions rencontrées (voire même simplement imaginables), dans la mesure où certaines observations locales peuvent être formulées par des règles syntaxiques. A l'extrême, on pourrait en effet envisager d'associer à une unité une construction précise (construction étant ici pris au sens de "phrase") rencontrée par exemple dans un corpus. La description de cette construction pourrait être en soi une construction qu'on ne pourrait rattacher à un comportement syntaxique décrit. Cette construction impliquerait alors la création d'une Usyn distincte.

III : Descriptions

Une Usyn est définie par une **Description de base** et 0 à n **Descriptions transformées**.

Une Description est définie par Self et une Construction.

On peut ainsi partager les constructions indépendamment des contraintes sur Self : par exemple, certains verbes partagent la même Construction, alors qu'ils diffèrent par l'auxiliaire et/ou la particule préverbale :

Ex : partir

s'en aller

Une description n'est pas intrinsèquement de base ou transformée, mais remplit ce rôle dans le cadre d'une Usyn donnée. Une même description peut d'ailleurs jouer le rôle de description de base pour une Usyn et de description transformée pour une autre.

IV : Constructions

Une **Construction** décrit le contexte syntaxique appelé et/ou contraint par l'entrée décrite. Disons en d'autres termes et pour le verbe, qu'une construction décrit ce qu'on appelle couramment "schéma de complémentation".

Une construction est définie par une liste canoniquement ordonnée de positions, par leur optionnalité, leur solidarité et leur organisation de surface (décrits en section VI-4). Elle peut par ailleurs porter une étiquette syntagmatique non terminale et des traits restrictifs.

Pour reprendre notre distinction entre les deux lectures du modèle :

- Dans la vision **atomiste**, la construction n'est que la structure regroupant les descripteurs de schémas de complémentation. A ce titre, elle n'est pas comprise comme un syntagme et aucune étiquette syntagmatique ne lui est associée. De plus, la construction ne décrira que les comportements majeurs.
- Dans la vision **syntaxiste**, une construction décrit implicitement un syntagme, c'est pourquoi on peut lui associer une étiquette syntagmatique. Le contexte décrit pourra se situer suivant les cas au niveau de la phrase, du syntagme nominal ou prépositionnel, etc... Par ailleurs, la construction décrivant un contexte d'occurrence, elle ne décrit pas exclusivement le comportement majeur : elle peut aussi rendre compte de comportements mineurs d'unités décrites comme majeures (comme les adjectifs) ou des comportements d'unités de catégorie mineure.

V : Self

1 : Définition générale

L'élément **Self** permet de décrire les propriétés de l'entrée pour une Usyn

- dans le cadre de sa construction
- dans un cadre extérieur à sa construction.

Pour les unités composées, Self permet de plus de décrire la structure interne du composé.

2 : Self intervenant dans sa construction

Etant donnée une construction syntaxique, Self permet d'exprimer l'ensemble des caractéristiques de l'entrée pour cette construction, en tant qu'appelant ou appelé :

- **catégorie de l'entrée** : celle-ci n'est pas nécessairement égale à sa catégorie morphologique (catégorie grammaticale de l'Um), ce qui permet de noter un écart entre catégorie morphologique et catégorie fonctionnelle (catégorie notée par l'étiquette syntagmatique de l'Usyn). On se donne ainsi les moyens de décrire par exemple le fonctionnement adjectival du nom "abricot"

Ex : abricot

Self : IntervConst : ADJ

- **auxiliaire de conjugaison** si c'est un verbe

Ex : tomber

Self : IntervConst : V[Aux:ETRE]

- **restrictions morphologiques** :

Ex : lustres

Self : IntervConst : N[Nombre:PLURIEL]

- **particule préverbale** pour les emplois de "vrais pronominaux"

Ex : s'en aller

Self : IntervConst : V[Pronominal:SE_EN]

Ces couples (catégorie ; traits restrictifs) correspondent formellement à des syntagmes au sens où ceux-ci sont définis dans le formalisme du modèle. Les caractéristiques de Self en tant qu'Intervenant de Construction seront donc exprimées par une liste de syntagmes terminaux, portée par un élément que l'on appellera **IntervConst**. On peut ainsi rassembler dans une même description des emplois de verbes ayant la même construction pour un Self variable. On en trouve quelques exemples tels que :

Ex : Nous (sommes / avons) convenus d'un rendez-vous

(auxiliaire ETRE ou AVOIR)

Self : IntervConst : V[Aux:ETRE]

V[Aux:AVOIR]

Ex : (se) moquer

Self : IntervConst : V[Pronominal:SE]

V

• **fonction et rôles thématiques** : ceux-ci ont un sens dans la vision syntaxiste où les constructions sont assimilées à des syntagmes où Self fonctionne :

- soit uniquement comme tête lorsqu'il est appelant (comportement de catégorie "majeure")
- soit comme élément de réécriture de niveau n recevant une fonction lorsqu'il est appelé (comportement de catégorie "mineure"), ce qui n'exclut naturellement pas la description du syntagme dont il est la tête.

Ex : Il est intéressant de remarquer cela

cb : P0 P1 P2

P0 : PRO[Lex:il][SsCatMorph:IMPERSONNEL]

P1 : V[SsCatSynt:COPULE]

P2[Fonction:ATTRIBUT_SUJET] : **SADJ**

SADJ : (P0) SELF P1

Self : IntervConst[Fonction:TETE] : ADJ

P0 : SADV

P1 : SP[Prep:de]

3 : Self appelé par un élément externe à sa construction

Par ailleurs on peut noter le(s) comportement(s) de Self en tant qu'appelé par un élément qui n'est pas décrit dans sa construction.

Dans la vision **atomiste**, on ne décrit (par une construction) que les comportements d'appelant, on peut ainsi préciser que telle construction décrite pour un adjectif se rapporte à son emploi épithète ou attribut. Ces Comportements en tant qu'Appelé sont décrits par une liste de **ComportAppele**, éléments équivalents à des syntagmes, constitués d'un couple (catégorie ; traits restrictifs) auquel s'ajoutent éventuellement une fonction et un ou plusieurs rôles thématiques.

Dans une vision **syntaxiste** au contraire, tous les comportements d'appelé peuvent être entièrement décrits par la construction. On se réserve cependant le droit de les décrire autrement, notamment lorsque la lourdeur d'une construction à déployer semble injustifiée. On pourra notamment décrire des comportements de Self en tant qu'épithète ou attribut par recours à ComportAppele, si l'on ne souhaite pas préciser ces contextes plus avant.

VI : Positions

1 : Approche générale et justification

1.1 : L'exemple de la complémentation verbale

Pour présenter les éléments qui ont présidé à l'élaboration de la notion de **Position**, nous nous baserons sur la catégorie la mieux décrite à ce jour en syntaxe : le verbe. On notera que la description des schémas de complémentation verbale se fait cependant en des termes très différents selon les théories. On citera à titre d'exemples :

- sujet, objet direct, indirect, attribut, etc. : tradition, fonctionnalisme, LFG
- N0 (paradigme nominal), N1, Prep N2 : distributionnalisme et grammaire transformationnelle
- A0 (actant), A1, A2 : Tesnière
- Arg0 (argument), Arg1, Arg2 : logique prédicative, Montague
- Rôle thématique : grammaire générative, Jackendoff et Fillmore

1.2 : Emergence de la notion de Position

Nous concevons la position comme une vision synthétique de l'ensemble des notions rappelées ci-dessus, une abstraction qu'on peut formaliser en s'attachant à restructurer l'information élémentaire indépendamment de la façon dont elle est organisée dans les notions d'origine.

Pour un verbe, une position est un paradigme qui regroupe les différentes réalisations syntaxiques de sujet ou de complément de verbe et qui participe à la définition maximale de la valence verbale. (GENELEX Juin 91)

J.-C. Milner [1989] introduit explicitement la notion de position, dans un système plus complexe qui comprend aussi des sites et des places.

Il n'y a pas recouvrement total entre notre définition de la position et la notion dégagée par Milner :

nous maintenons la distinction entre position et occupants de position. Nous pensons comme lui que

"L'appartenance catégorielle d'un terme X et l'étiquette catégorielle de la position Y occupée par X [sont] indépendantes en droit" [Milner, 89, p.370].

Il nous semble donc justifié d'étiqueter différemment les deux types de "catégories" alors que les positions portent chez Milner l'étiquette de leur réalisation syntaxique typique. Il nous semble que les traiter en termes d'homonymie ne peut être qu'un facteur de confusion entre nature et fonction.

Dans le format GENELEX, les positions sont définies (entre autres) par l'ensemble des syntagmes (étiquettes syntagmatiques et/ou syntagmes réécrits) qui peuvent les instancier : il n'est pas nécessaire de dégager une "vedette" (nécessaire pour étiqueter la position chez Milner) parmi cet ensemble de syntagmes.

De fait, les paradigmes syntaxiques ne se prêtent pas tous aussi bien à l'extraction d'une "vedette". Certes, on s'accorde à considérer le Syntagme Nominal (SN) comme la réalisation typique de la position à fonction sujet, bien qu'on trouve divers syntagmes en position à fonction sujet :

SN

P [SsCatSynt : COMPLETIVE]

P [Mode : INFINITIF]

...

Ex : Cette décision regarde Marie.

Qu'il prenne cette décision regarde Marie.

Prendre cette décision regarde Marie.

...

Sur la position à fonction objet, il est en revanche parfois impossible (ou alors arbitraire) de trancher.

C'est le cas des verbes de discours où phrases complétives, infinitives et SN ou SP sont dans le même paradigme.

Ex : Jean pense que Marie est partie

Jean pense à écrire

Jean pense écrire à Marie

Jean pense au départ de Marie

Jean dit que Marie est partie

*Jean **dit** partir avec Marie*

*Jean **dit** de partir à Marie*

*Jean **dit** des choses à Marie*

*Jean **recommande** d'être sage à Marie*

*Jean **recommande** la sagesse à Marie*

1.3 : Cas de la détermination gauche

Cette détermination gauche renvoie par exemple pour les noms aux divers déterminants, pour les verbes à une position à fonction "sujet". Cette position à fonction déterminant ou sujet pourra être mise dans la dépendance de l'unité décrite, nom ou verbe. Nous l'avons vu, le sujet peut avoir différentes réalisations en français.

Or certains verbes excluent une ou plusieurs de ces réalisations.

*Ex : Pierre **opine** de la tête*

P0 : SN

PRO[Lex:il][SsCatMorph:PERSONNEL_FORT]

Certains verbes encore, restreignent une ou plusieurs de ces réalisations.

*Ex : Il **pleut***

(verbe météorologique)

*Il **semble** qu'il neige*

(verbe employé comme modalisateur d'énoncé)

P0 : PRO[Lex:il][SsCatMorph:IMPERSONNEL]

*Ex : Les fourmis **grouillent** dans le jardin*

(un des emplois d'un verbe symétrique, l'autre étant *le jardin grouille de fourmis*)

P0 : SN[Nombre:PLURIEL]

Mettre le sujet dans la dépendance du verbe permet d'exprimer ces exclusions ou restrictions sur le sujet. On rend compte ainsi des phénomènes de sélection lexicale.

Pour les mêmes raisons, nous mettons d'autres déterminants dans la dépendance du déterminé.

Ainsi, certains substantifs ne supporteront que des déterminants définis.

*Ex : la **plupart***

*le **paléolithique***

2 : Application aux autres catégories

Penser le schéma de complémentation ou la description d'un contexte en termes de positions permet d'étendre le principe à toutes les autres catégories.

Rappel : la ligne [cb : P0 P1 . . .] est utilisée pour faire apparaître de façon synthétique les informations susceptibles d'être portées par la construction et concernant son actualisation : optionnalité, contraintes sur l'organisation de surface. La présence systématique de Self, adoptée par souci d'homogénéité de présentation, n'est pas pertinente pour la lecture atomiste du modèle.

2.1 : Les Noms

On peut ainsi décrire les substantifs. Dans la littérature, le problème de la complémentation des substantifs se cantonne le plus souvent à l'étude des nominalisations déverbiales. On retrouve alors sensiblement les mêmes positions que pour le verbe d'origine :

*Ex : la terrible **destruction** de la ville par les ennemis*

cb : P0 (P1) SELF (P2) (P3)

P0 : DET

P1 : SADJ

P2[Fonction:GENITIF] : SP[Prep:de]

P3[RoleTh:AGENT] : SP[Prep:par]

Mais ce type de description convient aussi aux autres substantifs. On peut par exemple caractériser les noms quantifieurs :

*Ex : un petit **sac** de clous*

*un **sac** de farine*

cb : P0 (P1) SELF (P2)

P0 : DET

P1 : SADJ

P2 : SP[Prep:de]

Certains déterminants nominaux sélectionnent un déterminant défini.

*Ex : la **plupart** du temps*

cb : P0 SELF P1

P0 : DET[SsCatMorph:DEFINI]

P1 : SP[Prep:de]

D'autres sélectionnent un complément pluriel.

*Ex : une énorme **meute** de loups*

cb : P0 (P1) SELF (P2)

P0 : DET

P1 : SADJ

P2 : SP[Prep:de][Nombre:PLURIEL]

Les réalisations des positions permettent donc de sous-catégoriser les substantifs à différents degrés de finesse.

2.2 : Les Adjectifs

Le même système peut être mis en œuvre pour les adjectifs. On opposera tout d'abord ceux qui régissent un complément aux autres.

*Ex : très **capricieux***

cb : (P0) SELF

P0 : SADV

*tout à fait **apte** au travail*

cb : (P0) SELF (P1)

P0 : SADV

P1 : SP[Prep:à]

Les restrictions sur les syntagmes occupant une position (voir plus loin paragraphe sur ce point) permettent de faire des sous-catégorisations encore plus fines.

Ex : content de venir

content de ses résultats

cb : (P0) SELF (P1)

P0 : SADV

P1 : P[Prep:de][Mode:INFINITIF]

SP[Prep:de]

difficiles à satisfaire

cb : (P0) SELF (P1)

P0 : SADV

P1 : P[Prep:à][Mode:INFINITIF]

2.3 : Les Prépositions

Les prépositions pourront aussi être définies sur ce modèle.

Ex : à Paris

à ma mère

à boutons

à faire des bêtises

cb : SELF P0

P0 : SN

P[Mode:INFINITIF]

2.4 : Les Déterminants

La construction donnée ici représente un contexte d'occurrence d'une catégorie mineure dans la vision syntaxiste du modèle.

Ex : certaines petites bêtises qui sont pardonnables

cb : SELF P0* P1 P2*

P0 : SADJ

P1 : N

P2 : SADJ

P[SsCatSynt : RELATIVE]

SP

On remarquera que, dans cet exemple, l'entrée n'est pas la tête du syntagme o• elle s'inscrit, car il s'agit d'une catégorie mineure. Il en va de même pour les conjonctions.

2.5 : Les Adverbes

La portée de l'adverbe peut être exprimée par des positions au travers de différentes unités syntaxiques.

Adverbe portant sur un adjectif :

Ex : très grand

cb : SELF P0

P0 : SADJ

Adverbe portant sur un adverbe :

Ex : très poliment

cb : SELF P0

P0 : SADV

3 : Définition de la Position

Rappelons que sur la base des positions, une même unité lexicale peut avoir un ou plusieurs

comportements syntaxiques, qui donnent lieu à une ou plusieurs Usyn.

Ex : 1. L'homme vole une pomme (transitif)

cb : P0 SELF (P1)

2. L'oiseau vole. (intransitif)

cb : P0 SELF

Ex : 1. Il arrive à Paris

cb : P0 SELF (P1)

P1 : SP[SsCatSynt:LIEU]

P1 est définie par un SP introduit par un sous-ensemble de prépositions appartenant à l'ensemble des "prépositions locatives" (de plus elle est facultative).

2. Il arrive à partir (modal)

cb : P0 SELF P1

P1 : SP[Prep:à]

P[Prep:à][Mode:INFINITIF]

P1 est définie par un SP régi introduit par la préposition à (de plus elle est obligatoire).

3.1 : Définition formelle

Sur le plan opératoire, la position est un élément entrant dans la définition d'une construction.

Une Position est définie par un triplet : distribution ; fonction ; rôles thématiques.

Les positions sont notées P, o• P est l'abréviation de Position. Pour les distinguer commodément, on les note ici Pi, i étant un entier naturel de 0 à n qui indique le rang de la position parmi les autres dans l'ordre canonique établi pour une construction.

Cette notation n'est utilisée que dans le texte d'optionnalité et de solidarité des constructions. Les autres éléments du modèle manipulant les positions les identifieront généralement par leur seul rang i.

Une **Distribution** sur une position est l'ensemble des Syntagmes qui peuvent instancier cette position. Distribution est donc à prendre au sens linguistique de paradigme de substitution.

Une position a 0 ou une **Fonction**, 0, un ou plusieurs Rôles Thématiques (que nous appellerons **RoleTh**).

- **Renseignement de la Distribution :**

la distribution jouant un rôle essentiel dans la définition de la position, elle sera toujours renseignée. Dans le cas où l'on veut indiquer la présence d'une "position vide", c'est-à-dire non réalisée en surface, on a recours à la valeur "e" (élément vide) du Syntagme. La catégorie "e" permet de noter les traces pour les tenants de la grammaire générative et de les considérer comme des syntagmes fantômes (servant notamment à représenter les sujets des infinitives supportant des contraintes de coréférence) auxquels on peut associer les restrictions nécessaires.

- **Renseignement de la Fonction**

la fonction peut ne pas être renseignée pour plusieurs raisons :

- aucune appellation particulière en usage au niveau fonctionnel

Ex : Un mammifère qui allaite ses petits

Un fumeur qui ne supporte pas la fumée

si l'on veut distinguer relative appositive et relative restrictive

- absence d'information liée à une incomplétude volontaire ou involontaire.

- **Renseignement du Rôle Thématique**

le rôle thématique peut ne pas être renseigné pour plusieurs raisons :

- aucune appellation particulière en usage
- syntaxe autonome
- incomplétude volontaire ou involontaire

NB : les valeurs proposées par le modèle pouvant paraître insuffisantes à l'utilisateur, aussi bien pour la fonction que pour le rôle thématique, il lui est donné la possibilité d'y ajouter ses propres valeurs.

3.2 : Positions et Distributions

3.2.1 : Occupants de Position

La liste des syntagmes pouvant occuper une position constitue la distribution sur cette position.

Dans la vision **atomiste**, les syntagmes ne sont pas réécrits, sauf cas exceptionnel où le jeu conjugué des étiquettes syntagmatiques et des traits restrictifs ne suffirait pas à exprimer toutes les contraintes.

Dans la vision **syntaxiste**, les syntagmes sont réécrits ou non, suivant les besoins.

Il peut de plus exister des liens de transformation entre les syntagmes occupant une même Position (appelés TransfSyntagme).

3.2.2 : Partage des Positions

Les positions peuvent être partagées par différentes entrées, indépendamment de leur rang dans une construction.

On peut envisager de proposer, dans un poste lexicographique, une liste prédéfinie de positions, liste pouvant être augmentée à volonté.

Ainsi, sur la base des travaux du Ladl, on pourrait proposer pour les différentes positions verbales des listes du type :

Réalisations de **P [SUJET]** :

e (position non réalisée en surface)

SN

PRO

P [Mode : INFINITIF]

P [SsCatSynt : COMPLETIVE] [Mode : SUBJONCTIF]

etc.

Réalisations de **P [OBJET]** :

SN

PRO

PRO [SsCatMorph : PERSONNEL_FAIBLE]

SP

SADJ

SADV

P[Mode:INFINITIF]

P[Prep:à][Mode:INFINITIF]

P[Prep:de][Mode:INFINITIF]

P[SsCatSynt:COMPLETIVE][Mode:INDICATIF]

P[SsCatSynt:COMPLETIVE][Mode:SUBJONCTIF]

P[Prep:à][SsCatSynt:COMPLETIVE]

P[Prep:de][SsCatSynt:COMPLETIVE]

P[SsCatSynt:INTERROGATIVE_DRI]

P[SsCatSynt:INTERROGATIVE_DRD]

etc.

3.3 : Positions et Fonctions

On peut associer une fonction à toute position : la fonction est une propriété définitoire de la position.

Les constructions sont donc aussi différenciables par les fonctions associées à leurs positions.

Ex : Jean aime Marie

P0 SELF P1

P0[Fonction:SUJET]

P1[Fonction:OBJET_DIRECT]

Les fonctions associées aux positions se définissent par rapport à la tête, qui peut être :

- l'unité appelante, c'est-à-dire Self ayant fonction de TETE, si la construction décrit le comportement majeur de cette unité ;
- une Position de même niveau à fonction TETE si la construction décrit le comportement mineur d'une unité.

On notera que la fonction se définit pour une position, sans que la nature des syntagmes occupants

de cette position soient forcément déterminante : en effet, SN et SP peuvent très bien cohabiter dans une même position. C'est le cas de verbes pour lesquels on trouve une position à fonction 'locatif' comme complément essentiel du verbe :

Ex : Jean fouille ses poches cf. [Boons Guillet Leclère, 1976]

Jean fouille dans ses poches

GENELEX propose une liste de base des fonctions ; elle peut être modifiée ou étendue par l'utilisateur.

3.4 : Positions et Rôles Thématiques

Sur l'utilité d'utiliser des rôles thématiques pour définir des positions, nous ferons une démonstration par l'absurde.

Ex : (1) Jacques traîne la valise. cf. [G.Bès, 1991]

(2) Jacques traîne.

(3) La conversation traîne.

Premier cas de figure :

On se donne la notion de paradigme à l'intérieur duquel différents syntagmes peuvent alterner.

On se donne l'optionnalité comme moyen technique de regroupement de structures de surface.

Pour le verbe traîner, on obtient donc la structure : P0 SELF (P1). On ne rend donc pas compte du fait que l'emploi transitif sélectionne l'interprétation "tirer" alors que l'emploi intransitif sélectionne les sens "flâner" et "s'éterniser".

Le dictionnaire ne lève donc pas les ambiguïtés qu'il pourrait lever dès le niveau syntaxique : un analyseur syntaxique utilisant ce dictionnaire ne pourra sélectionner le sens "tirer" devant l'énoncé (1). Les constructions (2) et (3) restent ambiguës, la désambiguïsation se faisant au moment du passage au niveau sémantique (choix de la "bonne" unité sémantique parmi celles associées à l'unité syntaxique unique).

Deuxième cas de figure :

Pour une syntaxe opératoire en liaison avec la sémantique, il faut donc se donner les moyens de distinguer le "traîner" transitif du "traîner" intransitif, c'est-à-dire donner à l'optionnalité un statut linguistique et non plus un statut technique.

Or les seules fonctions ne suffisent pas à justifier la séparation en P0 P1 pour (1), P0 pour (2) et

(3). Ce qui a permis cette distinction, ce sont implicitement les rôles thématiques :

(1) cb : P0 SELF P1

P0[RoleTh:AGENT]

P1[RoleTh:THEME]

(2)et(3) cb : P0 SELF

P0[RoleTh:THEME]

NB : Les interprétations (2) et (3) seront distinguées dans la couche sémantique.

GENELEX permet à ceux qui voudront faire ce découpage d'expliciter l'implicite en spécifiant les rôles thématiques sur les positions dès la syntaxe.

4 : Positions et Constructions

Rappelons qu'une Construction est définie par une liste ordonnée de 1 à n positions, leur optionnalité, et leur solidarité.

On peut par ailleurs vouloir exprimer le point d'insertion de SELF dans la construction (ou le syntagme dans lequel SELF appara"t). On a pour ce faire recours à l'attribut *inseself* qui prend pour valeur *i*. La signification en est que SELF s'insère avant la position *Pi*. Dans le cas o• SELF vient après toutes les positions de la construction, la valeur de *i* sera la valeur de la dernière position + 1. Pour ne pas noter le point d'insertion, on ne renseignera pas l'attribut.

4.1 : Optionnalité

4.1.1 : Optionnalités simples

L'actualisation des positions peut être obligatoire ou facultative, l'**Optionnalité** étant exprimée par des parenthèses.

Ex : Jean s'adresse à ses invités

cb : P0 SELF P1 => P1 obligatoire

Il mange du pain

Il mange

cb : P0 SELF (P1) => P1 optionnel

Lorsque plusieurs positions sont optionnelles, on a donc par combinatoire un ensemble de réalisations de surface possibles.

*Ex : Pierre **parle** de sa soirée à Marie*

*Pierre **parle** de sa soirée*

*Pierre **parle** à Marie*

*Pierre **parle***

P0 (P1) (P2)

-> P0 P1 P2

-> P0 P1

-> P0 P2

-> P0

4.1.2 : Optionnalités distantes ou interdépendantes

Le fait qu'une position soit effectivement réalisée en surface (ou non) peut avoir des incidences sur la réalisation des autres positions :

- leur réalisation devient interdite
- leur réalisation devient obligatoire

Ces phénomènes seront exprimés par des conditions sur les optionnalités. Celles-ci permettent d'exprimer des contraintes sur l'optionnalité mutuelle des positions.

*Ex : Jean **attire** Marie à lui*

*Jean **attire** Marie*

*Jean **attire***

Jean **attire à lui*

P0 (P1) (P2)

Cond : Si !P1 Alors !P2

Les conditions étant orientées, on en émettra autant que nécessaire.

Ex : Cela ajoute des souffrances à sa peine

Cela ajoute des souffrances

Cela ajoute à sa peine

**Cela ajoute*

P0 (P1) (P2)

Cond : Si !P1 Alors P2

Cond : Si !P2 Alors P1

On pourra ainsi rendre compte de phénomènes tels que :

• **Positions mutuellement exclusives :**

Les positions ne peuvent être réalisées ensemble, l'occurrence de l'une implique l'absence de l'autre et vice-versa :

Ex : Jean dit de Marie qu'elle est belle cf. [M. Gross 75]

Jean dit à Paul que Marie est belle

** Jean dit à Paul de Marie qu'elle est belle*

**? Jean lui dit de Marie qu'elle est belle*

cb : P0 SELF (P1) (P2) P3

P0 : SN

P1 : SP[Prep:à]

P2 : SP[Prep:de]

P3 : P[SsCatSynt:COMPLETIVE]

Cond : Si P1 Alors !P2

Cond : Si P2 Alors !P1

(Il s'agit d'un cas où le lexicographe a réuni tous les arguments dans un même cb et où la structure maximale n'est jamais entièrement réalisée. La présence de P1 et P2 ensemble est impossible.)

• **Positions optionnelles ensemble :**

Les Positions ne peuvent être réalisées ou effacées qu'ensemble, l'absence de l'une entraîne l'absence de l'autre :

(P1) (P2)

Cond : Si !P1 Alors !P2

Cond : Si !P2 Alors !P1

• **Toute autre contrainte** plus ou moins complexe :

En particulier, le pouvoir expressif de telles conditions vaut quel que soit le nombre de Positions optionnelles (et notamment lorsque plus de deux le sont).

Ex : Il a été répondu oui à Luc par le service du personnel

Il a été répondu oui à Luc

Il a été répondu oui par le service du personnel

Il a été répondu oui

Il a été répondu à Luc

db : Self cb

Self : V[Aux:ETRE][Temps:COMPOSE]

cb : P0 SELF (P1) (P2) (P3)

P0 : PRO[Lex:il]

[SsCatMorph:IMPERSONNEL]

P1 : SN

P

P2 : SP[Prep:à]

P3 : SP[Prep:par]

Cond : Si !P1 Alors (P2 et !P3)

Cond : Si !P2 Alors P1

Soit en extension :

P0 SELF P1 P2 P3

P0 SELF P1 P2

P0 SELF P1 P3

P0 SELF P1

P0 SELF P2

4.2 : Actualisation des Positions

4.2.1 : Phénomènes linguistiques

Nous avons vu qu'une position, (définie formellement par un paradigme de substitution, c'est-à-dire l'ensemble des syntagmes pouvant l'occuper), peut être ou non réalisée en surface, actualisée en d'autres termes.

En ce qui concerne l'actualisation des positions en surface, différents phénomènes linguistiques traditionnellement distingués appellent des commentaires :

1. adjonction de modifieurs

Ex : La porte gauche du palier que j'ai ouverte.

-

2. coordination

Ex : La porte gauche et qui s'ouvre mal

La porte que j'ai ouverte et refermée

3. apposition

Ex : La porte, ce seuil de la maison

• Vision atomiste

Les 3 phénomènes linguistiques sont des phénomènes de surface distincts qui appellent des analyses et des traitements différents en terme de grammaire.

Ils ne sont pas du ressort du lexique qui se contente d'enregistrer les positions appelées par l'entrée lexicale ; c'est la grammaire qui gère leur actualisation simple ou multiple et en donne la représentation adéquate.

Au niveau du lexique donc, on considère que toute position peut avoir une réalisation nulle, simple ou multiple, c'est-à-dire qu'elle peut être instanciée par les Syntagmes entrant dans sa distribution, et ce quelle que soit sa fonction. Ces Syntagmes sont au nombre de 0 à N si la Position est optionnelle, 1 à N si la Position est obligatoire.

-

• Vision syntaxiste

Les trois phénomènes recouvrent des faits linguistiques très différents.

1. adjonction de modifieurs

Est perçue comme une propriété de la fonction associée à la position, les modifieurs étant, par essence, répétables N fois.

A l'inverse, la fonction sujet a pour propriété de ne pas être répétable.

2. coordination

Est perçue comme une méta-opération (cf. HPSG) sur les syntagmes. Cette opération s'applique potentiellement à tous les syntagmes d'une même distribution (position) suivant des règles de grammaire très complexes.

Elle n'est pas du ressort du lexique sauf à signaler les coordinations obligatoires

Ex : A ses risques et périls

et à en donner la représentation adéquate au cadre théorique

Ex : SN -> SN CONJ [SsCatMorph : COORDINATION] SN

3. apposition

Est perçue comme un élément de stylistique extérieur aux positions, à leurs occupants, géré - dans le meilleur des cas... - par la grammaire.

On le constate, ces phénomènes linguistiques (et notamment l'adjonction de modifieurs) reçoivent des interprétations très différentes suivant qu'on a une vision atomiste ou syntaxiste du lexique.

En conséquence, l'indicateur de répétabilité porté par les positions s'interprétera différemment suivant le cadre théorique où l'on se place.

4.2.2 : Indicateur de répétabilité sur les Positions

Le nombre possible des réalisations en surface est exprimé par un indicateur de répétabilité porté par la position. Cet indicateur (*repetable*) a pour valeurs OUI/NON (cette dernière valeur s'applique notamment aux mots composés, pour les positions saturées par un composant).

Ce sont donc des propriétés locales aux positions (à la différence des optionnalités distantes).

-

• Vision atomiste

Par défaut, toute position est actualisable n fois, les positions optionnelles sont actualisables 0 à n fois : dans tous les cas, l'attribut *repetable* est implicitement à OUI.

On ne s'interdit toutefois pas de pouvoir bloquer, le cas échéant, cette propriété d'actualisation multiple. Dans ces cas seulement, on utilisera la valeur NON.

• Vision syntaxiste

Par défaut, les positions ne sont pas répétées et elles ne sont actualisées que par un et un seul occupant, les positions optionnelles sont actualisées ou non par un occupant : dans les deux cas *repetable* = NON.

Lorsqu'on veut signaler qu'une position est répétée, on utilise la valeur OUI.

Dans les deux cas cependant, et fort heureusement, ces valeurs permettront de rendre compte des mêmes phénomènes linguistiques, c'est-à-dire de la cooccurrence en surface d'un SAdj, SP, P [SsCatSynt:RELATIVE] par exemple.

*Ex : La **porte** gauche du palier que j'ai ouverte*

(P0) P1* SELF P2*

P2 : SAdj

SP[Prep:de]

P[SsCatSynt:RELATIVE]

Seule la lecture varie : dans la vision atomiste on répète l'occupant, dans la vision syntaxiste on répète la position.

4.3 : Linéarité

On établit une linéarité canonique de convention : le "rang" d'apparition d'une position dans une construction (noté dans les exemples par un numéro) ne correspond pas nécessairement à une place précise dans l'ordre linéaire de surface. Il lui est attribué dans l'ordre linéaire canonique, et ce de façon (partiellement) arbitraire, et désigne l'ordre d'enregistrement dans une construction. Le terme de position n'est donc pas à prendre au sens strict de place (distinction établie par Milner [1989]), la contrainte de linéarité de surface ayant été écartée.

4.3.1 : Linéarité canonique

On veut référer aux positions telles qu'elles sont définies dans la structure maximale [Fradin & Marandin, 1979], même si cette structure maximale n'est jamais rencontrée en surface. Ainsi, certains verbes décrits comme ayant trois compléments ne verront jamais ces trois compléments réalisés conjointement en surface.

*Ex : Jean **dit** de Marie qu'elle est belle cf. [M. Gross 75]*

*Jean **dit** à Paul que Marie est belle*

** Jean **dit** à Paul de Marie qu'elle est belle*

? Jean lui **dit de Marie qu'elle est belle*

(Voir positions inter-conditionnées)

Le bouleversement de la linéarité canonique fait partie d'un système de règles de grammaire, voire de règles stylistiques, qui dépendent éventuellement du type de positions et qui n'ont en général pas leur place dans la description des constructions de l'unité syntaxique.

4.3.2 : Linéarité et transformations

Certains phénomènes de pronominalisation peuvent bouleverser l'ordre de réalisation de ces positions en surface.

*Ex: Je **pense** à Pierre*

*Je **pense** à lui*

*J'y **pense***

Le français offre en outre une assez grande liberté quant à la place qu'occupent les différents constituants de l'énoncé :

*Ex : Je **promets** de venir à Pierre*

*Je **promets** à Pierre de venir.*

4.3.3 : Linéarité et optionnalité

Nous avons vu que les réalisations de positions peuvent être facultatives. A une structure P0 (P1) (P2) peuvent correspondre les réalisations P0 P1 P2, P0 P1, P0 P2 et P0.

Ex : *Quelqu'un **parle** de quelque chose à quelqu'un*

*Quelqu'un **parle** de quelque chose*

*Quelqu'un **parle** à quelqu'un*

*Quelqu'un **parle***

L'omission de P1 ne doit pas conduire à la redéfinition de P2 en P1.

4.4 : Phénomènes de surface

Les positions d'une construction étant rangées dans un ordre canonique, on permet d'indiquer la place du Self au milieu de ces positions, par recours à l'attribut *inseself* (ce, bien entendu dans la vision syntaxiste où l'on conçoit la construction comme un syntagme).

Ex : P0 SELF P1 (P2)

correspond à *inseself* = 1

De plus, si l'unité syntaxique est composée, on pourra noter les éventuels phénomènes de réentrance d'éléments de la construction externe dans la structure interne, par recours à l'attribut *insereinsertion_l*. Son fonctionnement est comparable à celui de l'attribut *inseself*, si ce n'est que plusieurs insertions peuvent se succéder entre deux positions.

Ex : P0 **I0** P1

correspond à *insereinsertion_l* = 1

Ces indications ne décalent pas la numérotation canonique, pas plus qu'elles ne remettent en cause la canonicité de la numérotation des positions, c'est-à-dire que les règles de bouleversement de la linéarité de surface peuvent toujours s'appliquer.

Il arrive toutefois que l'ordre de surface soit contraint pour telle ou telle position :

Ex : *Il **crain**t de Marie qu'elle ne vienne.*

cb : P0 SELF **P1** P2

P0 : SN

P1 : SP[Prep:de]

P2 : P[SsCatSynt:COMPLETIVE]

On ne permet pas d'indiquer, dans la vision actuelle du modèle, comment les règles de bouleversement de linéarité peuvent être inhibées.

4.5 : Solidarité des Positions

Certaines positions sont solidaires, c'est-à-dire qu'elles n'autorisent entre elles aucune insertion d'aucun type : il s'agit d'une suite insécable. Cette convention est utile pour la composition, pour interdire par exemple l'insertion d'incises, d'appositions, etc... L'ordre d'apparition en surface n'est cependant pas nécessairement contraint (cf. *temps plein, plein temps*).

Ex : P1-P2

Le **tiret** entre deux positions indique que ces deux positions sont solidaires, c'est-à-dire qu'aucune insertion n'est possible entre elles et que si elles se déplacent, elles se déplacent ensemble.

4.6 : Réalisations de Positions inter-conditionnées

Il faut pouvoir exprimer que les réalisations de positions peuvent contraindre la réalisation d'autres positions. C'est au moyen de **Conditions** que l'on formulera ces contraintes.

*Ex : Pierre **répond** à la question.*

*Pierre **répond** que c'est exact.*

*Pierre **répond** à sa mère que c'est exact.*

*Qu'il ait une telle attitude **répond** à la question.*

Qu'il ait une telle attitude **répond que c'est exact.*

cb : P0 SELF (P1) (P2)

P0 : SN

P[SsCatSynt:COMPLETIVE]

[Mode:SUBJONCTIF]

P1 : SP[Prep:à]

P2 : P[SsCatSynt:COMPLETIVE]

[Mode : INDICATIF]

Cond : Si P0 == P[SsCatSynt : COMPLETIVE]

[Mode : SUBJONCTIF]

Alors !P2

Ex : Il a volé un livre à Marie

Il a volé un livre

Il a volé Marie

Si le lexicographe décide que *Marie* et *à Marie* sont différentes réalisations de la même position (RoleTh PATIENT), la réalisation de cette position devra être conditionnée par l'actualisation de la précédente :

cb : P0 SELF (P1) (P2)

P0 : SN

P1 : SN

P2 : SN

SP[Prep : à]

Cond : Si !P1

Alors P2 = SN

Cond : Si P2 = SN

Alors !P1

4.7 : Conditions entre Self et les Positions

Les contraintes de réalisations peuvent également porter sur l'actualisation du Self en tant qu'Intervenant dans la Construction.

Ex : Jean a descendu la colline

Jean est descendu de la colline

Si on choisit d'avoir une seule Usyn décrivant l'emploi purement "locatif" de *descendre* (par opposition à *descendre quelqu'un* ou *sa valise*) alors on pourra lui attacher les conditions suivantes :

```

Cond : Si Self == V[Aux:ETRE]

Alors P1 != SN

Cond : Si Self == V[Aux:AVOIR]

Alors P1 != SP[SsCatSynt:LIEU]

```

VII : Syntagmes

1 : Définition générale

Une position peut s'actualiser de diverses façons. On appelle **Syntagme** une réalisation donnée de position.

Un syntagme occupant une position est formellement décrit par une étiquette syntagmatique (attribut *etiquettesynt*), de niveau terminal ou non-terminal, à laquelle est associé, s'il y en a, un ensemble de contraintes (contraintes de réécriture - pour les seuls syntagmes non-terminaux - et traits restrictifs).

2 : Etiquette Syntagmatique

On admet a priori une nomenclature de symboles syntaxiques de niveau maximal {P, SV, SN, SP, SADV, SADI}, à laquelle s'ajoutent l'élément vide {e}, {Nbarre} (noté N') et la liste des catégories terminales telle que définie en morphologie.

On suppose que ces symboles sont connus et décrits dans une grammaire extérieure au dictionnaire. GENELEX en propose une liste extensible par l'utilisateur.

Nous donnons néanmoins quelques précisions sur la sémantique de *e* et de Nbarre.

La catégorie *e* sert traditionnellement à exprimer

- l'optionnalité d'un constituant immédiat dans les grammaires distributionnelles et transformationnelles. Ceci est dû à la pauvreté du formalisme et n'a pas sa place dans GENELEX. Rappelons que DET_VIDE a pour correspondance en réécriture l'absence ou l'optionnalité de la Position Spécifieur.
- les traces en grammaire générative. C'est pour cela qu'elle a été introduite dans GENELEX et elle ne doit avoir aucune autre sémantique. On veut absolument éviter tous les détournements d'usage qui conduiraient à faire d'une position occupée par *e* une "Position vide" pour des raisons formelles et non linguistiques.

Pour ce qui est de l'étiquette Nbarre, nous l'avons introduite essentiellement pour les *Syntagme_NT_S*, syntagmes qui permettent de décrire les Structures internes des composés. En effet, si le composé dans son ensemble peut jouer le rôle de N en tant qu'**IntervConst**, c'est bien qu'il n'est pas en soi un SN : il lui manque au moins ses pré-déterminants. Nous avons pris le parti de recourir à cette étiquette qui nous semblait la moins ambiguë pour noter un niveau de réalisation intermédiaire entre N et SN.

3 : Contraintes par Traits

Les syntagmes occupant les positions peuvent être restreints par un ensemble de **Traits**. La combinaison de ces traits permet de spécifier un occupant de position. Ces restrictions peuvent aussi être exprimées par la réécriture du syntagme (voir plus bas). Le lexicographe pourra respecter un principe d'économie et n'utiliser la réécriture que lorsque les traits dont ils dispose ne lui permettent pas de spécifier toute l'information nécessaire.

L'utilisation de traits (développée par les grammaires d'unification LFG, HPSG, etc.) permet au lexicographe de coder des descriptions syntaxiques à différents degrés de finesse. Elle permet aussi d'imaginer une hiérarchie Générique-Spécifique sur des syntagmes et de pointer directement sur le niveau requis : SP, SP[Prep:à], SP[Prep:à][SsCatMorph:DEFINI], sans pour autant déclarer explicitement cette hiérarchie. Le lien est déductible du partage de l'étiquette syntagmatique et de l'inclusion d'une liste de contraintes - ainsi que de l'inclusion des distributions et autres relations entre syntagmes lorsqu'ils sont réécrits.

On distingue différentes familles de traits accessibles depuis la couche syntaxique, tous ces traits pouvant s'ajouter les uns aux autres.

Ex : V[Aux:ETRE]

[Pronominal:SE]

[Temps:COMPOSE]

[Aspect:PROCESSIF]

L'usage des traits restrictifs est régi par un ensemble de règles, détaillées dans le chapitre les concernant (Chapitre VIII).

4 : Contraintes de structure

Dans la majorité des cas, l'étiquette référant le syntagme suffit à le décrire comme occupant de la position, aucune contrainte sur sa structure n'étant à exprimer pour l'entrée décrite. Il s'agit en quelque sorte d'un prototype absolu.

Mais on peut avoir besoin d'émettre depuis une entrée lexicale des contraintes sur la structure où elle s'insère.

Pour ce faire on utilise :

- soit des traits de sous-catégorie syntaxique
- soit des listes de positions enchâssées qui permettent de décrire des arbres syntaxiques de profondeur n ; on parlera de réécriture arborescente de syntagmes.

4.1 : Le Trait de sous-catégorisation syntaxique

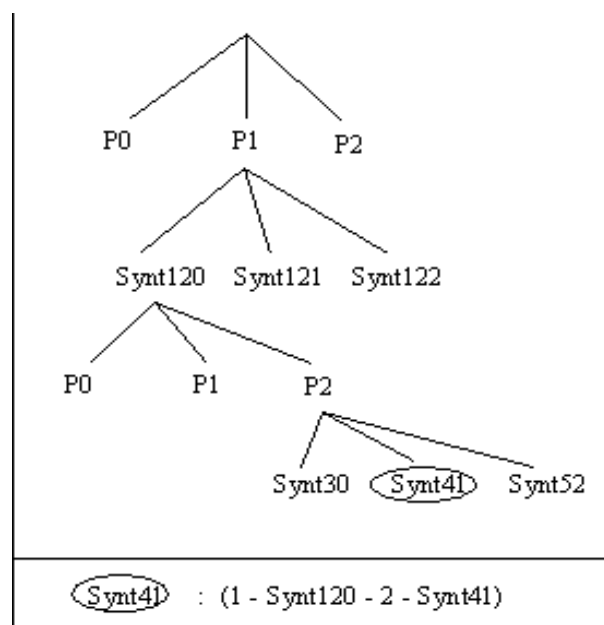
Certaines valeurs de sous-catégories syntaxiques spécifient implicitement un certain type de structuration (ex : DET_VIDE). Lorsqu'on utilise les traits de sous-catégorisation à ces fins, on ne doit pas perdre de vue que la valeur de SsCatSynt n'est qu'une chaîne de caractère sur laquelle le formalisme ne fait pas de contrôle de cohérence : l'interprétation du trait par une grammaire passe par la définition de ce trait ; la cohésion doit être assurée par l'équipe lexicographique.

4.2 : La réécriture arborescente

4.2.1 : Description

La réécriture des syntagmes, (applicable aux seuls syntagmes non terminaux), se fait par recours au formalisme des positions. Chaque syntagme peut être réécrit sous forme d'une liste de positions, pour laquelle on peut préciser les conditions d'optionnalité, l'ordre de surface et la solidarité.

Un **arbre syntaxique** dans GENELEX fera alterner à chaque niveau position et syntagme :

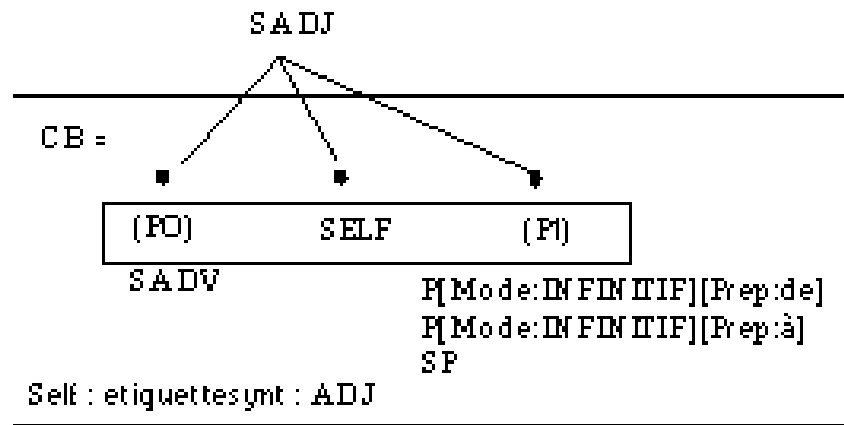


Le cheminement dans un arbre syntaxique s'exprime par une alternance d'entiers et d'identifiants. Chaque entier indice une position dans sa liste, chaque identifiant réfère un syntagme de réécriture de cette position.

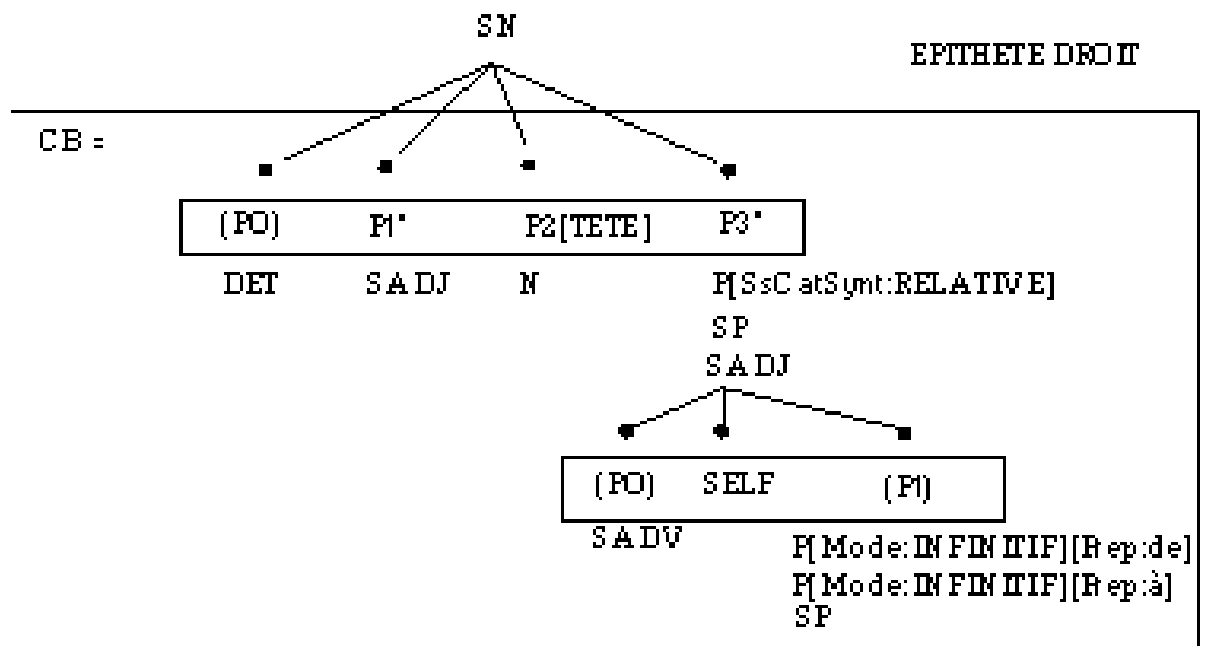
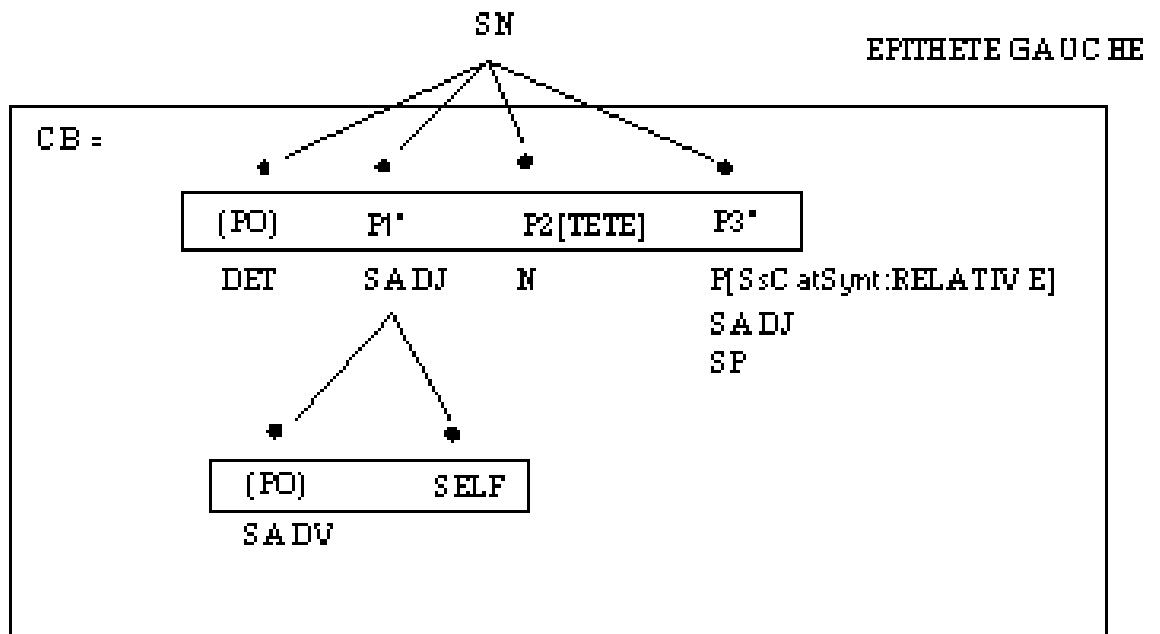
4.2.2 : Exemples d'utilisation

- description des adjectifs

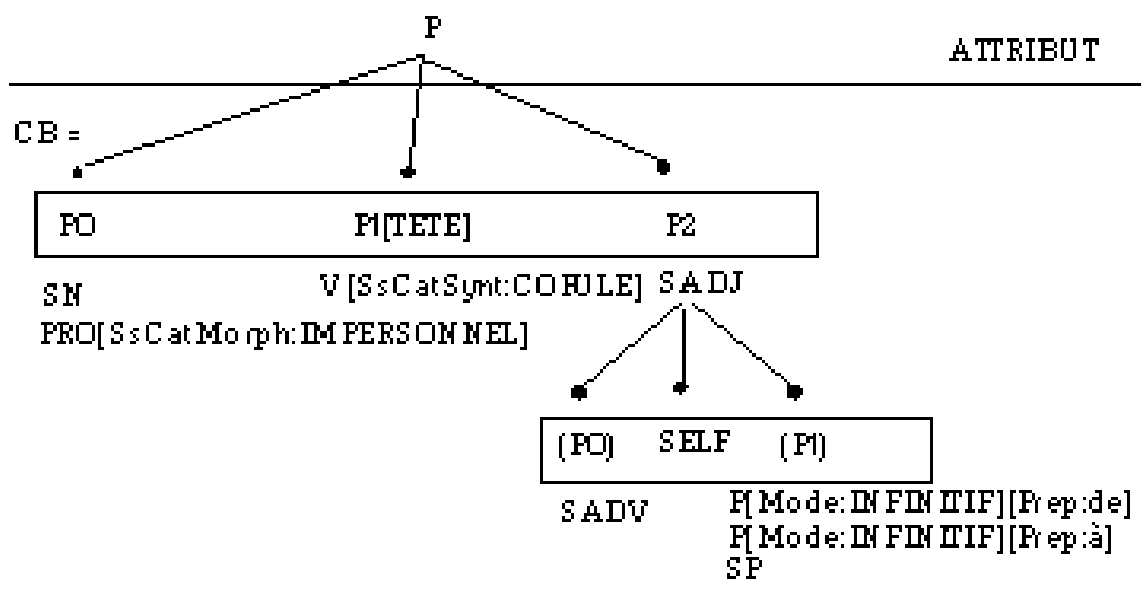
La réécriture d'un SADJ se fera par le biais des entités GENELEX de la manière suivante :



• description des adjectifs épithètes gauche et droit



• description des adjectifs attributs du sujet



Ex : Il est très *intéressant* de remarquer cela

Il est très *intéressant* que tu remarques cela.

db : Self cb

Self : IntervConst : ADJ

cb : P0 P1 P2 P3

P0 : PRO[Lex:il][SsCatMorph:IMPERSONNEL]

P1 : V[SsCatSynt:COPULE]

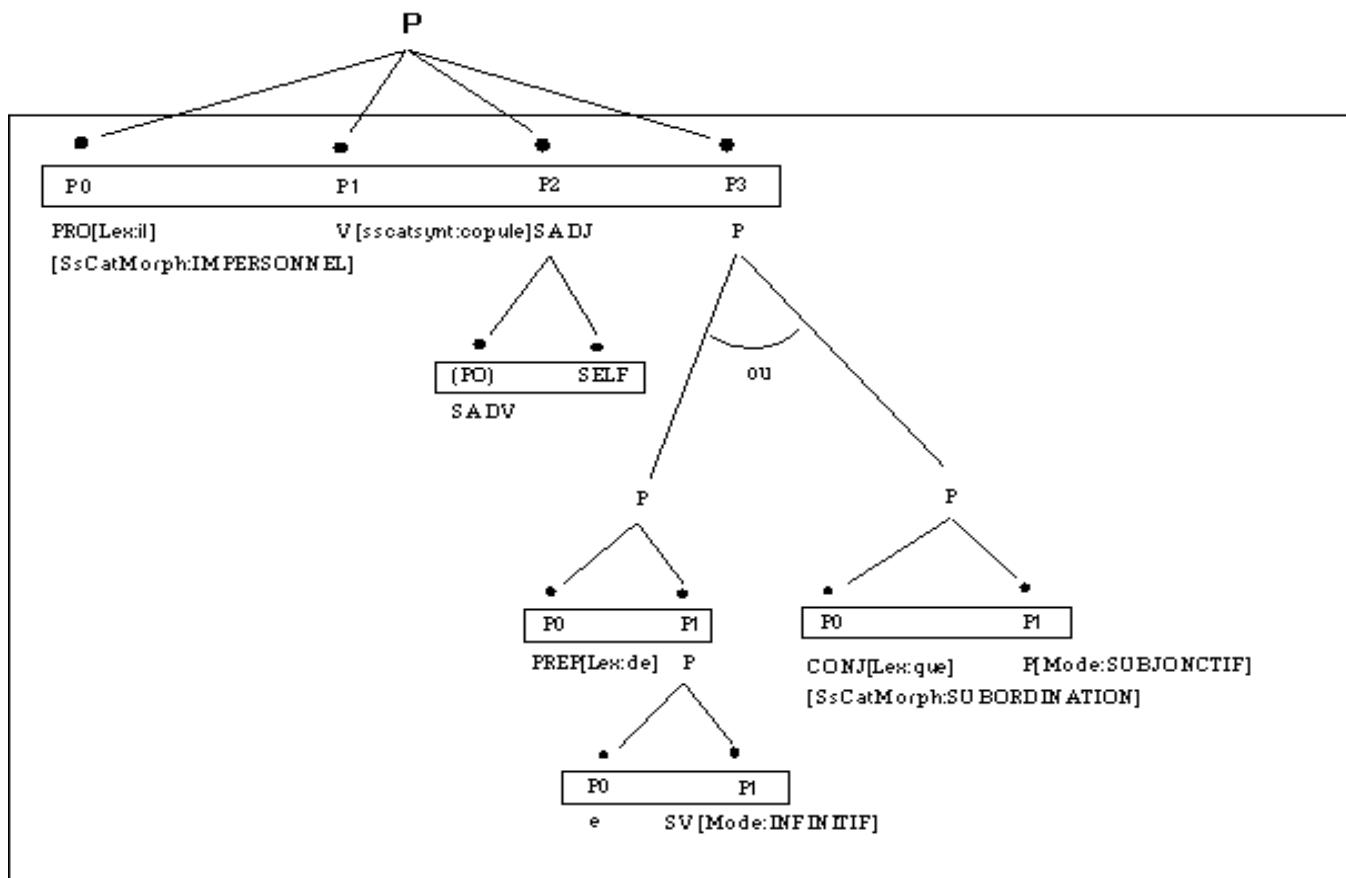
P2 : **SADJ**

P3 : P[Prep:de][Mode:INFINITIF]

P[Conj:que][Mode:SUBJONCTIF]

SADJ : (P0) SELF

P0 : SADV



4.3 : Réécriture partielle de Syntagmes

Pour certaines entrées, on veut pouvoir émettre des restrictions partielles sur un syntagme sans avoir à le réécrire totalement, car ce n'est pas toujours possible ; par exemple, dans le cas d'un syntagme verbal, on ne sait pas quel est le nombre de positions si l'on ne connaît pas la tête.

Dans ce cas, il y a conservation de la structure d'un syntagme "prototypique" (liste de positions), mais certains occupants de position sont restreints, tant au niveau de leur liste (retrait d'occupants), qu'au niveau des contraintes (ajout de traits restrictifs) sur chacun.

On ne donne alors que la liste des positions dont on restreint la liste ou le type des occupants, et on a recours à l'attribut *listepositions* dont les valeurs (OUVERTE, FERMEE), permettent de préciser si la liste des positions de réécriture spécifie totalement (FERMEE) ou partiellement (OUVERTE) la réécriture du syntagme.

VIII : Les Traits

Les **Traits** sont des restrictions s'ajoutant à l'étiquette syntagmatique dans la spécification d'un Syntagme (voir Chapitre VII).

Nous allons tout d'abord détailler les différents types et sous-types de traits utilisables, leurs attributs et les valeurs qu'ils peuvent prendre.

Nous aborderons ensuite les différentes contraintes régissant leur emploi, pour chaque type de trait

mais aussi pour les différentes combinaisons envisageables de ces traits.

1 : Types, sous-types et valeurs de Traits

1.1 : Traits lexicaux

Les traits lexicaux permettent de contraindre tout ou partie de la lexicalisation d'un syntagme.

Pour ce faire, on peut :

- soit spécifier la graphie canonique de l'Um "lexicalisante",
- soit pointer directement vers cette Um (et lever ainsi une éventuelle ambiguïté sur la graphie).

On distingue deux types de traits lexicaux : les introducteurs et le trait Lex proprement dit.

1.1.1 : Les Introducteurs (Prep, Conj, ProRel, ProIntrog et Introd)

Généralités

Ces traits permettent de spécifier la lexicalisation des introducteurs de syntagmes sans avoir à les réécrire. Leur portée est non ambiguë et ils ne s'appliquent qu'à des syntagmes non terminaux.

Il s'agit en fait d'un sous-type qui regroupe 5 traits.

-

Les différents types de traits

1.- Prep

pour la préposition introduisant un syntagme non terminal, typiquement un Syntagme Prépositionnel.

Le trait **Prep** a trois emplois, selon le type du syntagme qui le porte :

P[Prep : à] [Mode : INFINITIF]

introducteur (au sens plein) de syntagme

SP[Prep : à]

élément de réécriture de syntagme

A ces deux emplois "standards", nous adjoignons la possibilité, pour la préposition, d'être un introducteur au sens plein de syntagme nominal. On peut en effet vouloir distinguer les cas où la

préposition est régie par un élément extérieur au syntagme prépositionnel, cas classique dans la rection verbale par exemple, des cas où elle semble contrôlée par le substantif tête du groupe prépositionnel comme dans en nourrice par exemple. On autorisera donc le syntagme non-terminal SN à porter le trait **Prep**.

SN[Prep:en][Lex:nourrice]

introduceur (au sens plein) de syntagme

2. - Conj

pour la conjonction introduisant une Phrase.

3. - ProRel

pour le pronom relatif introduisant une Phrase.

4. - ProIntrog

pour les pronoms interrogatifs qui peuvent être notamment spécifiés par les verbes qui appellent des interrogatives (verbes de discours principalement).

5. - Introd

pour tout introduceur (particule, "recatégorisateur", etc.) de syntagme non terminal n'appartenant à aucun des quatre autres types.

Réécriture

Dans l'alternative de réécriture on utilisera le trait **Lex** (voir paragraphe suivant) sur les syntagmes terminaux :

EX : SP[Prep:à]

pourra être réécrit :

SP : P0 P1

P0 : PREP[Lex:à]

P1 : SN[. . .]

si l'on veut spécifier davantage le SN du SP.

1.1.2 : Lex

Le trait **Lex** permet de spécifier la lexicalisation

- d'une feuille syntagmatique, s'il s'applique à un syntagme de catégorie terminale
- de la tête d'un Syntagme, s'il s'applique à un Syntagme de catégorie non terminale. Puisqu'on s'est donné par ailleurs les Introduceurs, on admettra de fait que les têtes syntagmatiques sont :

N pour SN

N pour SP

V pour SV ou P

ADJ pour SADJ

ADV pour SADV

On pourra noter que la structuration dudit syntagme est implicitement déterminée par la tête lexicalisée, plus précisément par le comportement syntaxique de son Usyn tête (qui existe par ailleurs). Si l'on veut contraindre plus avant la structuration du syntagme, on pourra utiliser des traits de sous-catégorie, ou le réécrire.

Un cas de restriction particulier est celui où le syntagme est saturé par sa tête, c'est-à-dire que le SN se restreint au N. On se dote, pour exprimer cette propriété de l'attribut *saturesynt*. Lorsque sa valeur est OUI, la tête du syntagme en est la seule feuille avec l'Introduceur, s'il y en a un. Ceci vaut pour les syntagmes entrant dans la définition d'Usyn simples comme d'Usyn composées.

Le champ *saturesynt* doit toujours avoir la valeur OUI lorsque le trait porte sur une catégorie terminale : la feuille est par définition saturée par l'élément lexicalisant.

1.1.3 : Cooccurrence de traits lexicaux

Un même syntagme ne pourra supporter qu'un et un seul trait Lex qui lexicalisera dans un cas le syntagme terminal, dans l'autre la tête et la tête seulement du syntagme non terminal.

Chaque syntagme non terminal n'ayant qu'un introduceur, un même syntagme ne pourra donc supporter qu'un seul trait spécifiant son introduceur.

En revanche, pour un syntagme non terminal, les introduceurs pourront être combinés au trait Lex.

Ex : SP[Prep:de][Lex:concert]

Ces remarques valent par effet miroir pour les traits RefLex, RefPrep, RefConj, RefProRel, RefProIntro, RefIntro, dont la description se trouve dans le chapitre sur les unités syntaxiques composées.

1.1.4 : Cas particulier des Agglutinés

L'agglutination est un phénomène morphologique qui n'a pas d'incidence sur la représentation syntaxique.

Le trait Lex ne sera donc en général pas utilisé pour référencer un agglutiné. On préférera retrouver ce dernier par référence aux éléments agglutinants.

Ainsi, un même Syntagme

Ex : SP[Prep:à][SsCatMorph:DEFINI]

pourra être ou non réalisé en surface par une forme agglutinée :

Ex : au pharmacien

à la pharmacienne

1.2 : Traits morphologiques

Ils permettent d'exprimer une restriction sur la valeur d'un trait morphologique (**Mode, Temps, Personne, Genre, Nombre, NombrePosseur**) du syntagme ou de l'un de ses composants.

Ex : SN[Nombre:PLURIEL]

Mode, Temps, Personne, Genre, Nombre, NombrePosseur sont des traits qui peuvent se combiner pour former, précisément, une combinaison de traits morphologiques.

Ex : [Mode:SUBJONCTIF]

[Temps:PRESENT]

[Personne : 3]

[Nombre : SINGULIER]

1.3 : Traits morpho-syntaxiques

Cette famille de traits regroupe **SsCatMorph**, **Aux**, **Pronominal** et **Accord**. Toutes les contraintes émises par ces traits peuvent s'ajouter les unes aux autres.

Ex : V[Aux:ETRE][Pronominal:SE]

1.3.1 : SsCatMorph

Ce trait a pour valeurs possibles les sous-catégories morphologiques (cf. Rapport GENELEX sur la Couche Morphologique) qui sont des critères d'éclatement d'Um.

Ils peuvent porter sur les syntagmes terminaux et non terminaux.

Ex : SN[SsCatMorph:DEFINI]

DET[SsCatMorph:DEFINI]

La liste des valeurs est finie et correspond à la liste des sous-catégories grammaticales définies en morphologie :

PROPRE, COMMUN, POSSESSIF, DEMONSTRATIF, PARTITIF, DEFINI, INDEFINI, INTERROGATIF, CARDINAL, ORDINAL, RELATIF, PERSONNEL_FORT, PERSONNEL_FAIBLE, IMPERSONNEL, EXCLAMATIF, QUALIFICATIF, COORDINATION, SUBORDINATION, COMPLETIF, COMPARATIF_EGALITE, COMPARATIF_SUPERIORITE, COMPARATIF_INFERIORITE, SUPERLATIF_SUPERIORITE, SUPERLATIF_INFERIORITE, SUPERLATIF_ABSOLU.

On remarquera que certaines valeurs de traits de sous-catégorie morphologique recouvrent entièrement un paradigme de formes fléchies.

C'est le cas des déterminants défini et partitif :

- Déterminant défini : { *le, l', la, les* }
- Déterminant partitif : { *du, de l', de la, des* }

Dans ces cas, et dans ces cas seulement, il sera toujours possible d'adjoindre ou de substituer un trait lexical au trait de sous-catégorie :

Ex : DET[SsCatMorph:DEFINI]

DET[Lex:le]

DET[SsCatMorph:DEFINI] [Lex:le]

les 3 syntagmes induisant une même lexicalisation.

Cependant, même s'ils réfèrent la même unité, le trait de sous-catégorie reste le plus parlant et le plus puissant des deux, car il peut porter sur un syntagme non terminal.

Ex : SN[SsCatMorph:PARTITIF]

1.3.2 : Aux

Aux est un trait complexe dont la **Valeur** (ETRE ou AVOIR) peut être complétée par une restriction de **Temps** et de **Mode** sur l'auxiliaire.

Le plus souvent, seule la Valeur est renseignée, et spécifie pour un verbe donné l'auxiliaire qui lui est associé dans telle tournure syntaxique :

Ex : *descendre* à la cave (*être* descendu)

V[Lex:descendre] [Aux:ETRE]

descendre les escaliers (*avoir* descendu)

V[Lex:descendre] [Aux:AVOIR]

On peut avoir à indiquer que, dans une tournure donnée, le verbe appara"t à un Temps composé :

Ex : *être arrivé* socialement

V[Lex:arriver] [Temps:COMPOSE] [Aux:ETRE]

Mais on peut aussi avoir besoin de contraindre le Temps et le Mode de l'auxiliaire de conjugaison :

Ex : *étant donné*

V[Lex:donner] [Temps:COMPOSE]

[Aux:ETRE[Mode:PARTICIPE] [Temps:PRESENT]]

1.3.3 : Pronominal

Ce trait permet de spécifier pour un verbe donné la particule préverbale non référentielle qui lui est associée dans telle tournure syntaxique. On rend ainsi compte des comportements de verbes dits "vrais pronominaux" :

Ex : se plaindre

V[Lex:plaindre][Pronominal:SE]

Les valeurs de ce trait sont les suivantes en français :

SE, LE, LA, LES, EN, Y, SE_LE, SE_LA, SE_LES, SE_EN, SE_Y.

Ex : les avoir à zéro ; en baver ;

la ramener ; se le tenir pour dit ;

se la couler douce ; s'y coller ;

s'en moquer.

On notera que la spécification de la particule préverbale peut se faire sur le Self d'une unité simple ou composée.

Ex : se rendre compte

1.3.4 : Accord

Ce trait, lorsqu'il porte sur plusieurs syntagmes terminaux ou non terminaux, permet d'exprimer des contraintes d'accord déterminées par une entrée lexicale, notamment certains noms déterminatifs.

Ex : la moitié des gens est/sont venu(e/s)

Self : IntervConst : N[Nombre:SINGULIER]

[Accord:I]

cb : P0 P1

P0 : **SN**

P1 : SV[Accord:I]

SV[Accord:J]

o• P0 : **SN** se réécrit :

SN : P0 (P1) SELF (P2)

P0 : DET

P1 : SADJ

P2 : SP[Nombre:PLURIEL][Accord:J]

Ce trait Accord permet donc l'enregistrement explicite des accords difficiles. Bien entendu, en l'absence de ce trait, les règles générales d'accord s'appliquent.

1.4 : Traits syntaxiques

Cette famille de traits regroupe **SsCatSynt**, **Tournure**, **Neg** et **Passif**. Toutes les contraintes émises par ces traits peuvent s'ajouter les unes aux autres.

Ex : V[Passif:PLUS][Neg:NE_PAS]

1.4.1 : SsCatSynt

Ce trait a pour valeurs possibles les sous-catégories syntaxiques qui étiquettent des sous-ensembles de comportements syntaxiques pour une catégorie donnée.

Il peuvent porter sur des syntagmes terminaux ou non terminaux.

Ex : P[SsCatSynt:COMPLETIVE]

V[SsCatSynt:COPULE]

SP[SsCatSynt:LIEU]

SP[SsCatSynt:DET_VIDE][Prep:en]

[Lex:conséquence] (*en conséquence*)

La liste des valeurs est ouverte, mais GENELEX propose par défaut les valeurs suivantes :

RELATIVE, COMPLETIVE, COORDONNE, INTERROGATIVE_DRI,
INTERROGATIVE_DRD, SUBORDONNEE, TEMPS, LIEU, MANIERE, DEGRE, QUANTITE,
COPULE, DET_VIDE, DETERMINATIF, COMPARATIF, SUPERLATIF.

1.4.2 : Tournure

La présence d'un trait **Tournure** permet d'indiquer si une Phrase est obligatoirement EXCLAMATIVE ou INTERROGATIVE.

Il nous faut ici faire une brève digression sur les interrogatives. Une typologie minimale permet de distinguer :

- Interrogatives subordonnées (relatives ou non) qui, dans la dépendance d'un verbe, relèvent du discours rapporté, direct ou indirect :

Ex : *SN demander si/pourquoi qqchse*

SN demander quel/combien

Ex : *SN demander "pourquoi qqchse"*

SN demander "quel/combien... ?"

Ce sont des sous-catégories (trait **SsCatSynt**) de phrases entrant dans la complémentation d'un verbe.

- principales "libres", qui ne sont pas rapportées dans le discours. On parle de tournures interrogatives (trait **Tournure**). Ces phrases ont une incidence sur l'inversion du sujet.

On a donc deux valeurs d'interrogative, rattachées à deux traits différents :

SsCatSynt : INTERROGATIVE_DRI, INTERROGATIVE_DRD

Tournure : INTERROGATIVE

1.4.3 : Neg

La présence d'un trait **Neg** indique que le syntagme décrit est à la forme négative. Dans ce cas,

- soit la valeur lexicale de la négation est libre (valeur : libre)

Ex : *n'arrêter pas/plus/jamais/...*

- soit elle est contrainte et on la choisira parmi :

NE_PAS, NE_PLUS, NE_JAMAIS, NE, NE_QUE, NE_MAIS, NE_GUERE, NE_POINT, NE_RIEN, NE_RIEN_QUE, NE_PAS_QUE, NE_PLUS_QUE, NE_JAMAIS_QUE, NE_GUERE_QUE.

Ex : *n'en pouvoir plus*

n'en pouvoir mais

ne faire rien que des bêtises

1.4.4 : Passif

La présence d'un trait **Passif** indique que la Phrase décrite est à la forme passive. Ce trait est un booléen (valeurs : PLUS ou MOINS).

NB : L'absence du trait Passif implique que la forme est active, ce qui évite de faire porter le trait [Passif:MOINS] partout.

1.5 : Traits syntaxico-sémantiques

1.5.1 : Coref

Le trait de coréférence (**Coref**) est le seul représentant de cette famille de traits.

Il permet d'exprimer les indices de coréférence utilisés dans les grammaires distributionnelles pour rendre compte de la référence pronominale, du contrôle des infinitives, etc.

Les valeurs que peut prendre ce trait sont :

I, J, K, L, NON_I, NON_J, NON_K, NON_L.

Ex : (se) *laver* => **réfléchi**

cb : P0 SELF P1

P0 : SN[Coref:I]

P1 : PRO[Lex:se][Coref:I]

On exprime ici la coréférence entre le sujet et le réfléchi, le verbe n'étant pas un réfléchi essentiel.

Par convention, lorsque le trait Coref porte sur une Phrase, il concerne le sujet de P.

Les coréférences doivent être résolubles : si un trait de valeur I est présent sur un syntagme d'une position d'une construction donnée, il existe au moins un trait I ou NON_I lui répondant, sur un syntagme d'une autre position de la même construction. Les traits Coref ne forcent pas la co-réalisation des syntagmes les portant. Si l'on souhaite imposer cette co-réalisation, on le fera comme d'habitude par le biais de conditions sur les réalisations de position.

Contrôle de l'infinitive

Ce contrôle peut prendre plusieurs formes, nous en illustrerons quelques unes :

Le contrôle du **sujet de l'infinitive**

- par le sujet de la principale

*Ex : Marie **aime** danser*

cb : P0 SELF P1

P0 : SN[Coref:I]

P1 : P[Mode:INFINITIF][Coref:I]

- par l'objet direct de la principale

*Ex : Marie **voit** les enfants danser*

cb : P0 SELF P1

P0 : SN

P1 : SN[Coref:I]

P2 : P[Mode:INFINITIF][Coref:I]

On a ici dissocié le complément d'objet de l'infinitive ; on peut aussi considérer que l'infinitive fait partie intégrante de l'objet, auquel cas l'infinitive a un sujet exprimé qui n'est pas concerné par la coréférence

*Ex : Marie **voit** les enfants danser*

cb : P0 SELF P1

P0 : SN

P1 : P[Mode:INFINITIF]

- par le sujet et l'objet direct de la principale

La cooccurrence de deux traits de coréférence (utilisant deux variables différentes) s'interprète comme la "somme" des valeurs :

*Ex : Marie **emmène** son fils faire un voyage*

P0 : SN[Coref:I]

P1 : SN[Coref:J]

P2 : P[Mode:INFINITIF]

[Coref:I][Coref:J]

- par l'objet indirect (ou tout autre objet second) de la principale

*Ex : Jean **demande** à Marie de venir*

P0 : SN[Coref:I]

P1 : SP[Prep:à][Coref:J]

P2 : P[Mode:INFINITIF][Coref:I]

P[Mode:INFINITIF][Coref:J]

Cond : Si P1 Alors P2[Coref:J]

Le contrôle de l'objet de l'infinitive

- par l'objet de la principale (présenté ici dissocié de l'infinitive)

*Ex : Jean **a donné** ce travail à faire*

P0 : SN

P1 : SN[Coref:I]

P2 : P[Mode:INFINITIF][Prep:à]

avec

P[Mode:INFINITIF]

[listepositions : OUVERTE] :

P0[SUJET] : e

P1[OBJET_DIRECT] : e[Coref:I]

- par le sujet de la principale

*Ex : Ce bateau **est** difficile à conduire*

P0 : SN[Coref:I]

P1 : SADJ

avec

SADJ

[*listepositions* : OUVERTE] :

P1 : P[Mode:INFINITIF]

avec

P[Mode:INFINITIF]

[*listepositions* : OUVERTE] :

P0[SUJET] : e

P1[OBJET_DIRECT] : e[Coref:I]

1.6 : Traits sémantiques

Pour ceux qui ont choisi de recourir explicitement dès la syntaxe à des propriétés sémantiques, il est possible de les noter sous forme de traits.

Ces traits recouvrent **Aspect** du verbe, conditions dénotationnelles (Trait **Bin**) et classe sémantique (Trait **Libre**).

1.6.1 : Aspect du verbe

Certains verbes apparaissant dans le contexte de l'entrée ne sont utilisés qu'avec tel ou tel aspect.

Certaines tournures verbales contraignent un **Aspect** PROCESSIF, STATIF ou RESULTATIF.

Ex : V[Aspect:PROCESSIF]

1.6.2 : Conditions Dénotationnelles (Traits Bin)

Les CD (notion introduite par G.Bès) notent les différences minimales entre deux constructions. Prenons le cas d'un verbe pour lequel on identifie un P0 et un P1[Fonction:OBJET_DIRECT]. Différents objets, et donc P1, pourront être distingués par leurs conditions dénotationnelles.

Ex : (1) *il mange*

(2) *il mange une pomme*

P0 (P1)

(3) *il mange les virgules*

P0 P1

On veut pouvoir mettre en évidence deux structures P0 P1 et P0 (P1), c'est-à-dire ce qui différencie le P1 optionnel de l'autre. Ces propriétés syntaxiques sont liées aux conditions dénotationnelles. Ce qu'on note, c'est l'information sémantique minimale qui permet de caractériser une propriété syntaxique (optionnalité ou autre propriété syntaxique).

Il faut préciser ici que les conditions dénotationnelles dont nous parlons ne sont pas des conditions dénotationnelles sur l'univers d'interprétation de l'énoncé. Il ne s'agit pas d'une sémantique qui repose sur les valeurs de vérité, ou sur la vraisemblance d'un énoncé.

En effet, si "*un éléphant donne un pomme à Marie*" n'est pas un énoncé vraisemblable, il n'en est pas moins interprétable.

Les conditions dénotationnelles que nous notons sont celles qui permettent de bloquer certains processus interprétatifs : dans "*Jean mange la virgule*", on interprète "manger" dans le sens de "omettre", ce qui n'est pas le cas pour les énoncés suivants : "*Jean mange la pomme*", "*Jean mange*".

En recourant de manière explicite à des indications de type sémantique, on rend compte d'une propriété qui a des implications au niveau syntaxique (ici P1 est obligatoire). Les conditions dénotationnelles qu'on notera seront donc [comestible:MOINS] pour P1 en (2) et [comestible:PLUS] pour P1 en (1). Les conditions dénotationnelles sont des traits sémantiques de valeurs binaires.

Remarque : Des emplois figurés tels que "*la machine mange la carte*", et "*cette voiture mange beaucoup d'essence*" relèvent de la structure P0 (P1). C'est dire que, de façon "figurée", l'essence est comestible pour la voiture.

1.6.3 : Classe Sémantique (Traits Libre)

Très proche des conditions dénotationnelles, la classe sémantique s'en distingue formellement en ceci qu'elle n'a pas de valeurs binaires +/- mais un ensemble de valeur libres, et en ceci que le nom de la classe est plutôt un hyperonyme désignant un ensemble de réalisations lexicales possibles.

Ex : **couleur** : *jaune, bleu, rouge, ...*

discours : *speech, conférence, plaidoyer, ...*

vêtement : *veste, pantalon, chemise, ...*

profession : *médecin, commerçant, artisan, ...*

Les traits Bin et Libre pourront apparaître ensemble sur une même catégorie, notamment la catégorie NOM.

Ex : NOM[animé:PLUS][classe:profession]

2 : Contraintes d'intégrité sur les Traits

2.1 : Règles générales d'utilisation des Traits

- Les traits portent sur des syntagmes et des constructions.
- Chaque trait est au minimum défini par son nom et sa valeur (prise dans une liste de valeurs possibles, ou bien libre).
- On peut combiner autant de traits que l'on veut pour exprimer une restriction complexe.
- Toute liste de traits sur un syntagme doit être comprise comme une liste "et", c'est-à-dire que tous les traits de cette liste viennent ajouter des restrictions. On ne peut pas exprimer une alternative de traits sur un même syntagme. Il faut en passer par l'alternative de réalisation de position et donc par plusieurs syntagmes.

Ex : P[Prep:à][Mode:INFINITIF]

P[Prep:de][Mode:INFINITIF]

- Lorsque l'ensemble des syntagmes occupant une position partagent un ou plusieurs traits, il faut reporter ces traits sur chaque syntagme. La position ne factorise pas les traits de ses occupants.

Cependant, suivant leur type et leur valeur, tous les traits ne peuvent pas porter sur tous les syntagmes, et toutes les combinaisons de traits ne sont pas licites. Nous allons donc définir pour chaque famille de traits, ses conditions d'usage et ses contraintes d'intégrité associées.

2.2 : Sous-types de Traits autorisés selon l'étiquette syntagmatique

Tous les syntagmes peuvent potentiellement porter un trait dont les valeurs sont libres (puisqu'on ne connaît pas a priori ces valeurs). Il s'agit des traits : **Lex**, **Libre** ; les traits **Bin** sont également sans restriction d'emploi sur la catégorie du syntagme porteur : en effet, si leurs valeurs sont contraintes (+ ou -), leurs noms, eux, sont libres.

- **Héritage de Traits entre catégories**

Les catégories non terminales "héritent" des traits de certaines catégories entrant dans leur réécriture. Le tableau ci-dessous précise ces règles d'héritage pour chaque catégorie non-terminale recensée :

EtiquetteSynt : Héritage :

P Les Traits morphologiques du VERBE

SV Tous les Traits du VERBE

SN Tous les Traits du NOM

Tous les Traits de l'ADJECTIF

Tous les Traits du DETERMINANT

SP Tous les Traits du SN

Tous les Traits de la PREPOSITION

SADJ Tous les Traits de l'ADJECTIF

SADV Tous les Traits de l'ADVERBE

Dans ces cas, la valeur du trait permet, lorsqu'elle "signe une catégorie" (c'est-à-dire lorsqu'elle n'est licite que pour une catégorie), de déterminer le syntagme auquel s'applique ce trait (à savoir le porteur du trait ou un des éléments de sa réécriture).

Ex : SN[SsCatMorph : DEFINI]

s'applique au DET du SN

SN[Nombre : PLURIEL]

s'applique au SN

Si la valeur du trait peut s'appliquer à plusieurs catégories, la portée du trait reste implicite. C'est au lexicographe d'évaluer les dangers - ou les avantages - de cet implicite et de passer ou non en réécriture.

Ex : SN[SsCatMorph : CARDINAL]

deux enfants

les trois enfants

Dans les tableaux suivants de ce chapitre, on ne précisera, pour les catégories non terminales, que les Traits ou les valeurs de Traits qui ne sont pas déductibles de ces règles d'héritage.

• **Tableau des sous-types de Traits autorisés par catégorie**

Etant donné un syntagme terminal ou non-terminal identifié par une étiquette syntagmatique, seul un sous-ensemble des autres traits peut y être associé.

EtiquetteSynt Sous-Types de Traits autorisés

VERBE Mode

Temps

Personne

Genre

Nombre

Accord

Aux

Pronominal

SsCatMorph

SsCatSynt

Neg

Tournure

Passif

Aspect

NOM Genre

Nombre

Accord

SsCatMorph

SsCatSynt

Coref

ADJECTIF Personne

Genre

Nombre

NombrePosseur

Accord

SsCatMorph

SsCatSynt

ADVERBE Genre

Nombre

Accord

SsCatMorph

SsCatSynt

DETERMINANT Personne

Genre

Nombre

NombrePosseur

Accord

SsCatMorph

SsCatSynt

PRONOM Personne

Genre

Nombre

NombrePosseur

Accord

SsCatMorph

SsCatSynt

Coref

PREPOSITION SsCatMorph

SsCatSynt

CONJUNCTION SsCatMorph

INTERJECTION *Aucun trait*

PARTICULE *Aucun Trait*

P Conj

Prep

ProRel

Introd

SsCatMorph

SsCatSynt

Neg

Tournure

Passif

Coref

SN Introd

Prep

SADV Introd

e Tous les Traits sauf SsCatSynt

2.3 : Valeurs de Traits de sous-catégorie autorisées selon l'étiquette syntagmatique

Les valeurs de sous-catégorie, aussi bien morphologique que syntaxique, sont dans la dépendance de la catégorie du syntagme. En effet, une valeur de catégorie restreint l'ensemble des valeurs de sous-catégories possibles. Ceci devra être géré par des contraintes d'intégrité.

Les valeurs par défaut de **SsCatMorph** ont été définies de sorte qu'elles soient bien mutuellement exclusives lorsqu'elles sont associées à la même catégorie.

La même précaution a été prise pour les valeurs par défaut de **SsCatSynt**. L'utilisateur qui rajoute des valeurs à SsCatSynt devra prendre bien garde de respecter cette contrainte.

NB : les traits de sous-catégorie morphologique se distinguent des traits de sous-catégorie syntaxique en ceci que les traits de sous-catégorie morphologique sont des critères d'éclatement d'Um.

EtiquetteSynt Valeurs de SsCatMorph autorisées :

 verbe *Aucun Trait de SsCatMorph*

nom propre

commun

adjectif indéfini

possessif

interrogatif

cardinal

ordinal

exclamatif

qualificatif

comparatif_égalité

comparatif_supériorité

comparatif_infériorité

superlatif_supériorité

superlatif_infériorité

superlatif_absolu

adverbe comparatif_égalité

comparatif_supériorité

comparatif_infériorité

superlatif_supériorité

superlatif_infériorité

superlatif_absolu

déterminant possessif

démonstratif

partitif

indéfini

défini

exclamatif

interrogatif

cardinal

relatif

pronom personnel_fort

personnel_faible

impersonnel

indéfini

relatif

possessif

exclamatif

préposition *Aucun Trait de SsCatMorph*

conjonction coordination

subordination

interjection *Aucun Trait de SsCatMorph*

particule *Aucun Trait de SsCatMorph*

e Toutes valeurs permises

Les valeurs de traits de sous-catégorie syntaxique sont également dans la dépendance de la catégorie du syntagme. GENELEX propose un ensemble de traits de sous-catégorie, qui peut être augmenté par l'utilisateur. Sur les traits proposés, on peut émettre des contraintes d'intégrité.

EtiquetteSynt Valeurs de SsCatSynt autorisées :

verbe copule

nom déterminatif

adjectif comparatif

superlatif

adverbe temps

lieu

manière

quantité

degré

comparatif

superlatif

déterminant *Aucun Trait de SsCatSynt*

pronom *Aucun Trait de SsCatSynt*

préposition temps

lieu

manière

conjonction *Aucun Trait de SsCatSynt*

interjection *Aucun trait de SsCatSynt*

particule *Aucun Trait de SsCatSynt*

P relative

subordonnée

complétive

exclamative

coordonné

SV coordonné

SN DET_VIDE

coordonné

SADJ coordonné

SADV coordonné

e Aucun Trait de SsCatSynt

2.4 : Conjonctions possibles de Traits

Dans tous les cas, une liste de traits portée par un syntagme terminal ou non-terminal doit s'interpréter comme une "liste_et", c'est-à-dire que chaque trait de la liste ajoute une nouvelle contrainte.

Ainsi, les traits lexicaux, morphologiques, morpho-syntaxiques, syntaxico-sémantiques et sémantiques s'ajoutent les uns aux autres.

Ex : N[Lex:plupart]

[Nombre:SINGULIER]

[SsCatSynt:DETERMINATIF]

Mais tous les traits ne peuvent figurer ensemble dans la même liste.

Une liste ne peut contenir deux traits de même sous-type si les valeurs possibles pour ce sous-type sont rigoureusement exclusives les unes des autres. Ceci vaut pour tous les traits, avec les exceptions suivantes :

- Avec les traits **SsCatMorph** et **SsCatSynt**, on voudra pouvoir exprimer des restrictions du type :

SN[SsCatMorph:DEFINI][SsCatMorph:CARDINAL]

-> DEFINI s'appliquant au DETERMINANT

CARDINAL s'applique a l'ADJECTIF

- Les traits **Libre** et **Coref** ayant des valeurs non exclusives, un trait de même sous-type peut être répété. Les contraintes introduites s'ajouteront les unes aux autres. Dans le cas de Coref, on veillera cependant à ne pas utiliser des valeurs mutuellement exclusives, c'est-à-dire communes à l'un des sous-ensembles {I, NON_I}, {J, NON_J}, {K, NON_K}, ou {L, NON_L}.

Ex : SN[Coref:J][Coref:NON_I] est licite

SN[Coref:J][Coref:NON_J] ne l'est pas

- Le trait **Bin** a des valeurs mutuellement exclusives (PLUS et MOINS), mais son nom est libre. On pourra donc conjuguer des traits Bin à condition que leurs noms soient tous différents.

Ex : N[animé:MOINS][concret:PLUS]

[massif:PLUS]

Pour exprimer une alternative sur la valeur d'un trait, rappelons qu'il faut passer par l'alternative d'occupants de position.

Ex : Pi = DET[SsCatMorph:DEFINI]

DET[SsCatMorph: INDEFINI]

Ex : Pi = P[Mode: INFINITIF] [Prep: de]

P[Mode: INFINITIF] [Prep: à]

IX : Transformations

Pour ceux qui souhaitent enregistrer des transformations, le modèle offre différentes possibilités de codage de celles-ci : transformation entre Usyn, entre Descriptions et entre Syntagmes. C'est au lexicographe d'évaluer le niveau qui lui convient pour enregistrer telle transformation, et de se tenir à la même stratégie de codage pour tout son dictionnaire de manière à en assurer la cohérence.

On peut cependant donner quelques recommandations.

- Réserver les Transformations entre Usyn
 - à des emplois sémantiquement distincts pour des structures syntaxiques différentes : typiquement la tournure transitive ou intransitive des verbes neutres.

Ex : Pierre *plie* la branche

La branche *plie*

- à des Usyn provenant d'Um différentes en relation de dérivation

Ex : Les ennemis *détruisent* la ville

la *destruction* de la ville par les ennemis

- Réserver les Transformations entre Descriptions à des tournures syntaxiques différentes qui n'appellent pas de différence de sémantique fondamentale (si ce n'est le déplacement du thème d'un argument à l'autre).

Typiquement la passivation :

Ex : Jean *casse* la branche

La branche *est cassée* par Jean

- Réserver les Transformations entre Syntagmes à des structures syntaxiques identiques quant à l'ordre canonique, o• un syntagme occupant une position est le transformé d'un autre syntagme dans la même position.

Typiquement les pronominalisations :

Ex : Il aime Marie

Il l'aime

Mais ce ne sont que des pistes de travail et rien n'exclut formellement qu'un lexicographe code par exemple la neutralité entre Usyn, alors qu'un autre la codera entre Descriptions.

1 : Transformations entre Usyn

1.1 : Transformations entre Usyn provenant de la même Um

Deux Usyn provenant de la même Um peuvent être reliées par un lien de transformation nommé, qui permet de mettre en rapport dès le niveau syntaxique deux tournures distinctes

*Ex : Pierre **plie** la branche* verbe neutre

*La branche **plie***

1.2 : Transformations entre Usyn provenant d'Um différentes

Ce type de transformation correspond en fait au codage d'une dérivation syntaxique, typiquement les nominalisations. Le libellé de la transformation peut alors être par exemple "nominalisation".

Ex : coller (Verbe)

*le militant **colle** des affiches (Usyn01)*

collage (Nom)

*le militant fait du **collage** d'affiches (Usyn02)*

nominalisation

Usyn01 -----> Usyn02

De plus, les Um peuvent avoir plusieurs comportements syntaxiques. Dans ce cas, le lien de transformation mettra en relation pertinente les Usyn concernées, et permettra de mettre en évidence les différentes Usyn.

Soient deux entrées morphologiques reliées par un lien de dérivation (morphologique), ces deux entrées ayant chacune deux comportements syntaxiques :

Ex : briller (Verbe)

*Le jardin **brille** de lumière* (Usyn01)

*Cet étudiant **brille** par son intelligence* (Usyn02)

brillant (*Adjectif*)

*Le jardin est **brillant** de lumière* (Usyn03)

*Cet étudiant est **brillant**.* (Usyn04)

En reliant leurs Usyn par un lien de transformation dérivationnelle, on pourra donc mettre en parallèle :

Ex : Usyn01 -> Usyn03

- *Le jardin **brille** de lumière*

- *Le jardin est **brillant** de lumière*

Usyn02 -> Usyn04

- *L'étudiant **brille** par son intelligence*

- *Cet étudiant est **brillant***

1.3 : Calculatoire : le Mode de Transformation

Parallèlement à la description statique de l'Usyn à l'origine et de l'Usyn au résultat de la transformation, il est possible de décrire la calculatoire ayant régi cette transformation : on explicite pour cela les opérations de passage de la description de base (db) de l'Usyn origine à la description de base de l'Usyn résultat, à l'aide du Mode de Transformation.

Ce Mode de Transformation est présenté de façon détaillée en section 2.3.

2 : Transformations entre Descriptions

2.1 : Emploi et contexte d'utilisation

Une description transformée (**dt**) résulte de l'application d'une opération de transformation sur la description de base (db). Cette opération peut être formalisée par une calculatoire. Les transformations entre descriptions sont portées par l'Usyn et opèrent entre les Descriptions de cette Usyn. L'origine d'une de ces transformations peut être la description de base mais aussi une description transformée de l'Usyn, ce qui permet d'exprimer des transformations en chaîne.

Ainsi, on peut considérer que la description transformée est déductible de la description de base à laquelle on applique l'opération de transformation.

Les transformations rendent compte de trois phénomènes :

- transformation sur les réalisations de Self associées à la construction externe (IntervConst).
- transformation sur le syntagme structurel interne décrivant un composé.
- transformation sur la construction.

2.1.1 : Exemple de la passivation

Le passif est typiquement une transformation qu'on voudra représenter par ce biais. La transformation porte dans ce cas sur le Self **et** sur la construction.

Ex : Jean casse la branche

*La branche **est cassée** par Jean*

db : Self cb

Self : IntervConst : V

cb : P0 SELF P1

P0 : SN

P1 : SN

dt : Self cb

Self : IntervConst : V[Passif:PLUS]

cb : P0 SELF P1

P0 : SN

P1 : SP[Prep:par]

2.1.2 : Exemple de restructuration de complétive

Les transformations à ce niveau peuvent servir à coder des phénomènes plus complexes de restructuration de syntagmes, tels que la réduction de complétive en infinitive. Cette transformation affecte la Construction.

Ex : Ca empêche que le lait ne déborde

Ca empêche le lait de déborder

Ca l'empêche de déborder

Ca l'en empêche

**Ca en empêche le lait*

db : Self cb

Self : IntervConst : V

cb : P0 SELF P1

P0 : SN

P1 : P[SsCatSynt:COMPLETIVE]

[Mode:SUBJONCTIF]

ct : P0 SELF (P1) P2

P0 : SN

P1 : SN

PRO[Lex:le]

P2 : P[Mode:INFINITIF][Prep:de]

PRO[Lex:en]

Cond : Si P2 == PRO[Lex:en]

Alors P1 = PRO[Lex:le]

2.2 : Héritage des positions : observations

Sachant qu'une position est définie par trois champs : Fonction, RoleTh, et distribution, il y a entre une db et une dt héritage pour chacune des positions de la dt d'un sous-ensemble de ces champs :

1. On n'hérite pas de la fonction, on n'hérite pas systématiquement de la distribution (on ne mentionne ici qu'un des syntagmes élément de la distribution).

La passivation en est un exemple :

Ex : le vent casse la branche

PO[Fonction:SUJET] : SN

P1[Fonction:OBJET_DIRECT] : SN

la branche est cassée par le vent

PO[Fonction:SUJET] : SN

P1[Fonction:OBJET_INDIRECT] : SP[Prep:par]

2. Par contre, il doit y avoir héritage sur les rôles thématiques, au thème près. Toutefois la représentation choisie autorise la modification des rôles thématiques.

Ex : le vent casse la branche

PO[RoleTh:AGENT] : SN

P1[RoleTh:THEME] : SN

la branche est cassée par le vent

PO[RoleTh:THEME] : SN

P1[RoleTh:AGENT] : SP[Prep:par]

2.3 : Calculatoire : le Mode de Transformation

Qu'il s'agisse de la transformation de Self, de la transformation sur le syntagme structurel décrivant un composé ou de la transformation sur la construction, la transformation peut être décrite par un mode de transformation algorithmique.

Dans le cas d'une transformation portant sur une Construction, on pourra de plus, si on le souhaite, ne pas enregistrer la Construction résultat, et faire porter toute l'information de passage sur le mode de transformation.

2.3.1 : Explicitation du processus

Le mode de transformation consiste à

- identifier un squelette : il s'agit d'une construction dont les occupants de positions peuvent être partiellement spécifiés.

Ex : P1 : SP[Prep:par] pour un squelette de passif en "par"

- habiller ce squelette avec des éléments ou attributs provenant de la construction d'origine ou précisés dans la transformation.

Le couple squelette + habillage permet de déduire entièrement la construction résultat, qui figure explicitement dans la description transformée.

On peut exprimer la façon dont chaque position est affectée par la transformation en sélectionnant une position du "squelette" de la transformation appliquée. On parlera alors de Transformation de position. Un mode de transformation est donc un ensemble de transformations de positions. Outre les modifications de distribution, on indique également les caractéristiques de la construction modifiée par la transformation : optionnalité, solidarité, étiquette syntagmatique.

La position sélectionnée est alors "habillée" par des attributs et des éléments hérités des positions de la construction d'origine ou redéfinies par l'opération de transformation. On exprime ainsi la mise en correspondance de la position de la construction transformée (résultat) avec la position de la construction de base (origine). On sélectionne une position par référence à son rang et on lui applique les modifications voulues sur sa distribution. En particulier on précise les syntagmes hérités, inhibés, modifiés ou créés pour l'occasion.

Dans le cas d'une transformation portant sur le syntagme structurel interne d'un composé, le syntagme transformé est nécessairement totalement renseigné. Le mode de transformation décrit alors le passage du syntagme structurel origine au transformé.

Le mode de transformation de Self en tant qu'intervenant de construction indique les modifications subies par l'IntervConst pendant la transformation.

2.3.2 : Exemplification du processus

Nous illustrerons ce processus en reprenant l'exemple de passivation précédemment cité :

Ex : Jean casse la branche

La branche est cassée par Jean

db : Self cb

Self : IntervConst : V

cb : P0 SELF P1

P0 : SN

P1 : SN

dt : Selft ct

Selft : IntervConst : V[Passif:PLUS]

ct : P0 SELF P1

P0 : SN

P1 : SP[Prep:par]

Pour le Self :

- ajout du trait [Passif:PLUS]

Pour la Construction :

• sélection du squelette :

P0 SELF P1

P1 : SP

SP : P0' P1'

P0' : PREP[Lex:par]

• habillage :

P1(origine) ==> P0(resultat)

Fonction : OBJET_DIRECT ==> SUJET

RoleTh : PATIENT ==> (hérité)

Occupants : ==> (hérité)

P0(origine) ==> P1(resultat)

Fonction : SUJET ==> OBJET_INDIRECT

RoleTh : AGENT ==> (hérité)

Occupants : SN ==> SP

SP : P0' P1'

P0' : PREP[Lex:par]

P1' : SN(hérit )

3 : Transformations entre Syntagmes occupant la m me position

On veut pouvoir sp cifier, le cas  ch ant, les relations de transformation entre occupants de la m me position.

Par exemple, on veut pouvoir dire qu'une compl tive est ou n'est pas en relation de transformation avec l'infinitive, ou encore qu'un pronom ne peut  tre la pronominalisation que d'un sous-ensemble de la distribution sur la position. On met alors en relation les deux syntagmes concern s.

Ex : il aime Marie

-> *il l'aime*

il aime le chocolat

-> *il l'aime*

-> *il aime  a*

il aime dormir

-> *il aime  a*

il aime que je vienne

-> *il aime  a*

P1 : SN

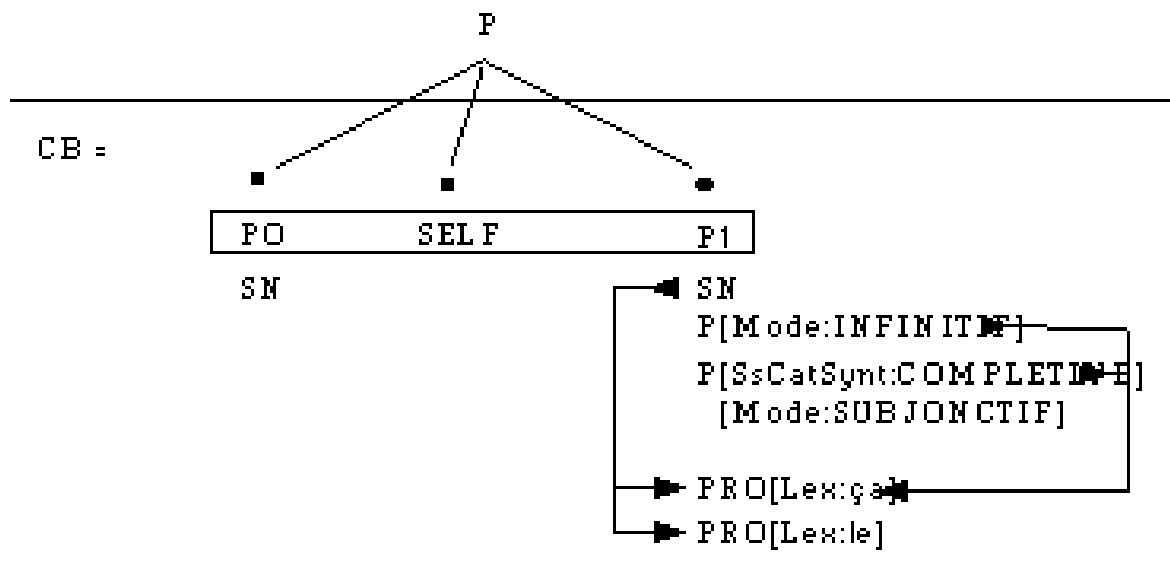
PRO[Lex:le]

PRO[Lex: a]

P[Mode:INFINITIF]

P[SsCatSynt:COMPLETIVE]

[Mode:SUBJONCTIF]



C - ANALYSE DES Unités syntaxiques composées

I : Introduction

1 : Enregistrement des composés

Les expressions composées, quel que soit leur type, sont de celles qu'un **locuteur natif** "connâit" sans avoir l'impression de les avoir jamais apprises, mais surtout **pratique** "spontanément" (avec plus ou moins de bonheur par ailleurs...). Elles sont fondamentalement de celles qu'un **locuteur étranger** doit **apprendre** pour maîtriser parfaitement la langue. Ceci vaut pour le français comme pour n'importe quelle autre langue.

Pour être connues jusqu'à l'automatisme, ces expressions devront a fortiori être données à la machine, entendez le logiciel de traitement de la langue (particulièrement en génération). Il est donc important qu'un dictionnaire électronique puisse les enregistrer.

Toutefois, enregistrer des expressions figées dans un dictionnaire ne fige en rien la **langue**, mais permet au contraire de donner le point de référence à tout ce qui est perçu comme défigement, jeu sur la langue.

2 : Dictionnaires électroniques, dictionnaires éditoriaux

Les dictionnaires électroniques sont de fait affranchis du "format papier" attaché aux dictionnaires éditoriaux et de toutes les contraintes liées à ce format. C'est un avantage notable en ce qui concerne les composés. En effet, les composés peuvent prendre une autonomie d'existence qui n'est pas sans conséquences sur :

- l'enregistrement de propriétés, puisqu'il est possible d'attacher en nombre illimité sur les composés les propriétés les plus fines et les plus complexes.
- le mode de consultation. En effet, l'accès au composé peut se faire :
 - soit par le composant plein de son choix. Le composé ne dépend donc plus, de manière plus ou moins aléatoire pour le néophyte, d'un ou de plusieurs articles de mots simples. Ainsi, dans le DNT Hachette et le Petit Robert, *mettre en Ïuvre* est trouvé à l'article de *mettre*, alors que *mettre en marche* est trouvé aux articles de *mettre* et de *marche*.
 - soit par la forme canonique, voire par une des formes fléchies du composé dans son entier.

3 : Couverture des composés dans GENELEX

On veut couvrir des formes figées de natures très différentes : locutions adverbiales (*à l'avenant*), locutions prépositionnelles (*à l'instar de*), locutions conjonctives (*au moment o•*), noms ou verbes composites (*bec de gaz*, *mettre en garde*), idiomes (*tomber des cordes*), expressions toutes faites (*glisser comme un pet sur une toile cirée*), proverbes (*Pierre qui roule n'amasse pas mousse*), maximes (*un geste vaut parfois tout un discours*) et sentences (*dis moi qui sont tes amis, je te dirai qui tu es*).

Dans GENELEX, on s'est appliqué à bien distinguer les phénomènes de sélection lexicale, collocation et composition.

La **sélection lexicale** (préposition régie, pronom sujet impersonnel contraint, etc.) est traitée en syntaxe par des traits lexicaux qui contraignent la réalisation des syntagmes occupant une position.

Ex : il est facile de critiquer il sujet impersonnel,

complément de l'adjectif :

infinitive introduite par de

Les **collocations** sont traitées dans la couche sémantique qui permet de mettre en relation de sens différentes entrées lexicales.

Ex : éluder une question relation "a_pour_objet" entre

éluder et question

meute de loups relation "a_pour_collectif" entre

loup et meute

En ce qui concerne la **composition**, on répartit les formes composées, et c'est l'originalité de notre

modèle, entre "composés morphologiques" d'une part et "composés syntaxiques" d'autre part.

Ex : sage-femme composé morphologique

bec de gaz composé syntaxique

Dans GENELEX, les "**composés morphologiques**" sont des réalisations multi-lexicales de catégories terminales qui ont une des caractéristiques suivantes :

- particularité lexicale :
 - un des composants n'apparaît que dans l'expression complexe (non nécessaire mais **suffisant**).
- particularité graphique :
 - séparateurs (tiret, ...) (non nécessaire mais **suffisant**)
- particularité flexionnelle :
 - pas de respect des règles générales d'accord entre les composants (non nécessaire mais **suffisant**).
- particularité syntaxique :
 - structuration syntaxique ne répondant pas aux règles générales dans la suite des composants (non nécessaire mais **suffisant**).

Les composés morphologiques ont donc une catégorie indépendante de la catégorie de leurs constituants et de leur éventuel regroupement syntagmatique.

Les "**composés syntaxiques**" sont aussi des éléments complexes, mais qui ont, par opposition aux composés morphologiques, les caractéristiques suivantes :

- pas de séparateurs graphiques, sauf apostrophe et blanc.
- respect des règles générales d'accord entre les composants.
- bonne formation syntaxique et éventuellement propriétés de modification, déplacement, effacement, insertion, substitution et transformation applicables.

Ce sont les composés syntaxiques et eux seuls que nous présenterons dans cette partie.

II : Caractéristiques des composés syntaxiques

1 : Syntagmes bien formés

Les "composés syntaxiques" de GENELEX sont des syntagmes bien formés (cf. synapsies de [Benveniste 74]) partiellement ou totalement contraints (cf. TAG's de [Abeillé 84]) : contraintes lexicales, morphologiques (nombre, temps, etc.), syntaxiques et sémantiques. En tant que syntagmes, ils sont sujets à différentes opérations syntaxiques régulières telles que l'accord, l'effacement, le déplacement, la modification, la transformation et la coordination.

Si leur Structure interne est décrite par un syntagme de catégorie non terminale ("catégorie interne"), leur comportement externe peut, lui en revanche, être assimilé à celui d'un syntagme terminal ("catégorie externe").

On devra décrire "l'intérieur" et "l'extérieur" du composé, bien que la frontière ne soit pas toujours facile à établir.

1.1 : Lexicalisation

• Lexicalisation totale ou partielle

Les composés syntaxiques sont des syntagmes totalement ou partiellement lexicalisés. Tous les constituants du composé syntaxique ne sont pas nécessairement lexicalisés et il peut y avoir liberté de réalisation lexicale sur un, voire même plusieurs constituants. Ces constituants libres sont donc plus ou moins nombreux et ils peuvent recouvrir indifféremment un syntagme terminal ou non-terminal.

Ex : carte bleue (lexicalisation totale) (12)

éprouver DET SADJ *admiration* (lexicalisation partielle) (15)

NB : Tous les exemples de cette partie suivis d'un numéro de référence entre parenthèses sont détaillés avec le même numéro dans la partie D (Annexe aux Unités Syntaxiques composées).

• Lexicalisation continue ou discontinue

La lexicalisation d'un syntagme peut être continue ou discontinue, de telle sorte qu'alternent éventuellement dans l'ordre linéaire canonique des constituants lexicalisés et des constituants libres.

- lexicalisation continue

Ex : carte bleue (12)

- lexicalisation discontinue

Ex : éprouver DET SADJ *admiration* (15)

N et N, •a fait deux

à N égal, N égal

Le composé ne comporte donc pas de "noyau de lexicalisation" autour duquel graviteraient des éléments libres. Ce qu'on doit décrire, c'est un syntagme dont certains constituants, n'importe lesquels, en n'importe quel nombre et en n'importe quel ordre, sont lexicalisés.

1.2 : Flexion et accord

En tant que syntagmes bien formés, les composés syntaxiques font l'objet de phénomènes de flexion et d'accord. On observe ainsi que les constituants subissent les variations morphologiques qui s'appliquent à leur catégorie : variation de personne, genre et nombre pour les déterminants ; variation de mode, temps, personne, genre et nombre pour les verbes.

Ex : Il est à l'/son aise (2)

Ils sont à l'/leur aise

Il a mis les invités à l'/leur aise

Ils ont mis l'invité à l'/son aise

1.3 : Alternatives de réalisation

La réalisation du composé n'est pas toujours considérée unique, et l'on juge parfois qu'il y a des alternatives de réalisation lexicale ou syntagmatique sur un même composé. Les variations de préposition et de verbe support (si l'on se sert des composés syntaxiques pour enregistrer les couples Vsupport-N) tombent sous cette rubrique. Sont aussi concernées, les variations sur des classes ou sous-classes lexicales définies par leur liste extensive ainsi que toutes les alternatives syntagmatiques plus complexes, bien qu'elles soient beaucoup plus rares. Qu'ils soient terminaux ou non terminaux, les syntagmes que l'on juge représenter des alternatives de réalisation s'inscrivent dans un paradigme de substitution.

- variation de la préposition

*Ex : appareil **à/de** projection [Anne Poncet-Montange, 1990]*

- variation du verbe support

*Ex : **éprouver/ressentir/avoir** de l'admiration (15)*

- classe extensive

*Ex : éprouver une **grande / immense / énorme...** admiration (15)*

*éprouver une admiration **inconsidérée***

- sous-classe extensive

Ex : en toute/parfaite connaissance de cause

- variation syntagmatique SP / SADJ

Ex : moules à la (mode) marinière (5)

moules marinière(s)

1.4 : Effacement

Les formes "éolidées" des composés sont bien connues. Elles sont liées à l'optionnalité de certains constituants qui ne sont donc pas nécessairement réalisés en surface. Bien que l'élision du nom soit la plus typique et la plus régulière, on observe ce phénomène sur n'importe quelle autre catégorie. L'effacement peut donc porter sur un spécifieur, un modifieur ou sur la tête elle-même.

- effacement de tête

Ex : fromage de chèvre

chèvre

- effacement de N tête ou enchâssé

Ex : fil de fer barbelé (3)

fil barbelé

barbelé

- effacement de spécifieur

Ex : L'argent est la cause de tous les maux

L'argent est cause de tous les maux

- effacement de modifieur

Ex : crouler sous les applaudissements de SN

crouler sous les applaudissements

1.5 : Modification

ftant structurés comme des syntagmes ordinaires, certains composés syntaxiques admettent tous les types de modification qui s'appliquent au syntagme décrivant leur structure interne. La modification peut être facultative, obligatoire, avoir des réalisations libres ou contraintes.

Ex : éprouver de l'admiration

*éprouver une **grande** / **immense** / **énorme** admiration (15)*

*éprouver une admiration **inconsidérée** / **sans borne***

Ex : en connaissance de cause

*en **toute** / **parfaite** connaissance de cause*

1.6 : Déplacement

Contrairement à ce que l'on pourrait croire en première approche, le figement induit par la lexicalisation n'implique pas forcément un figement de l'ordre linéaire de surface. En effet, on pourra observer des déplacements ou permutations de constituants entraînant des variations dans l'ordre de surface. Ceci vaut également entre intérieur et extérieur du composé. On parlera dans ce cas de phénomène de réentrance (cf. chap. sur la frontière entre intérieur et extérieur du composé).

*Ex : SN **mettre en garde** SN (10)*

*SN **mettre** SN **en garde***

En conséquence, il sera nécessaire, ici encore, de distinguer ordre canonique et ordre de surface et pour ce faire, on réutilisera très naturellement la notion de position. Les positions indiqueront l'ordre canonique. Le bouleversement de la linéarité de surface (pronominalisation, effacement, liberté des placements des SN-SP) fera partie d'un système de règles de grammaire, voire de règles stylistiques.

1.7 : Transformations

Les composés syntaxiques admettent parfois (mais pas toujours) les transformations (dont les restructurations telles que la réduction de complétive en infinitive sont un type particulier et plus complexe) qui s'appliquent au syntagme les décrivant et à leur construction. Ces transformations sont de différents types : possessivation, démonstratation, pronominalisation, relativation, passivation et même dérivation syntaxique sur un des composants.

- possessivation

*Ex : au grand dam **de** SN*

*à **son** grand dam*

- démonstratation

Ex : au moment de SN / P[Mode:INFINITIF]

à ce moment

**à son moment*

- relativation

Ex : SN livrer bataille à SN

la bataille que SN livrer à SN

SN donner libre cours à son imagination

**le libre cours que SN donner à son imagination*

- passivation

Ex : SN1 mettre en garde SN2 (10)

SN2 être mis en garde par SN1

SN prendre son pied

**son pied être pris par SN*

- dérivation sur composant tête

Ex : mettre au point [Anne Poncet-Montange, 1990]

mise au point

prendre en grippe

**la prise en grippe*

- dérivation sur autre composant

Ex : festival de musique

festival musical

machine à laver

**machine à lavage*

1.8 : Coordination

En tant que syntagmes, les composés syntaxiques peuvent être eux-mêmes coordonnés ou contenir des syntagmes coordonnés. Il faut donc pouvoir spécifier l'élément coordonnant ainsi que les éléments coordonnés.

Ex : à ses risques et périls (7)

ménager la chèvre et le chou

2 : Figement

2.1 : Figement total ou partiel

Bien qu'un composé syntaxique soit un syntagme régulier, il s'en distingue toutefois par le nombre des contraintes qu'il supporte et tout particulièrement par l'importance des contraintes sur les réalisations lexicales de ses feuilles.

Toutes ces contraintes sont des contraintes de figement, i.e. plus il y a de contraintes, plus le syntagme est figé. C'est bien en cela qu'on peut parler de **continuum de figement** qui va du totalement libre au totalement figé.

Les motivations qui président à l'enregistrement de formes composées sont d'ordres différents suivant que l'on constitue un dictionnaire de langue générale ou un dictionnaire de spécialité comme une base terminologique. Ce sont les contraintes de figement purement linguistiques qui prévalent dans le premier cas. Dans le second en revanche, ce sont des critères pratiques tels que la fréquence ou le niveau hiérarchique d'un concept qui pourront motiver l'enregistrement.

2.2 : Constituants solidaires

We distinguish two kinds of discontinuities : discontinuities that come from internal structures and discontinuities that come from the insertion of modifiers. Anne Abeillé [89]

Certains modifieurs, les appositions, les incises, les quantifieurs flottants, etc. sont des éléments qui partagent cette caractéristique de pouvoir s'insérer en de nombreuses places dans une suite de constituants et dont l'occurrence dans la forme de surface du composé n'a pas à être prévue dans le lexique.

Toutefois, dans le cas des composés syntaxiques, de telles incises ne sont pas totalement libres. Certains constituants de composés syntaxiques sont solidaires, c'est-à-dire qu'ils n'autorisent entre eux aucune insertion d'aucun type : il s'agit d'une suite insécable.

Ex : bo"te aux lettres, disait-il (16)

**bo"te, disait-il, aux lettres*

3 : Intérieur et extérieur du composé

3.1 : Assimilation à une catégorie terminale

Tandis qu'ils sont décrits par des syntagmes de catégorie non terminale, les composés syntaxiques, en tant qu'unités, peuvent être assimilés à des catégories terminales, de sorte qu'on parle de "têtes multi-lexicales".

Cette assimilation est faite sur plusieurs bases :

• syntaxique

Bien qu'une forme soit complexe, elle peut jouer le rôle de tête, appeler des compléments, restreindre son contexte d'occurrence. Par exemple, un composé structuré comme un SV se comporte dans sa construction comme un verbe et reçoit donc cette catégorie (externe) en tant qu'intervenant dans la construction.

Ex : SN mettre en garde SN contre SN (10)

V

On dira plus généralement que le composé occupe toujours la même position fonctionnelle (spécifieur du nom, modifieur de phrase, etc.) qui a pour archétype une catégorie terminale (déterminant, adverbe, etc.) à laquelle on a tendance à l'assimiler ("déterminants complexes", "syntagmes prépositionnels adverbiaux", etc.).

- SV assimilé à V (*mettre son poing sur la gueule // frapper*)
- SP assimilé à CONJ (*en conséquence // donc*)
- SP assimilé à ADJ (*sans vergogne // indélicat*)
- SP assimilé à ADV (*à ses risques et périls // imprudemment*)
- SN assimilé à DET (*la plupart des // les*)
- Nbarre assimilé à N (*fil de fer // c%oble*)

Enfin, et au même titre qu'une unité simple, le composé peut posséder en propre un ensemble de propriétés restrictives.

Ex : à ses risques et périls [Nombre:PLURIEL] (7)

RoleTh:MANIERE

• sémantique

La non-compositionnalité sémantique de certains composés syntaxiques (qui n'est cependant pas

requis pour les traiter comme unités syntaxiques composées) leur donne un statut de tout indivisible.

Bien qu'il ne soit pas toujours si facile de trancher sur la compositionnalité ou non-compositionnalité sémantique d'une expression (expressions fossiles à compositionnalité étymologique, expressions métaphoriques, sémantique des verbes supports), il est certaines expressions qui sont clairement non compositionnelles en synchronie.

Ex : SN se rendre compte de SN

V

3.2 : Distinction intérieur / extérieur d'un composé

L'étiquette de syntagme qui décrit la structuration interne du composé est en quelque sorte sa **catégorie interne**. La catégorie terminale désignant de manière privilégiée la position fonctionnelle occupée par le composé est sa **catégorie externe**. C'est sur cette distance entre les deux étiquettes que reposent les notions d'intérieur et d'extérieur du composé. L'extérieur du composé est la complémentation ou le contexte d'occurrence du composé. L'intérieur du composé est le syntagme décrivant sa structure. Lorsqu'on parle de tête multi-lexicale, on se réfère au comportement externe d'une unité dont la particularité est d'avoir un intérieur composite et non atomique.

3.3 : Frontière entre extérieur et intérieur du composé

La frontière entre intérieur et extérieur du composé n'est pas toujours facile à établir pour deux raisons :

- étant donné que le composé est un syntagme partiellement lexicalisé, certains de ses constituants sont "libres" (non lexicalisés), au même titre que la plupart de ses compléments externes. Le fait qu'un syntagme (dans la dépendance d'un composant) soit "libre" ne préjuge donc en rien de son appartenance à l'intérieur ou à l'extérieur du composé, même s'il dépend de la tête du composé.

Ex : SN éprouver DET SAdj admiration pour SN (15)

- on observe des phénomènes de réentrance (ou insertion) entre intérieur et extérieur du composé, c'est-à-dire qu'un complément **externe** peut apparaître en surface à l'**intérieur** du composé. En conséquence, l'ordre de surface ne préjuge en rien du rattachement interne ou externe d'un constituant.

Ex : SN tirer SN au clair

SN prendre SN en compte

SN mettre SN en œuvre (6)

Il s'agit donc d'une vraie difficulté lexicographique. C'est précisément au lexicographe qu'appartient de décider d'une stratégie de codage, d'en évaluer la pertinence et la portée linguistique, et enfin d'en assurer la cohérence. Le problème se pose de manière assez aiguë pour toutes les "locutions" traditionnelles : locutions prépositionnelles ou conjonctives qui peuvent être affectées par des opérations de possessivation ou démonstrativisation.

Ex : au moment o• P (11)

à ce moment

au détriment de SN (9)

à son détriment

4 : Héritage des propriétés syntaxiques des composants

Nous avons déjà eu l'occasion de préciser que les composés dont nous traitons ici (ceux qui ne sont pas figés au point d'être des composés morphologiques) sont des syntagmes réguliers présentant un certain degré de figement ; si tel n'était pas le cas, ils seraient déclarés asyntaxiques. Ces composés bien formés sont des **réalisations autorisées par les comportements syntaxiques de leurs différents composants**.

En tant que syntagmes bien formés, les composés syntaxiques obéissent donc aux règles générales de la syntaxe, de sorte que la suite des constituants libres et lexicalisés est une suite licite.

Les composants s'insèrent en conséquence dans le composé syntaxique avec leur première propriété qui est celle de catégorie.

En outre, le composé hérite du comportement syntaxique de ses composants, avec ou sans ajout de restrictions lors du processus de composition, ce dernier déterminant de surcroît l'agencement de ses composants. On dira que le composé détermine le "mode de composition" de ses composants.

En ce qui concerne l'héritage, le nombre des restrictions à apporter lors du processus de composition est parfois tel qu'on préférera ne pas hériter du comportement syntaxique d'un composant.

Chaque mode de composition décrit l'héritage par le composé des propriétés syntaxiques d'un composant. L'héritage porte sur la construction du composant ainsi que sur la catégorie et les traits restrictifs le caractérisant en tant qu'intervenant de construction. La construction est héritée à l'intérieur et éventuellement à l'extérieur du composé. Les caractéristiques du composant sont héritées à l'intérieur du composé.

4.1 : Héritage des constructions

Ces propriétés d'héritage valent pour chaque Unité Syntaxique (Usyn) entrant dans la composition

du composé.

• Héritage total versus héritage partiel

Moins le composant conserve de propriétés lorsqu'il entre en composition syntagmatique, plus le composé résultant est figé. On parle d'héritage total lorsque toutes les propriétés du composant sont héritées telles quelles. On parle d'héritage partiel dès qu'au moins une des propriétés du composant est inhibée ou filtrée. Ainsi, certaines positions du composant peuvent être complètement inhibées dans le composé, alors que d'autres seront conservées mais filtrées par sélection d'un sous-ensemble de syntagmes pouvant l'instancier : ajout et relâchement de contraintes sur ces syntagmes (déterminant qui devient défini, nombre libre restreint au pluriel, etc.) en sont des illustrations.

Ex : Il a abattu sa dernière carte ma"tresse, qui est excellente. (13)

DET SAdj **carte** SADJ/SP/P => sélection d'un sous-ensemble de

syntagmes sur le modifieur droit.

Ex : tarte à la crème (14)

DET ~~SADJ~~ **tarte** ~~SADJ/SP/P~~ => inhibition du modifieur gauche,

sélection du SP sur le modifieur droit et

restriction à la préposition à.

DET ~~SADJ~~ **crème** ~~SADJ/SP/P~~ => inhibition des modifieurs gauche

et droit, restriction au déterminant

défini.

Dans le cas d'un figement important, certains composants, voire tous, peuvent se voir privés de l'ensemble de leurs caractéristiques. C'est le cas de *bleu* dans *carte bleue* qui est privé de toutes ses possibilités de modification : *une carte très bleue* n'est plus *une carte bleue*. Dans ce cas, il ne semble pas justifié de référer à une Usyn composante qu'on inhiberait complètement. On pourra référer à l'Um (Unité morphologique).

Autre exemple, les adjectifs. Il est intéressant de remarquer que c'est l'adjectif qui détient la connaissance sur son placement à gauche ou à droite du nom dans tel ou tel emploi et que cette connaissance est héritée dans le composé.

Ex : temps plein

plein temps

temps partiel

**partiel temps*

• Héritage à l'extérieur du composé

Le composé pris comme catégorie terminale hérite dans sa construction externe de tout ou partie de la construction de base du composant qui constitue la tête du composé et d'elle seule. C'est ainsi que la construction externe de *carte bleue* est la même que celle de *carte* (d'un point de vue syntaxique, pas d'un point de vue sémantique) et pas de *bleu*.

Ex : La très belle carte bleue dorée du Crédit Lyonnais. (12)

• Héritage à l'intérieur du composé et saturation

Le composé pris comme syntagme non terminal hérite dans son intérieur des schémas de complémentation propres à chacun de ses composants Usyn. Ceci par abus de langage, car c'est plus exactement chaque composant (et non le composé) qui hérite, lorsqu'il entre en composition (dans la structure interne du composé), du schéma de complémentation lui étant associé en tant qu'unité libre.

Certaines positions d'un composant sont dites "saturées" lors de la lexicalisation par un autre composant : ces positions héritées à l'intérieur du composé ne se retrouvent pas à l'extérieur. C'est ce qu'on observe par exemple sur *abattre carte ma"tresse*. *Carte ma"tresse* sature la position objet de *abattre* à l'intérieur du composé, position qui n'est pas héritée dans sa construction externe.

Ex : SN abattre carte ma"tresse (13)

4.2 : Héritage des caractéristiques propres au composant

Les composants conservent dans le composé tout ou partie de leurs caractéristiques propres.

Ainsi, une des unités syntaxiques de *dérive* se caractérise, entre autres, par la restriction au nombre singulier. Dans le composé syntaxique *dérive génétique*, le composant *dérive* conserve cette propriété. De la même manière pour les verbes, on hérite dans le composé de l'auxiliaire qui est défini sur le composant verbal. Ainsi *abattre carte ma"tresse* hérite de l'auxiliaire *avoir* de *abattre*.

5 : Verbes supports

Les Unités Syntaxiques Composées pourront aussi être utilisées pour enregistrer - si on le souhaite - les couples "verbe-support/nom" de manière à spécifier leur schéma de complémentation, et leur faire plus facilement correspondre - le cas échéant - un V (dérivé du N ou non) qui est (à l'aspect près) leur synonyme parfait.

Ex : SN donner/poser sa démission

SN démissionner

SN mettre en marche SN

SN démarrer SN

III : Principes de la représentation formelle

1 : Axes de représentation

La nature simple ou composée d'une entrée n'affecte pas l'enregistrement d'une **Construction** spécifiant sa complémentation ou son contexte. Les unités syntaxiques composées sont donc décrites comme les unités syntaxiques simples en ce qui concerne leur comportement externe.

Ce qui leur est spécifique, c'est le fait que l'entrée décrite (**Self**) soit non pas atomique mais composite (multi-lexicale).

La représentation de la composition en syntaxe se fait par une liste de composants lexicalisés et d'informations concernant leur articulation. Celles-ci sont prises en compte par plusieurs éléments formels (**Composition**, **R_Compose**, **MdC**, **Syntagme_NT_S**) qu'il convient d'introduire dès maintenant.

Les listes de composants (**Composition** et **R_Compose**) gèrent l'aspect multi-lexical. Le Mode de Composition (**MdC**) et le Syntagme de Structure (**Syntagme_NT_S**) gèrent respectivement la régularité syntagmatique du point de vue des composants (par héritage de propriétés) et du point de vue de la grammaire (syntagmes non terminaux réécrits). L'un et l'autre peuvent coexister pour se compléter puisque les angles d'approches sont différents mais non exclusifs.

Ce qui distingue donc formellement les unités syntaxiques composées des unités syntaxiques simples, c'est seulement :

- la présence de listes de composants sur l'Usyn,
- la présence des Modes de Composition de l'Usyn composée à partir de chacune des Usyn composantes,
- la présence d'une structure interne sur le **Self** de l'Usyn.

Pour le reste, elles sont décrites par le même formalisme que les unités syntaxique simples (**Description**).

L'articulation entre le comportement syntaxique externe du composé et la description de son comportement interne se fait par le biais de l'entité **Self** présente aussi bien sur les unités simples que composées.

2 : Vision atomiste vs vision syntaxiste

Le modèle GENELEX répond au souci d'accueillir les deux appréhensions différentes du lexique que sont la vision atomiste et la vision syntaxiste (se reporter au chapitre sur les unités syntaxiques simples pour la définition de ces visions).

Cela dit, la réécriture du Syntagme de Structure ne prend sens que dans la vision syntaxiste. Dans la

vision atomiste, la spécification du syntagme décrivant l'intérieur du composé sera réduite à sa plus simple expression, c'est-à-dire à l'étiquette syntagmatique.

Par ailleurs, les Modes de Composition (**MdC**) servent à décrire et à enregistrer des héritages de propriétés à partir des composants. Ainsi définis, ils supportent les deux visions, bien que leur usage soit facultatif dans la vision syntaxiste alors qu'il est obligatoire dans la vision atomiste.

IV : Listes de composants

1 : Type et choix des composants

Les composants qui lexicalisent la structure interne des Unités Syntaxiques Composées sont explicitement recensés sur l'Usyn. Les composants sont de type Usyn ou Um. Un composant est de type Usyn lorsqu'on veut hériter, totalement ou partiellement, de son comportement syntaxique. Sinon le composant est de type Um.

La liste des composants permet l'accès au composé, puisqu'on indexe le composé sur les éléments de cette liste. Elle devra donc comprendre minimalement tous les composants qui "signent" le composé, c'est-à-dire les Usyn dont on veut hériter et des Um dites "pleines".

Pour les autres constituants lexicalisés, on peut :

- les mettre dans les composants à titre d'Um "vides" ;
- ne pas les mettre dans la liste des composants et les contraindre comme à l'accoutumée par des traits Lex ou équivalents dans le MdC et/ou la structure interne. Ceci se révèle très utile pour :
- le déterminant partitif { *du/de l', de la, des* } auquel la notion de forme canonique s'applique assez mal, étant donné l'ambiguïté des formes *du* et *de l'*. On pourra ainsi préférer :

((*éprouver admiration*)) + DET [SsCatMorph : PARTITIF]

à

((*éprouver du admiration*))

- les prépositions non régies que l'on souhaite enregistrer du côté de la structure interne et non dans la liste des composants. On pourra ainsi préférer :

((*bec gaz*)) + PREP [Lex : de]

à

((*bec de gaz*))

En tout état de cause, il revient au lexicographe d'établir une stratégie de codage et de s'y tenir.

Disposant de la liste des composants, auxquels on référerá par leur rang depuis la description de la structure et dans le mode de composition, on peut donc partager les informations ne divergeant que par des lexicalisations différentes, car elles ne sont en relation avec eux qu'indirectement par leur rang. Ainsi pourra-t-on réaliser dans un premier temps un recensement des structures (qui auront l'appellation NA, AN, NdeN, NàN, PN, VN, NN, etc. pour les composés nominaux). Puis on recensera tous les composés y répondant, c'est-à-dire tous les composés lexicalisant ces structures.

Exemple de structure interne Nom Adjectif.

RECENSEMENT DES STRUCTURES INTERNE S	
<i>Structure interne : appellation : "N_ADJ"</i>	
FO	F1
NOM[RefLex:1.1]	ADJ[RefLex:1.2]
RECENSEMENT DES COMPOSES	
((voie lacté))	
((carte bleu))	
((bête noire))	
etc ...	

2 : Alternatives de composants

Afin de pouvoir enregistrer des variantes de lexicalisation, on se sert de deux niveaux de listes.

Ex: ((avoir admiration) (15)

(éprouver admiration)

(ressentir admiration)

(vouer admiration))

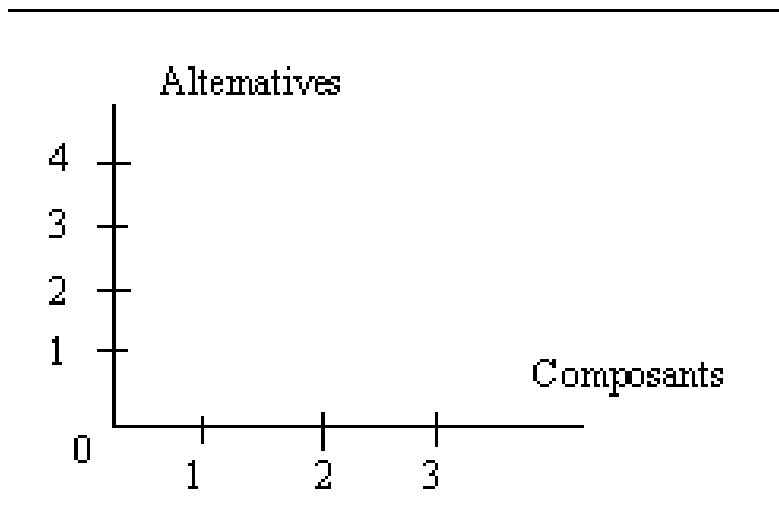
((à le cas o•) (11)

(en ce cas))

Le premier niveau de liste est celui des alternatives de lexicalisation (**Composition**).

Le deuxième niveau de liste est celui des composants lexicalisés pour chaque alternative (**R_Compose**).

On peut voir ces listes sous la forme de deux axes orthogonaux.



3 : Référence aux composants

On accède aux éléments des listes de composants par une double indexation : [RefLex : Nième alternative . Nième composant].

Ex : [RefLex:1.1]--->avoir

L'indice 0 sur l'axe des alternatives signifie qu'elles sont toutes sélectionnées.

Ex : [RefLex:0.1]--->avoir , éprouver ,

ressentir , vouer

NB : Dans les exemplifications en annexe, on prend le parti d'indiquer explicitement la lexicalisation afin de garantir une certaine lisibilité : [RefLex:mot], plutôt que d'utiliser la notation : [RefLex : Nième alternative . Nième composant]. Ceci vaut pour toute la famille des RefLex et inclut RefPrep, RefConj, etc.

V : Mode de Composition (MdC)

1 : Intérêt

Les Modes de Composition permettent de spécifier pour chaque composant Usyn quelles sont les propriétés de sa Construction et de son Self (IntervConst et Syntagme_NT_S) qui sont inhibées, filtrées ou héritées lors du processus de composition. Ils permettent aussi d'indiquer quels autres composants (Um ou Usyn) lexicalisent éventuellement leurs Positions. Il y a donc, pour une Usyn composée donnée, autant de MdC que de composants Usyn.

Spécifier le mode de composition sert donc à indiquer :

- les **relations** entre les Usyn et/ou Um composantes,
- l'**héritage** de Positions,
- les **ajouts** ou **rel%ochements** de contraintes sur les occupants de Position et Self.

2 : Inhibition, héritage et filtrage

2.1 : Inhibition, héritage et filtrage des Positions

Par défaut, toutes les Positions d'un composant qui ne sont pas explicitement héritées sont inhibées. Pour chaque Position héritée, on doit spécifier comment et où on en hérite, c'est-à-dire avec ou sans filtrage, en direction de l'INTERIEUR ou l'EXTERIEUR du composé. Une même Position, alors non "saturée" dans le composé, peut être héritée à l'extérieur **et** à l'intérieur du composé. Dans ce cas, on lui associera deux héritages de Position.

Filtrer une Position lors de l'héritage consiste à modifier au moins un élément parmi sa distribution, sa fonction, ses rôles thématiques ou sa valeur de répétabilité. On modifie la distribution par sélection d'un sous-ensemble des syntagmes listés ou par retrait/ajout de contraintes sur ces syntagmes (traits restrictifs).

2.2 : Filtrage de Self

Par défaut, les Self des Usyn composantes sont hérités à l'intérieur du composé sachant qu'ils perdent alors l'appellation "Self" pour devenir un syntagme tel que spécifié dans l'IntervConst. Il est possible de filtrer Self dans le processus de composition et de modifier, toujours par les mêmes mécanismes, son IntervConst et sa Structure interne si le composant est lui-même un composé.

2.3 : Exemplification

Ex : la vraie tarte à la crème du cinéma fran•ais (14)

Usyn_tarte :

cb : P0 P1* SELF P2*

DET SADJ SADJ

SP

P[SsCatSynt:RELATIVE]

MdC_tarte dans *tarte à la crème* :

HeritePosition : P0 EXTERIEUR

HeritePosition : P1 EXTERIEUR

HeritePosition : P2 EXTERIEUR

HeritePosition : P2 INTERIEUR

ModifPosition :

SADJ, P[SsCatSynt:RELATIVE]

SP + [Prep:à]

[RefLex:crème]

[Nombre:SINGULIER]

[SsCatMorph:DEFINI]

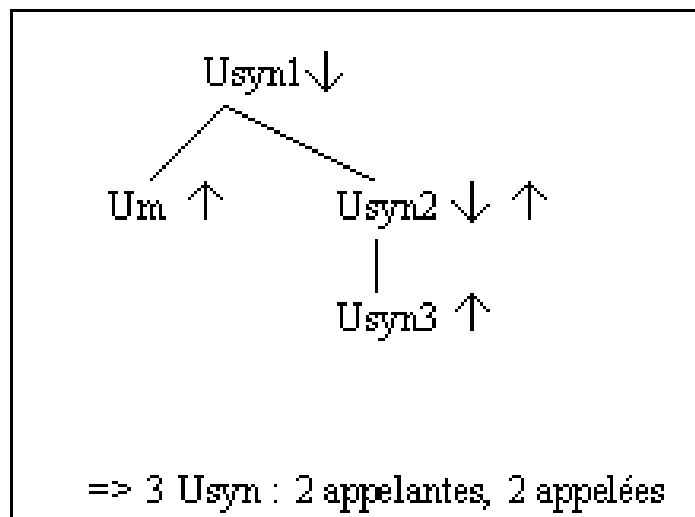
FiltreSelf

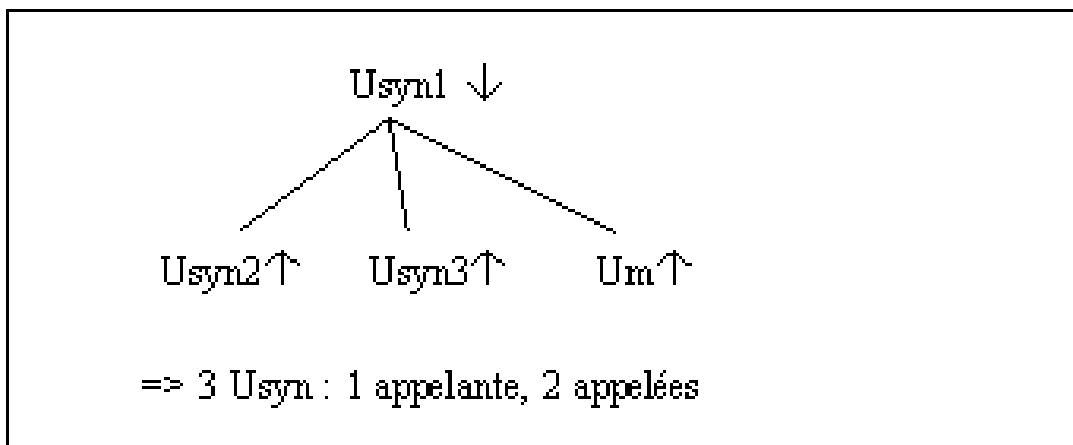
IntervConst : N + [Nombre:SINGULIER]

3 : Appels des composants pour lexicalisation

3.1 : Notion d'appel

L'agencement des composants, - i.e. le fait qu'ils s'organisent suivant des niveaux hiérarchiques différents et que certains soient sous la dépendance d'autres - est analysé comme un phénomène d'**appel**. On dit d'un composant qu'il est appelé s'il instancie une Position d'un autre composant. On dit d'un composant qu'il est appelant si au moins l'une de ses Positions est instanciée par un autre composant. Un même composant peut être appelant, appelé et appelant, ou seulement appelé. Cette information est portée sur les éléments **R_Compose** qui donnent la liste des composants.





Illustrons ces arbres d'appels abstraits par un exemple.

Ex : abattre carte maîtresse (13)

2 Usyn appelantes : Usyn_abattre, Usyn_carte

1 Usyn appelée : Usyn_carte

1 Um appelée : Um_ma"tre

3.2 : Tête du composé syntaxique

La tête de l'Usyn composée est l'Usyn composante la plus appelante.

Dans l'exemple précédent (*abattre carte ma"tresse*), l'Usyn "tête" du composé est l'Usyn_abattre

Dans le cas des composés coordonnés, on pourra considérer que la tête du composé est l'élément coordonnant et que c'est lui qui appelle tous les autres constituants.

Ex : à cor et à cri

3.3 : Lexicalisation

Un appelé réalise une contrainte lexicale. L'appel d'un composant Um ou Usyn est exprimé par l'ajout d'un trait **RefLex** sur un Syntagme d'une Position de la Construction de base de l'**Usyn appelante**.

Ex : SN[RefLex:admiration]

Le composant donne sa valeur au trait lexical qui l'appelle (RefLex, RefPrep, RefProRel, RefProIntrog, RefConj, RefIntrod) et les règles générales propres à ces traits s'appliquent (cf. \propto sur les traits restrictifs). L'attribut *saturesynt* des traits lexicaux permet d'indiquer si la tête est (avec un éventuel introducteur) la seule feuille du Syntagme.

3.4 : Compatibilité entre les catégories du syntagme appelant et du composant appelé

Il faut assurer la compatibilité entre l'étiquette du syntagme appelant (celui qui porte le trait RefLex) et la catégorie morphologique ou fonctionnelle du composant appelé (de type **Um** ou **Usyn**). Ceci revient à dire que le composant doit pouvoir être la tête du syntagme appelant. On reprend des unités simples les valeurs de tête autorisées en fonction du syntagme porteur du trait lexical :

- NOM pour SN, Nbarre ou SP
- ADJECTIF pour SADJ
- ADVERBE pour SADV
- VERBE pour P ou SV

La catégorie morphologique du composant est donnée par le champ *catgram* de l'**Um**, alors que sa catégorie fonctionnelle est donnée par les étiquettes des syntagmes de l'**IntervConst** de l'**Usyn**. Si le composant est de type **Usyn**, et étant donné que chaque **Usyn** simple est associée à une et une seule **Um**, on peut toujours retrouver l'**Um** à partir de l'**Usyn**.

3.5 : Unification

Le Syntagme qui porte le trait de lexicalisation peut aussi porter d'autres traits restrictifs. Ceci vaut quelle que soit la nature **Um** ou **Usyn** de l'élément lexicalisant. Toutes les restrictions émises sur une **Usyn** appelée qui seraient remontées sur le syntagme appelant par un mécanisme d'unification doivent être remontées "à la main" par le lexicographe, notre formalisme ne gérant pas ce mécanisme.

Ex : éprouver de l'admiration

Usyn_éprouver :

cb : P0 SELF P1 (P2)

SN SN SP[Prep:pour]

Usyn_admiration :

cb : P0 P1* SELF P2*

DET SADJ SADJ

SP

P[SsCatSynt:RELATIVE]

MdC_éprouver dans *éprouver de l'admiration* :

HeritePosition : P0 EXTERIEUR

HeritePosition : P2 EXTERIEUR

HeritePosition : P1 INTERIEUR

ModifPosition :

SN + [RefLex:admiration]

[Nombre : SINGULIER]

MdC_admiration dans *éprouver de l'admiration* :

HeritePosition : P0 INTERIEUR

HeritePosition : P1 INTERIEUR

HeritePosition : P2 INTERIEUR

FiltreSelf

IntervConst : N + **[Nombre : SINGULIER]**

Enfin, on considère en vision syntaxiste (avec arbres de réécriture) que pour une Usyn composante, le niveau qui entre en composition est a priori le niveau du Self. Dans le cas où l'on souhaite récupérer des contraintes sur le contexte syntaxique de Self (plus large que son strict schéma de complémentation), il faut l'indiquer explicitement par l'héritage des positions rendant compte de ce contexte.

VI : Structure interne du composé

1 : Choix de la catégorie

La catégorie interne du composé est portée par le **Syntaxme_NT_S** décrivant la Structure interne de **Self**. La catégorie externe du composé est portée par le(s) **Syntaxme(s)** de son **IntervConst**.

Ex : bo"te aux lettres (16)

(catégorie interne Nbarre , catégorie externe N)

Les catégories interne et externe doivent être compatibles. Voici à titre indicatif les couples que l'on a pu identifier au fil du travail dans GENELEX.

Catégorie externe

Catégorie interne

NOM	Nbarre, SN
ADJECTIF	SADJ, Nbarre, SP
ADVERBE	SADV, SP
VERBE	SV
PREPOSITION	SANS_E
CONJONCTION	SANS_E, SP
INTERJECTION	P
DETERMINANT	SANS_E, SN
PRONOM	?
PARTICULE	?
SANS_C	SANS_E

2 : Syntagmes de Structure (Syntagme_NT_S)

Dans le modèle GENELEX, la Structure **interne** de base des composés est décrite dans **Self** par un "syntagme non terminal de structure" (**Syntagme_NT_S**). Ce syntagme se caractérise par son étiquette, les traits qui lui sont associés et il peut être réécrit par une liste de positions, au même titre que tous les syntagmes non terminaux.

Ex : la plupart structure interne SN

tarte à la crème structure interne Nbarre

pierre qui roule n'amasse pas mousse structure interne P

mettre en ïuvre structure interne SV

en connaissance de cause structure interne SP

sain et sauf structure interne SADJ

plus ou moins structure interne SADV

Ces syntagmes sont parfaitement identiques aux Syntagmes non terminaux apparaissant dans les Positions décrivant les unités simples à ceci près que :

- ils ne contiennent pas d'élément Self, puisque c'est précisément lui que l'on est en train de décrire à travers eux.
- ils autorisent des traits de la famille RefLex sur les Syntagmes apparaissant dans leurs Positions,
- à la liste de Positions s'ajoute une éventuelle liste d'Insertions.

Pour le reste (réécriture d'arbre, étiquetage des noeuds, coordination, etc.), on se référera au paragraphe sur les Syntagmes dans le chapitre sur les unités syntaxiques simples.

Il est à noter que les composés syntaxiques ont une structure interne particulièrement sujette aux contraintes structurelles. Les contraintes structurelles sont des contraintes de réécriture sur un syntagme. Pour certains composés syntaxiques, un élément normalement facultatif dans ce syntagme sera obligatoirement réalisé ou omis ; ou encore, une structuration complexe, telle que la coordination ou l'enchâssement de SN, sera obligatoirement appelée.

Ex : à ses risques et périls coordination (7)

sans conteste absence de déterminant

au détriment de SN enchâssement (9)

en lieu et place coordination et absence de

déterminant

3 : Insertions

Les phénomènes de réentrance sont représentés par des Insertions de Positions de la Construction de base dans la Structure interne du composé.

Ex : mettre SN en marche

prendre SN en compte

mettre SN en Œuvre (6)

Dès qu'une Insertion est notée, il faut interpréter les Positions de l'intérieur comme des places, ce qui est en contradiction avec l'enregistrement d'un ordre canonique. Les Insertions font donc entorse à la distinction place/position (ordre linéaire de surface et ordre linéaire canonique. Voir unités simples). En admettant que l'ordre canonique ne soit qu'un des ordres de surface possibles, il devient, avec les Insertions, le seul possible.

Les insertions de **Positions** de la Construction externe sont notées à leur place dans le champ *insereinsertion_l* au milieu des **Positions** du **Syntagme_NT_S** figurant la Structure interne. Une **Insertion** est donc à la fois un pointeur vers une **Position** de la **Construction** externe et une marque du point d'insertion. Il peut y avoir plusieurs **Insertions**. Tout comme les **Positions**, les **Insertions** sont numérotées à partir de zéro.

Certaines **Insertions** sont obligatoires, alors qu'elles sont seulement possibles dans la plupart des cas.

- insertion possible

Ex : SN mettre SN en garde (10)

SN *mettre en garde* SN

- insertion obligatoire

Ex : SN connaître SN comme si PRO l'avoir fait

**SN connaître comme si PRO l'avoir fait SN*

4 : Solidarité

Les éléments qui peuvent être solidaires, c'est-à-dire constituer une suite insécable ne supportant ni incise, ni apposition, ni quantifieur flottant sont **Position, Self, Insertion**.

Cette suite insécable est représentée dans un champ *solidarité* à l'intérieur du composé par une chaîne graphique avec un tiret entre chaque paire d'éléments solidaires. On pourra donc avoir un champ de solidarité valant :

P0 I0-I2-P1

5 : Traits restrictifs

Tous les traits restrictifs définis pour les Unités simples sont disponibles pour les Unités composées. Certains d'entre eux appellent cependant des commentaires.

5.1 : Traits lexicaux

La lexicalisation des Syntagmes terminaux ou non terminaux entrant dans la description du composé syntaxique est exprimée au travers de traits lexicaux portés par lesdits Syntagmes. On pourra réutiliser tous les traits définis pour les unités simples, à savoir Lex, Prep, Conj, ProRel, ProIntro, Introd. On dispose de plus en miroir des traits RefLex, RefPrep, RefConj, RefProRel, RefProIntro, RefIntrod. Ces traits "Ref" ont exactement la même sémantique que leurs correspondants mais pointent sur des Unités Morphologiques (Um) ou des Unités Syntaxiques (Usyn) non pas directement, mais indirectement par des numéros de rang dans des listes de composants (cf. chap. sur les listes de composants). Cette indirection a été introduite de manière à pouvoir partager les structures internes (N à N, N de N, etc.) indépendamment de leur lexicalisation effective. Les traits RefLex ne sont portés que par des syntagmes de la structure interne et sont exclus de la construction externe.

Ex : voie lactée

carte bleue (12)

cerveau malade

N[RefLex:1.1] ADJ[RefLex:1.2]

Rappel : Dans les annexes à ce chapitre, les valeurs des traits RefLex sont résolues pour des raisons de lisibilité. On trouvera donc, par abus, l'information sous la forme N[RefLex:voie] ADJ[RefLex:lacté].

Il va sans dire qu'on ne peut pointer sur un composant par un trait de la famille RefLex que s'il a effectivement été enregistré comme "composant" à part entière dans la liste des composants.

En revanche, il n'est pas obligatoire de traiter tous les éléments de lexicalisation dans la liste des composants. On peut notamment si on le souhaite spécifier les prépositions et les déterminants dans la structure interne du composé et non dans sa liste de composants. Dans ce cas, les spécifications sont faites comme pour les unités simples au moyen de traits de la famille "Lex" (ex : SP[Prep:à]) ou bien au moyen de traits de sous-catégorisation (ex : DET[SsCatMorph:PARTITIF]).

5.2 : Traits morphologiques

Tous les traits morphologiques définis pour les Unités Syntaxiques simples sont disponibles pour la description des composés. Ils permettent, par exemple, d'exprimer dans la structure interne n'importe quelle restriction morphologique sur les composants, la restriction la plus courante étant celle de nombre.

Ex : prendre son pied => forme composée (1)

prendre ses pieds => forme libre

5.3 : Traits Accord et Coref

Les phénomènes d'accord au sens large sont traités exactement comme pour les unités simples. Les accords réguliers sont relégués à la grammaire ; les accords dépendants de coréférences sont gérés par l'introduction de traits Coref aux endroits appropriés ; les accords irréguliers sont contraints par l'introduction d'un trait Accord.

On veut pouvoir exprimer des coréférences et des accords entre des Syntagmes à l'intérieur et des Syntagmes à l'extérieur du composé. En conséquence, le domaine de résolution des traits Accord et Coref est l'ensemble de la Description de l'Usyn composée (Construction externe + IntervConst et Syntagme de Structure du Self).

Ex : Il est à son aise (2)

Ils sont à leur aise

Les variations plus que les contraintes de forme des pronoms personnels et déterminants posent généralement problème aux lexicographes lors de l'enregistrement des composés. Dans GENELEX, le regroupement lemmatique des formes *{je, tu, il, elle, nous, vous, ils, elles}*, *{ma, ta, sa, mon, ton, son, mes, tes, ses, notre, votre, nos, vos, leur, leurs}*, considérées comme des formes fléchies, permet de s'affranchir du problème.

Dans les cas de composition, on considérera les variations de personne, genre et nombre comme des variations flexionnelles sur un seul et même lemme entrant dans la composition du composé et on les traitera comme un simple phénomène d'accord.

A l'inverse, on peut bien sûr émettre, si besoin est, des restrictions morphologiques sur ces lemmes par le biais des traits morphologiques sur les syntagmes qu'ils lexicalisent.

5.4 : Traits sémantiques

On retrouve pour les composés comme pour les compléments des mots simples, l'ensemble des restrictions sémantiques à disposition (pour ceux qui le veulent seulement) dès la syntaxe :

- contrainte aspectuelle sur le verbe

Ex : être [Aspect : STATIF] à l'/son aise (2)

mettre [Aspect : PROCESSIF] à l'/son aise

- contrainte de classe sémantique

Ex : pour les beaux yeux de SN [humain : PLUS]

6 : Conditions

Comme pour les unités syntaxiques simples, il est possible de conditionner l'optionnalité et l'occupation d'une **Position**. Mais les **Positions** mentionnées dans le **Si** ou le **Alors** des **Conditions** peuvent appartenir non seulement à la **Construction** externe mais aussi à la structure interne du composé (si toutefois son **Syntagme_NT_S** a été réécrit). On pourra ainsi conditionner la réalisation en surface des compléments (ou éléments de contexte) du composé mais aussi des constituants du composé lui-même.

On note plusieurs cas de figure pour les **Conditions** des composés syntaxiques :

- Positions appartenant uniquement à la Construction externe

Voir exemples des unités syntaxiques simples

- Positions appartenant uniquement à la Structure Interne

Ex : au détriment de SN (9)

à son détriment

- o• la réalisation du génitif (champ *destination* INTERNE)

dépend de la sous-catégorie (défini ou possessif) du

déterminant (*destination* INTERNE).

- Positions appartenant aux deux

Ex : vouer/éprouver/... admiration à/pour (15)

- o• la préposition régie (EXTERNE) varie selon le verbe composant (INTERNE)

Les Conditions sont portées par la Description qui regroupe Self et Construction. Pour atteindre la Structure interne de Self dans les Conditions, on leur donne la portée INTERNE ; pour atteindre la Construction externe du Self composé, on leur donne la portée EXTERNE. Après cela, les CheminPosition et CheminSyntagme permettent de naviguer et de pointer sur n'importe quels Position ou Syntagme de la Structure interne ou de la Construction externe.

7 : Transformations

En tant qu'unités, les composés syntaxiques ont une **Construction** de base qui peut ou non faire l'objet de **Transformations**. Sur ce point, on pourra se reporter directement aux transformations sur les unités syntaxiques simples.

Mais les unités syntaxiques composées ont cette caractéristique de pouvoir se transformer sur elles-mêmes, dépendamment ou indépendamment de leur Construction, c'est-à-dire que la structure interne du composé peut se trouver altérée lors d'un processus de **Transformation**.

- **Construction** et **Syntagme_NT_S** de **Self** modifiés lors de la passivation

Ex : SN1 mettre en garde SN2 (10)

SN2 être mis en garde par SN1

- **Syntagme_NT_S** seul modifié lors de la possessivation

Ex : au détriment de SN (9)

à son détriment

Une Structure interne étant modélisée par un **Syntagme_NT_S**, tous les mécanismes de Transformation définis pour les Syntagmes des unités syntaxiques simples s'appliquent aux transformations du composé lui-même. De ce fait, les transformations peuvent ici aussi être décrites comme des opérations de calcul régulières. En effet, le passage par une arborescence apporte une certaine abstraction qui permet de coder de la même manière (parce qu'on opère sur des nœuds non terminaux) une même transformation qui s'appliquerait à des composés syntaxiques légèrement différents en surface.

VII : Construction

Les composés sont d'une manière générale (qu'ils soient composés morphologiques ou syntaxiques) des unités à réalisation lexicale complexe. En tant qu'unités, il faut décrire leur fonctionnement en spécifiant notamment leur schéma de complémentation ou leur contexte d'occurrence. De ce point de vue, ils se comportent exactement comme des unités simples et tous les mécanismes qui ont été définis pour celles-ci s'appliquent.

Dans la Construction, on réfère au composé par Self. C'est-à-dire qu'on procède exactement comme pour les unités simples, si ce n'est que dans la composition, Self donne en outre la Structure Interne de l'unité prise comme un tout dans la Construction de base.

• Composés SV en position fonctionnelle de VERBE

Les verbes composés doivent être décrits comme les verbes simples en spécifiant l'ensemble de leurs compléments et leur position sujet.

Ex : SN mettre en garde SN (10)

• Composés nominaux en position fonctionnelle de NOM

Définis comme étant en position N, ils ont par là-même un schéma de complémentation identique à un N simple : Déterminant, Modifieur-gauche, Modifieur-droit.

Ex : une bo"te aux lettres rouge (16)

une armoire à glace cultivée

une femme à poigne sans poigne

une énorme turbine à hélice

une rinceuse à bouteilles qui ne servira jamais

• Composés nominaux en position fonctionnelle de DETERMINANT

Ceux qui voudront traiter les déterminants complexes comme tels dès le niveau syntaxique pourront le faire en les associant à une position Déterminant et en spécifiant leur contexte d'occurrence (c'est à dire en l'analysant).

Ex : la plupart des gens (17)

NB : notre propos n'est pas de discuter ici l'analyse particulière ou prépositionnelle de ce *de* quantitatif, sachant que GENELEX permet d'enregistrer l'une ou l'autre indifféremment.

On devra toutefois garantir un minimum de cohérence avec l'analyse qui en est faite pour les mots simples (*beaucoup* dans l'exemple suivant).

Ex : beaucoup de gens

• Composés SP en position fonctionnelle de CONJONCTION

Certains SP servent de conjonction et introduisent des phrases entières. On veut pouvoir préciser le type des phrases introduites et leur mode.

Ex : En conséquence il n'est pas venu. (18)

• Composés SP en position fonctionnelle d'ADVERBE

S'appliquer à décrire que tel SP figé (ou partiellement figé) occupe une fonction adverbiale permet de définir la portée du SP et de lever des ambiguïtés quant au rattachement du groupe prépositionnel. On sait combien ce problème est épineux en analyse syntaxique.

Ex : Il empruntera cette voie à ses risques et périls. (7)

Dans cet exemple, le SP est un adverbe de phrase. Son "complément" est donc une phrase.

• Composés SP en position fonctionnelle d'ADJECTIF

Utilisés comme adjectifs, il est intéressant de savoir pour ces composés - au même titre que pour les adjectifs simples - s'ils peuvent être épithètes gauches, droits, attributs et s'ils prennent ou non des compléments.

Ex : Il est sans vergogne

Un garçon sans vergogne a déboulé dans la pièce

• Usyn composées sans complémentation externe

La dichotomie intérieur/extérieur du composé est fondée sur l'assimilation du composé à une catégorie terminale.

Si une unité complexe ne fonctionne pas comme catégorie terminale, alors elle n'a pas de comportement externe. Toute sa description est faite dans sa structure interne. C'est notamment le cas des phrases figées.

Ex: pierre qui roule n'amasse pas mousse

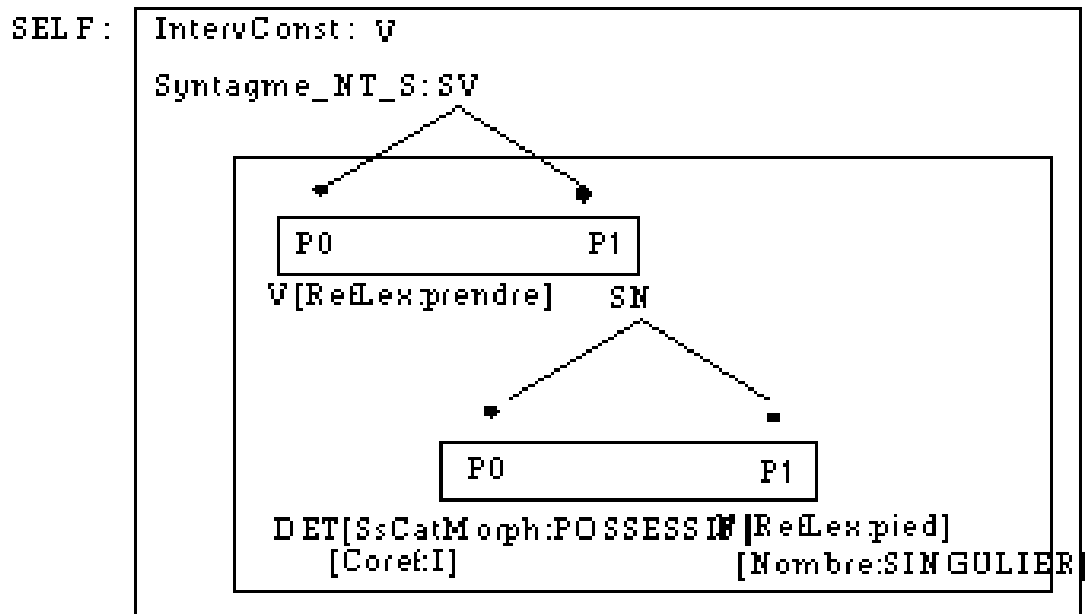
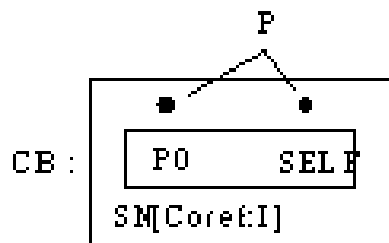
D - ANNEXE AUX Unités syntaxiques composées

AVERTISSEMENT

Les exemples de codage qui sont présentés ici correspondent à des **choix lexicographiques particuliers**, d'autres choix étant généralement possibles tout en restant dans le cadre du modèle GENELEX. Tant la structuration choisie pour les arbres syntagmatiques que le découpage en unités syntaxiques, la description des dites unités, le recours à des conditions associées ou encore le regroupement d'alternatives de lexicalisation ne prétendent pas imposer le codage de l'exemple, mais seulement illustrer comment, une fois certains choix lexicographiques réalisés, le modèle GENELEX permet d'en rendre compte.

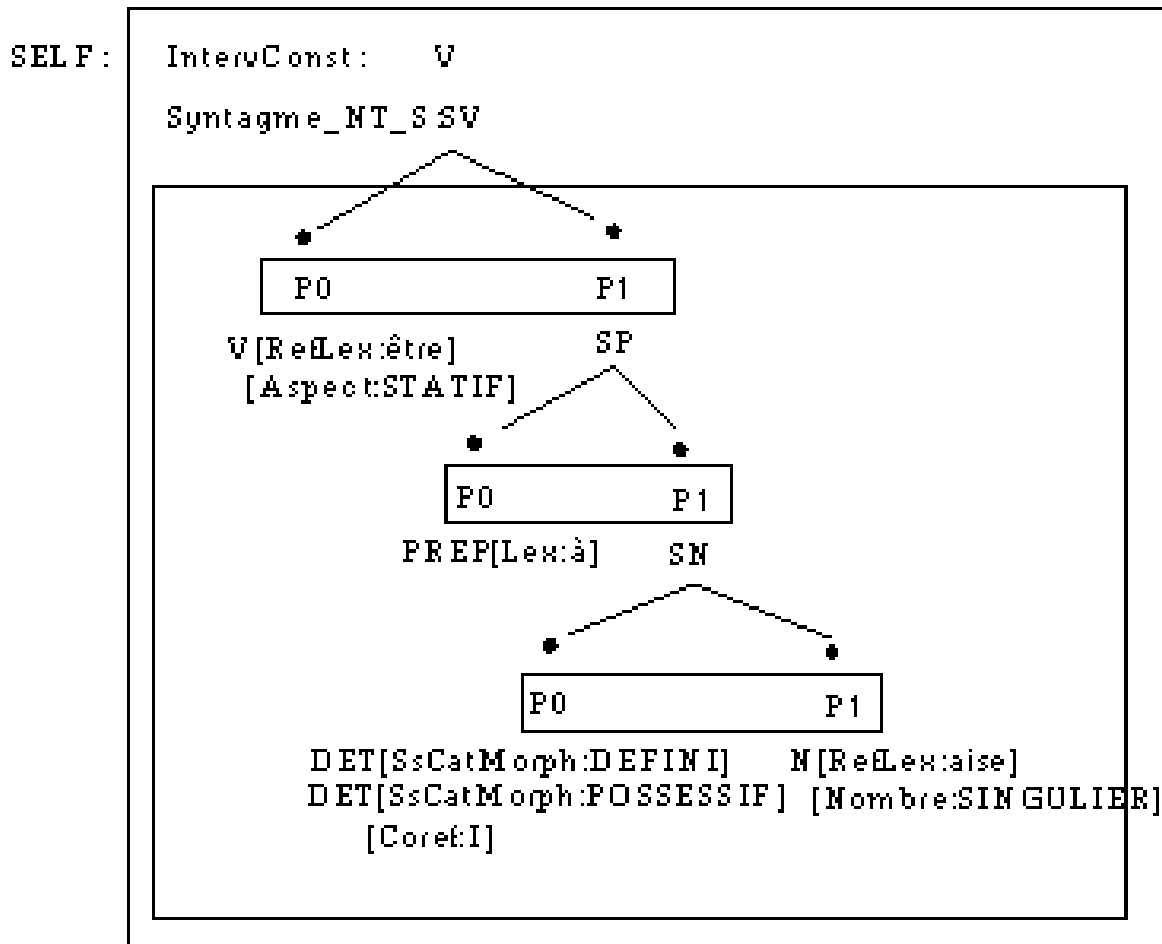
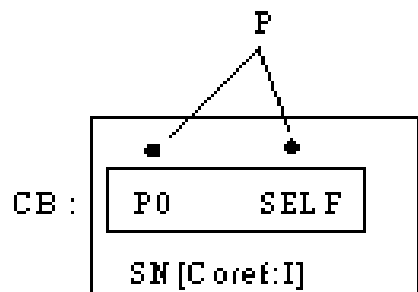
Il est donc très important de lire ces exemples dans cette optique et de ne pas oublier que la mise en oeuvre du modèle GENELEX passe d'abord par l'élaboration d'une stratégie lexicographique de codage, qui précise les critères de découpage en unités syntaxiques, les structurations de syntagmes utilisées et les regroupements à effectuer.

*Exemple (1) : SN **prendre son pied***



Exemple (2) : Il est à l'/son aise

Ils sont à l'/leur aise



Exemple (3) : fil de fer barbelé

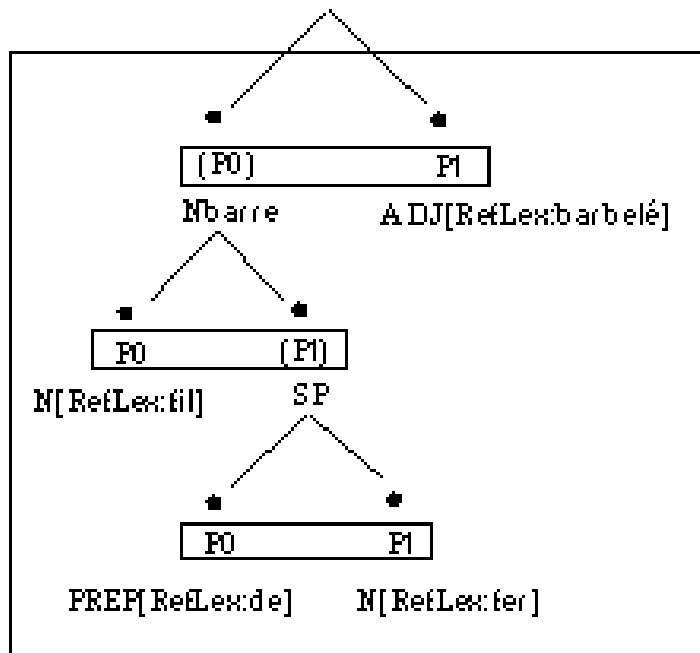
fil barbelé

barbelé

SELF :

InteruConst : N

Syntagme_NT_S : Nbarre



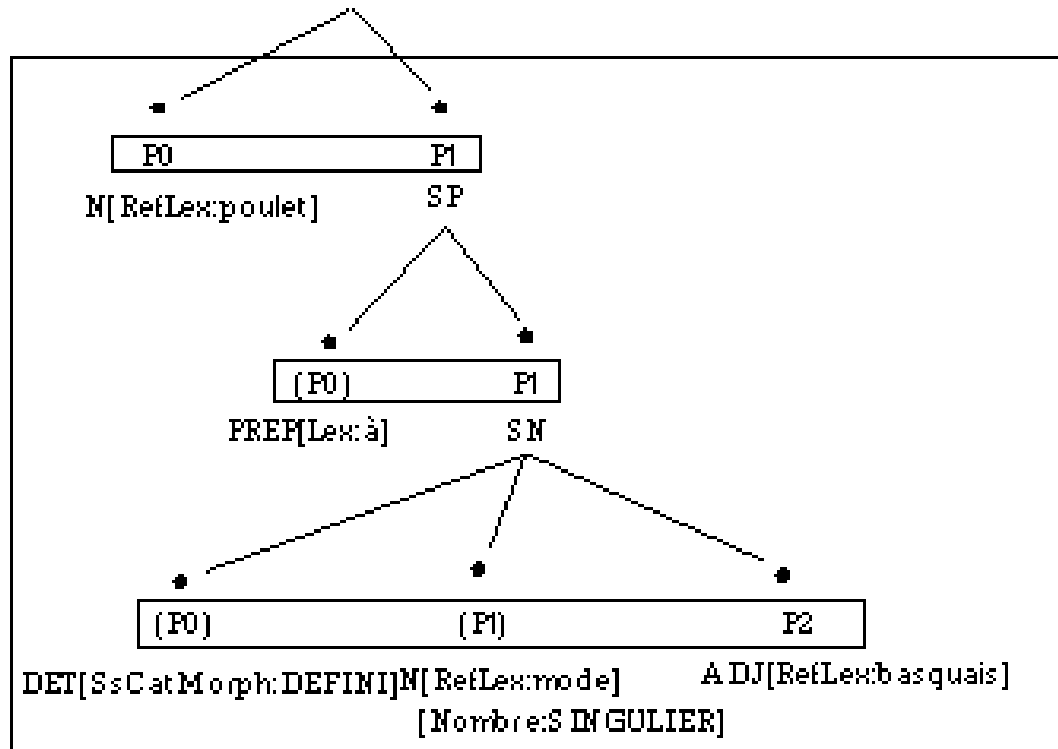
Exemple (4) : poulet à la mode basquaise

poulet à la basquaise

poulet basquaise

SELF : IntervConst : N

Syntaxme_NT_S : Mbarre



Cond : Si INTERNE ! F1.SP.FO Alors INTERNE ! F1.SP.F1.SN.FO & ! F1.SP.F1.SN.F1
 Cond : Si INTERNE ! F1.SP.F1.SN.FO Alors INTERNE ! F1.SP.FO & ! F1.SP.F1.SN.F1

Exemple (5) : moules à la mode marinière

moules à la marinière

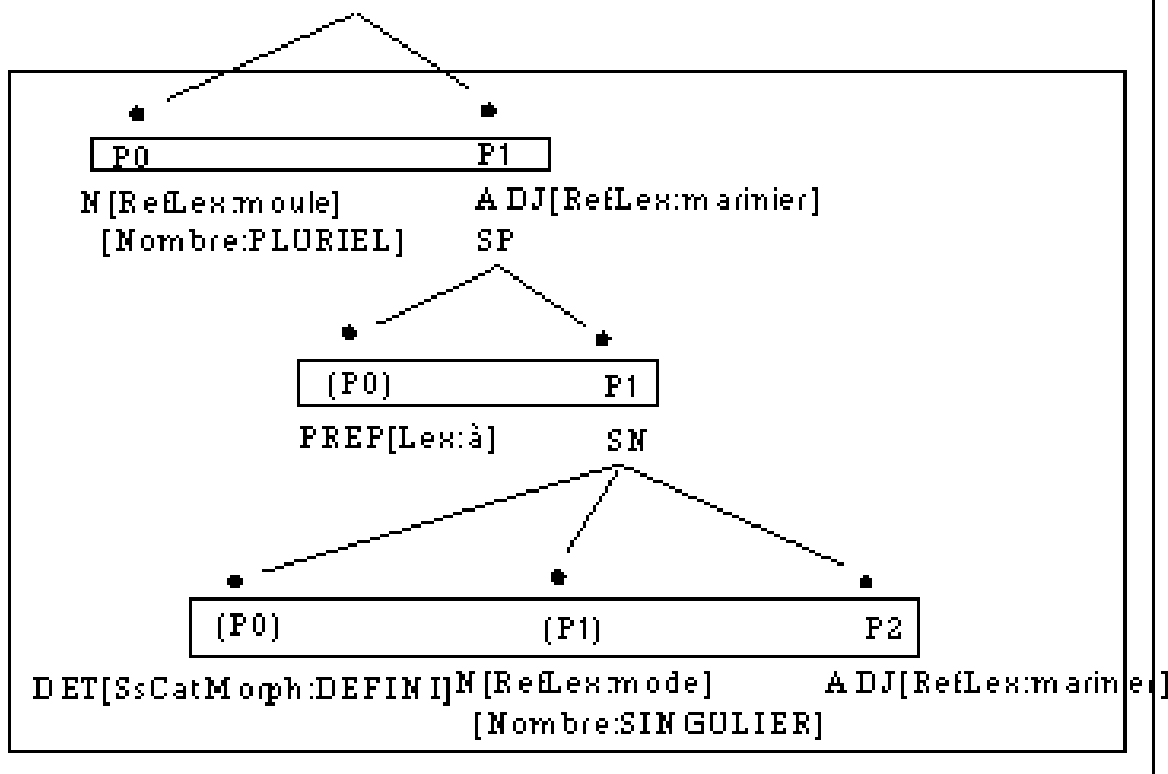
moules marinière

moules marinières

SELF :

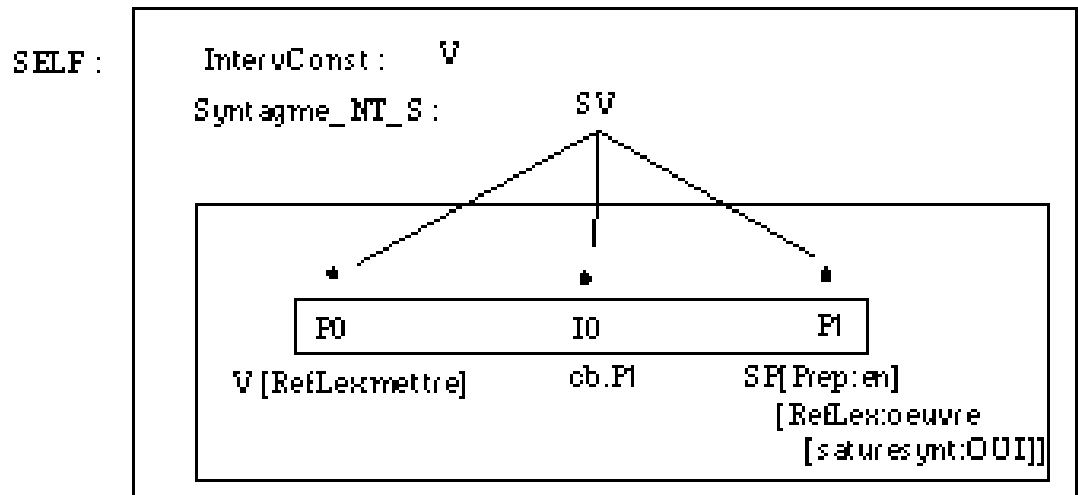
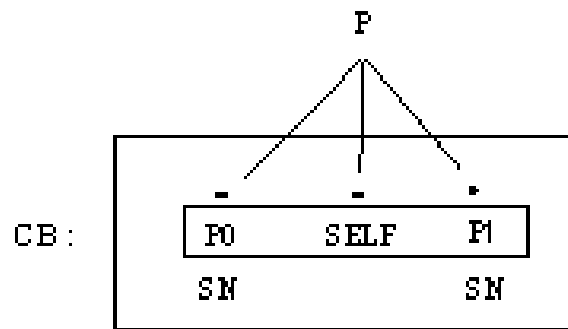
InterConst: N[Nombre:PLURIEL]

Syntagme_MT_S: Nbarre

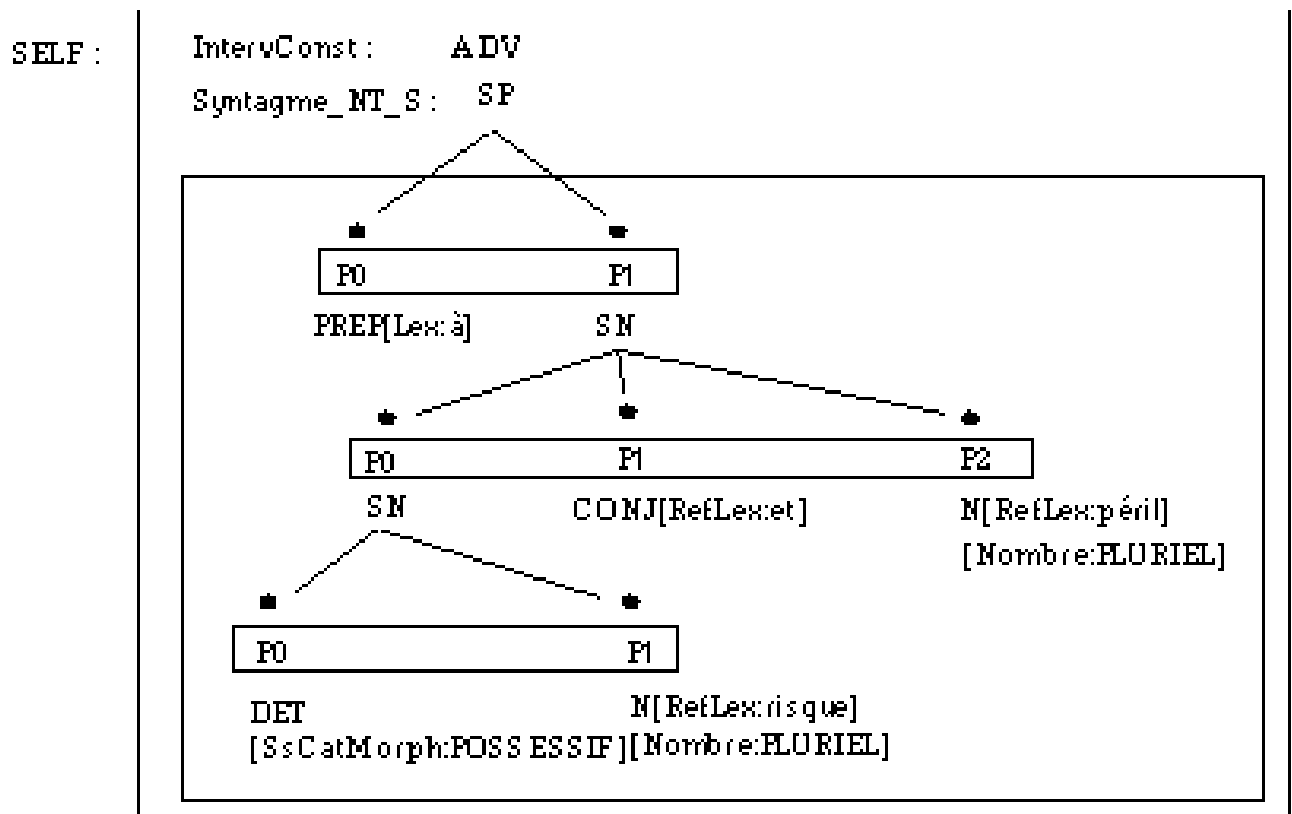


Cond: Si INTERNE P1.SP.P0 Alors INTERNE P1.SP.P1.SM.P0
 Cond: Si INTERNE P1.SP.P1.SM.P0 Alors INTERNE P1.SP.P0
 Cond: Si INTERNE P1.SP.P1.SM.P1 Alors INTERNE P1.SP.P0

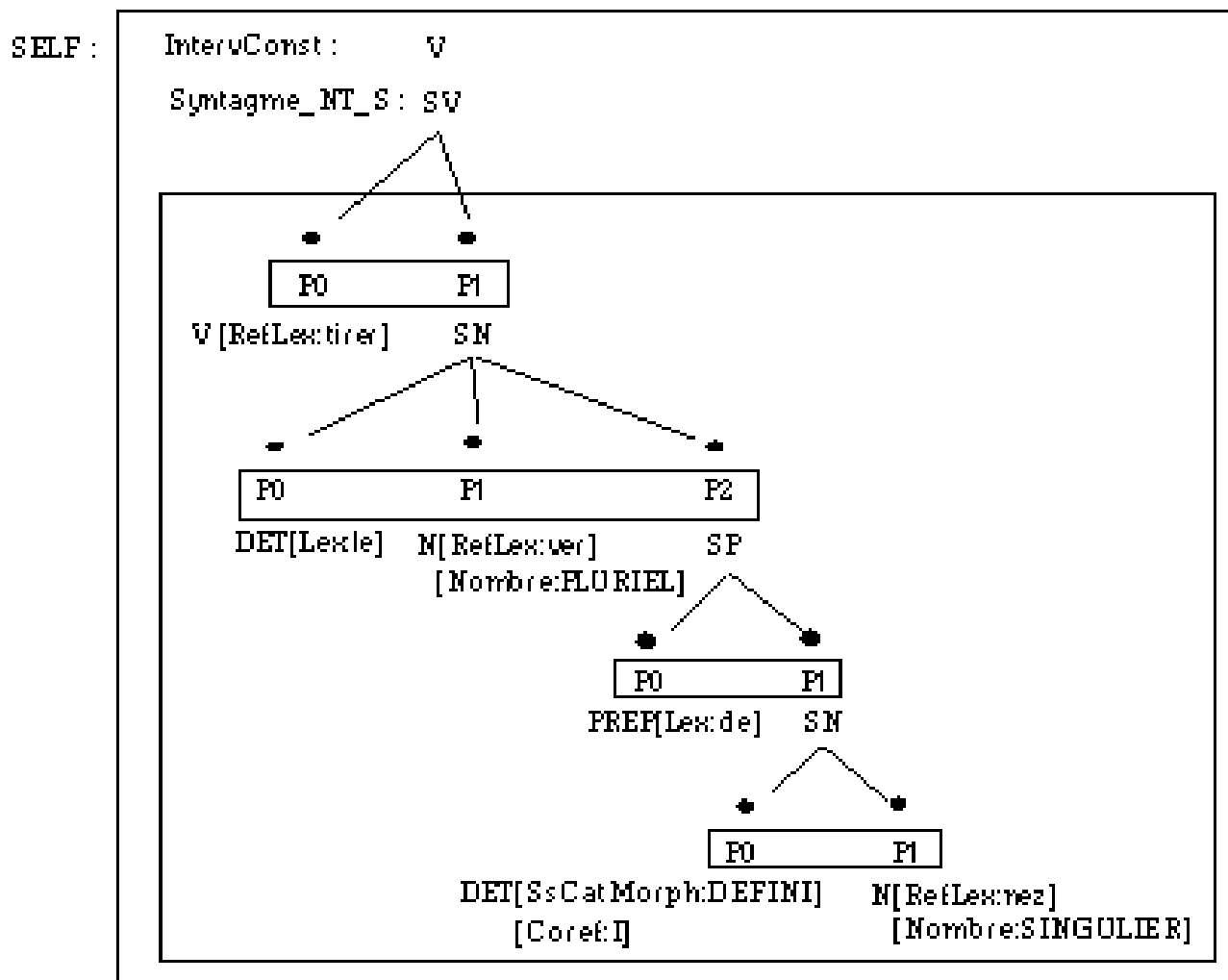
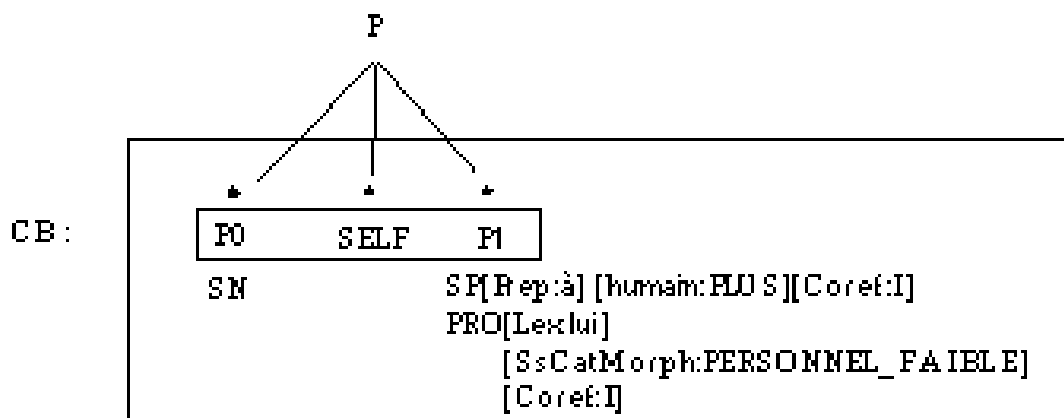
Exemple (6) : SN mettre SN en ïuvre



Exemple (7) : à ses risques et périls



Exemple (8) : tirer les vers du nez



Exemple (9) : au (réel) détriment de SN

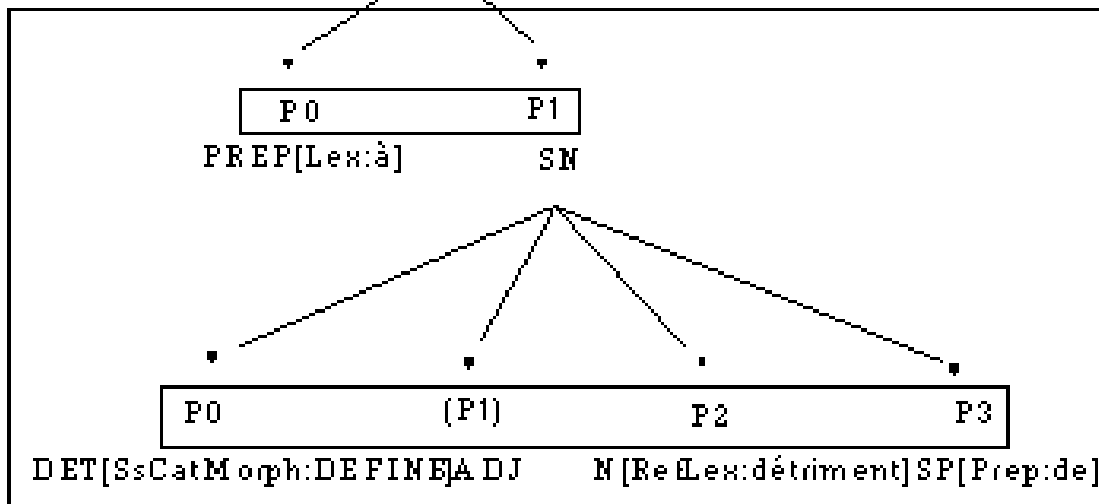
à son (réel) détriment

Solution 1 : Relation de transformation entre deux syntagmes :

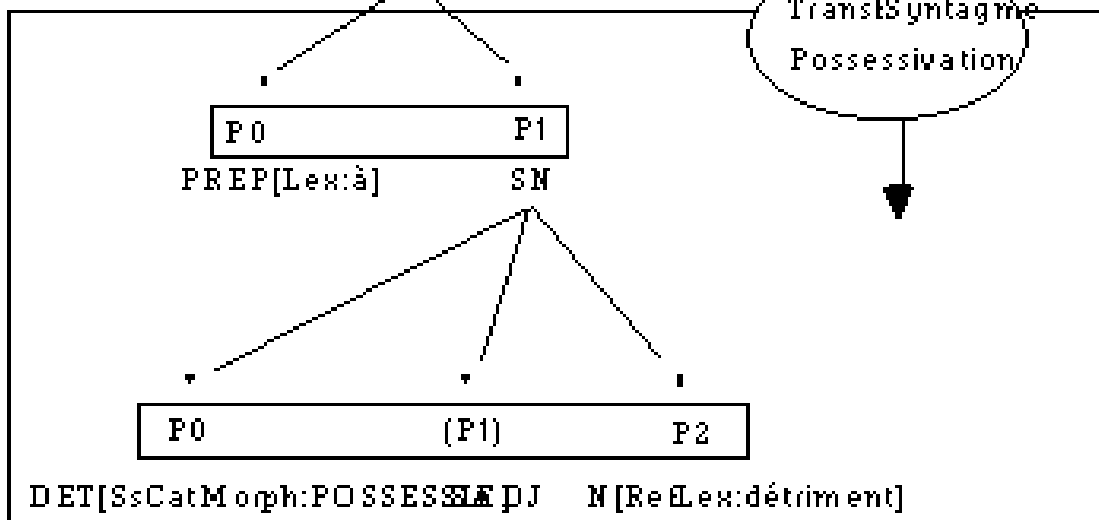
SELF :

InterwConst: ADV

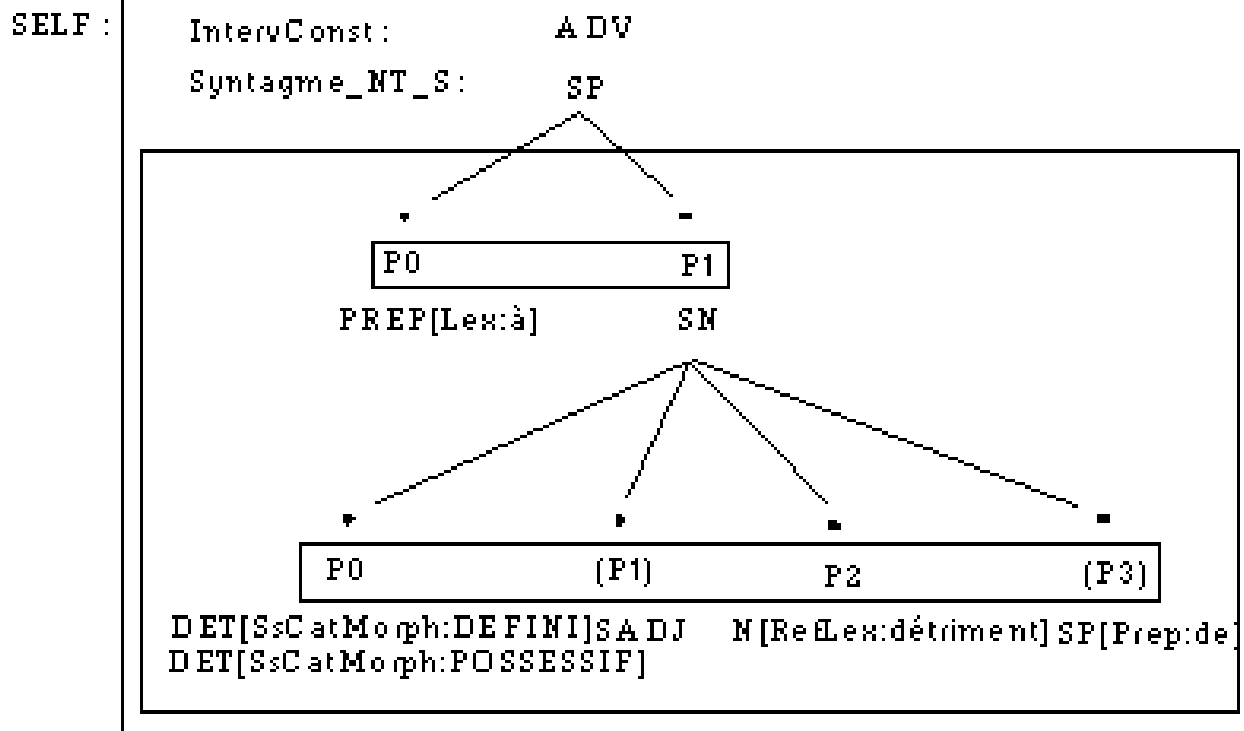
Syntagme_NT_S: SP



Syntagme_NT_S_I: SP

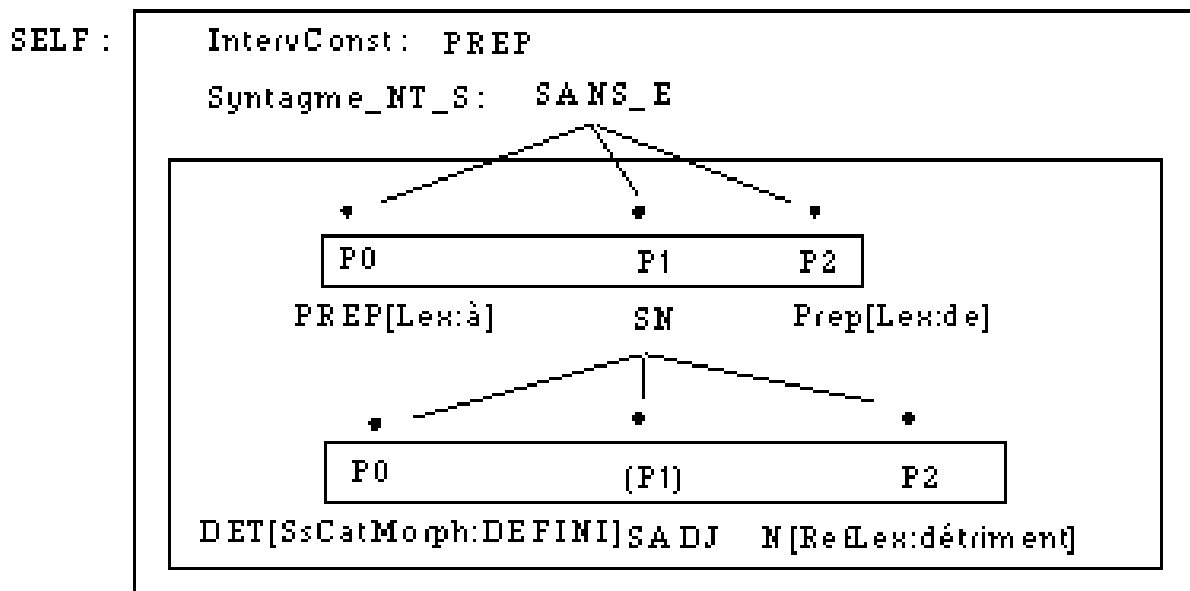
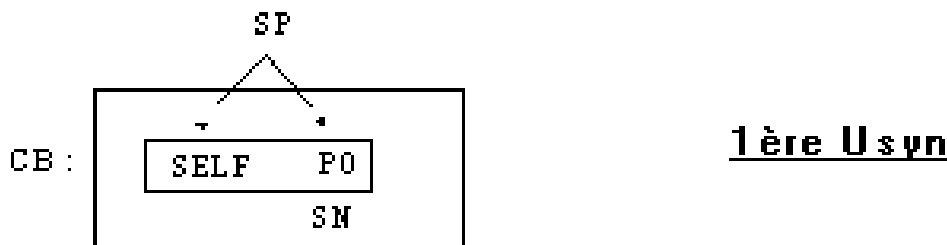


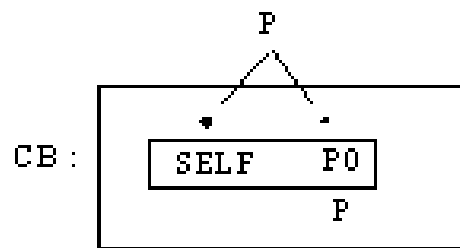
Solution 2 : Positions inter-conditionnées



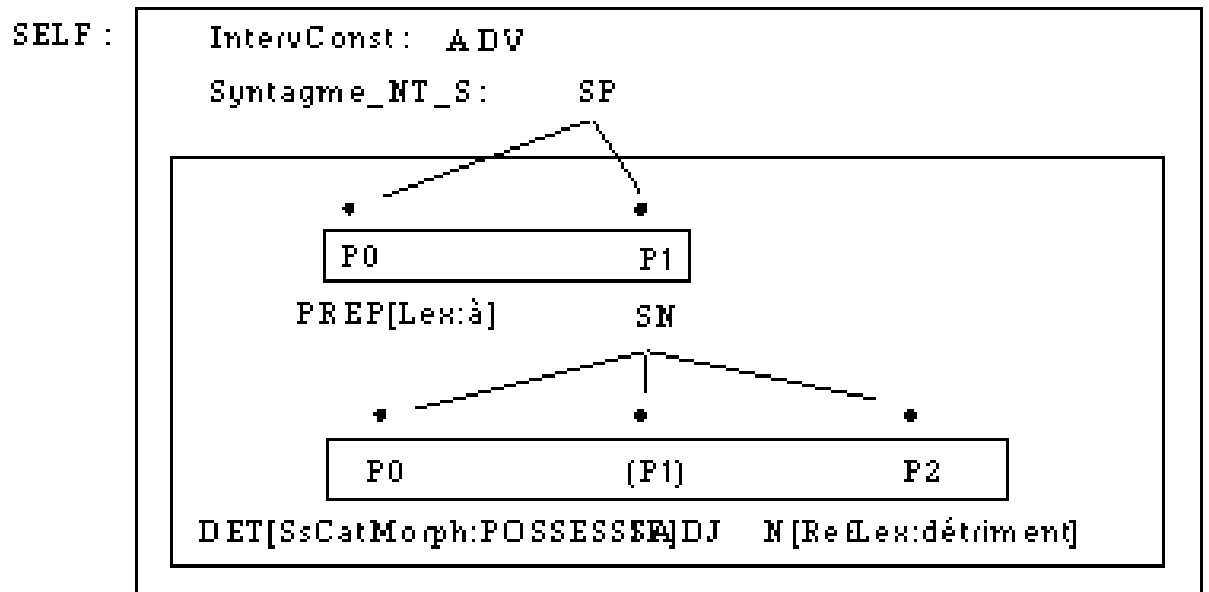
Cond : Si INTERNE P1.SN.P0.DET[SsCatMorph:POSSESSIF] Alors INTERNE
 Cond : Si INTERNE P1.SN.P0.DET[SsCatMorph:DEFINI] Alors INTERNE P3

Solution 3 : deux unités syntaxiques, dont une recouvrant la locution traditionnelle "au détriment de".



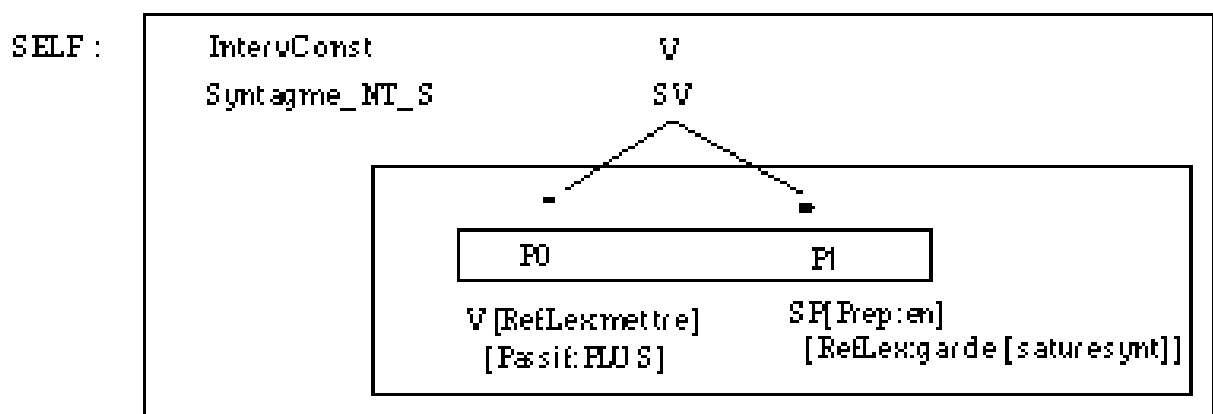
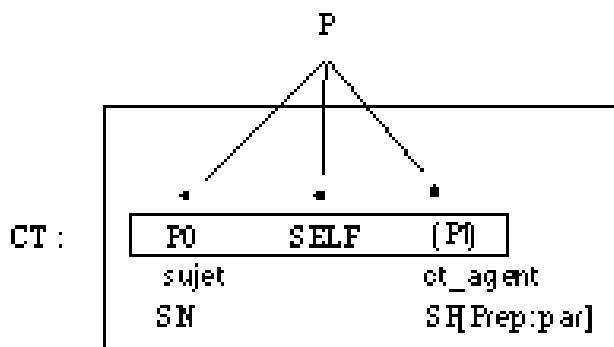
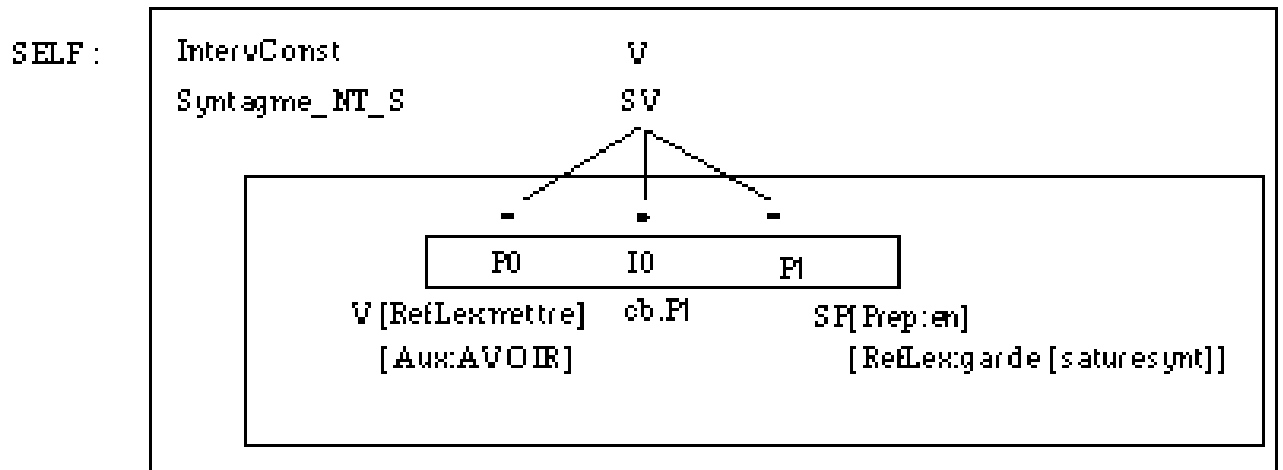
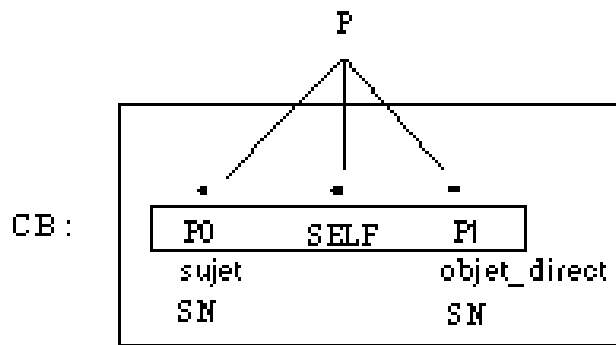


2ème U syn



Exemple (10) : SN1 mettre en garde SN2

SN2 être mis en garde par SN1



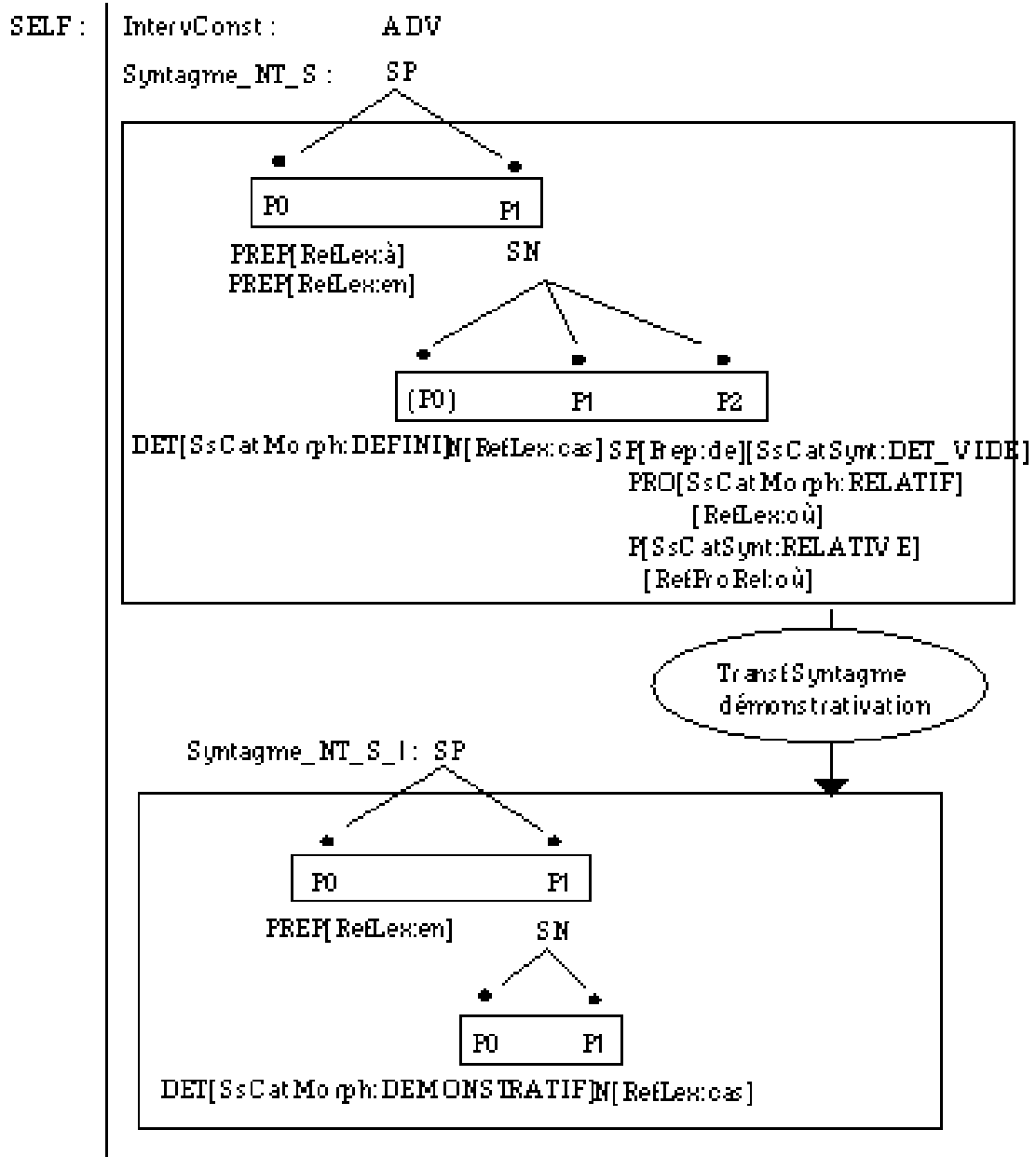
Exemple (11) : au cas où

au cas où P

en cas de SN

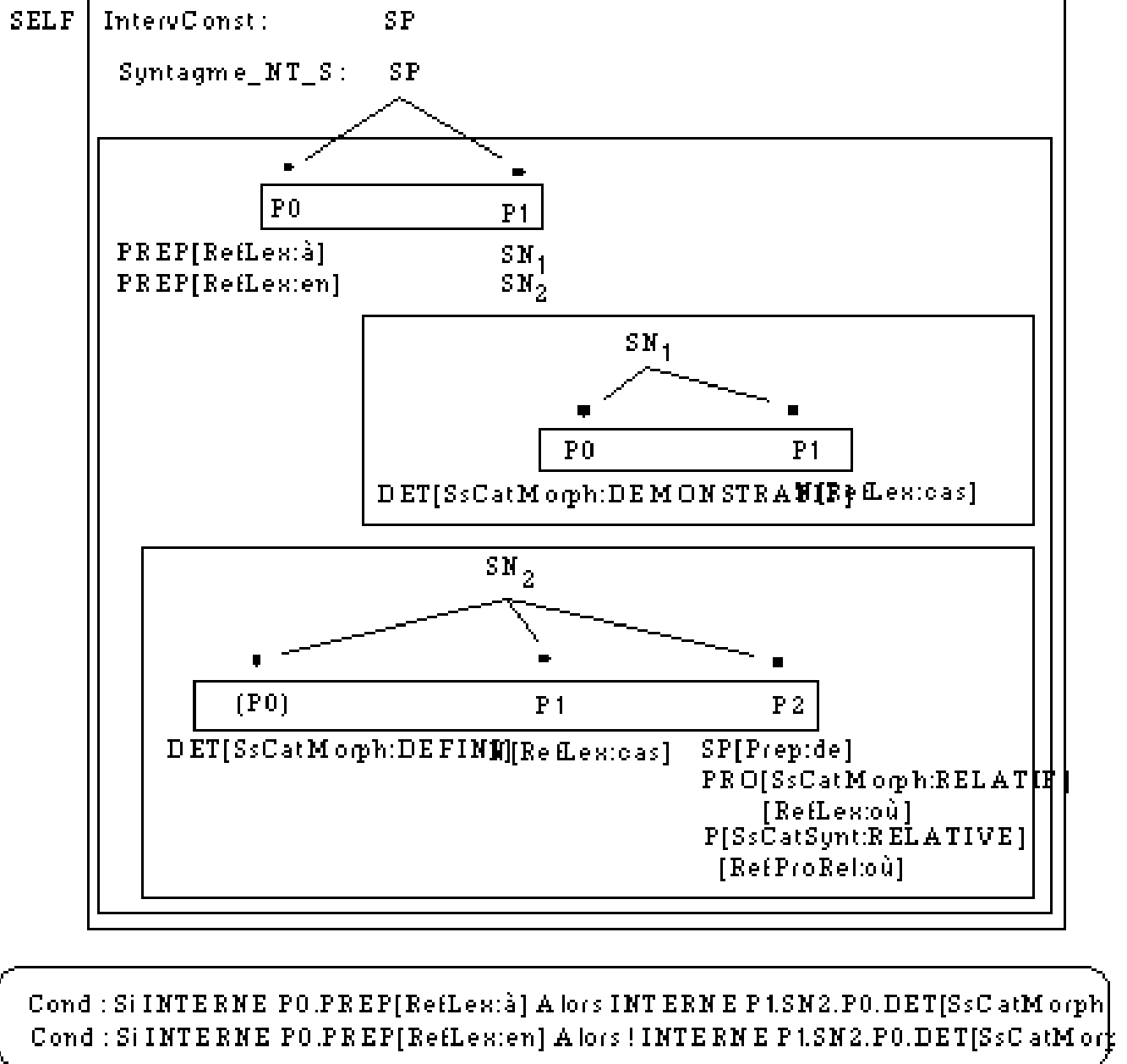
en ce cas

Solution 1 : Relation de transformation entre deux syntagmes :



Cond : Si PO.PREP[ReLex:en] Alors !P1.SN.PO & P1.SN.F2.SF[Prep:de][SsCat Synt:DET_VIDE]
 Cond : Si PO.PREP[ReLex:à] Alors P1.SN.PO & P1.SN.F2.SF[Prep:de][SsCat Synt:DET_VIDE]

Solution 2 : Positions inter-conditionnées



Conventions graphiques sur les Modes de Composition

Les Modes de Composition sont pour des raisons de lisibilité éclatés en deux : composition vers l'INTERIEUR, composition vers l'EXTERIEUR. Par contre on regroupe dans le même schéma, tous les MdC d'une Usyn composée - c'est-à-dire ses MdC sur chacune de ses Usyn composantes - et par abus de langage, on appelle l'ensemble "MdC".

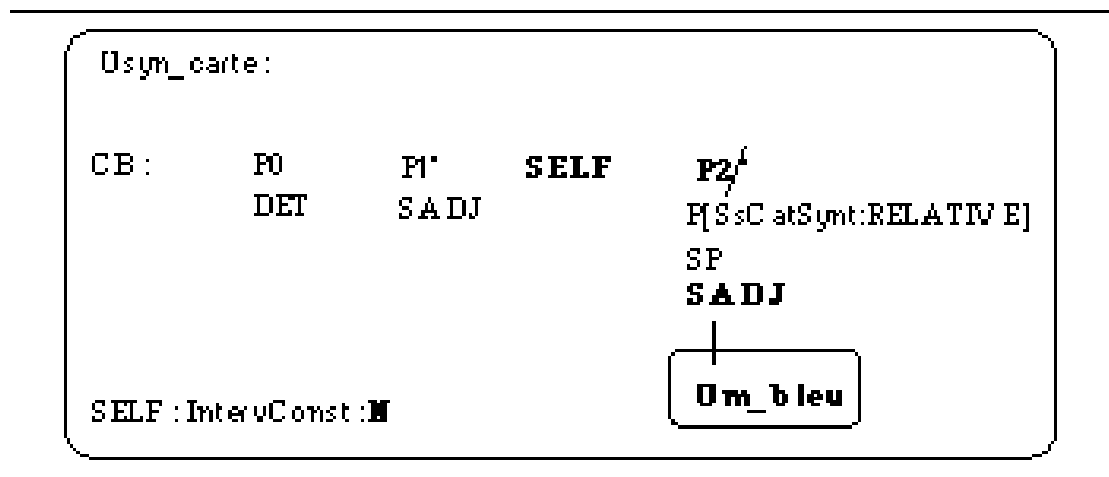
Les composants Um et Usyn sont dans des encadrés ronds.

Les appels de composants Usyn ou Um sont exprimés par une ligne reliant le syntagme appelant et le composant.

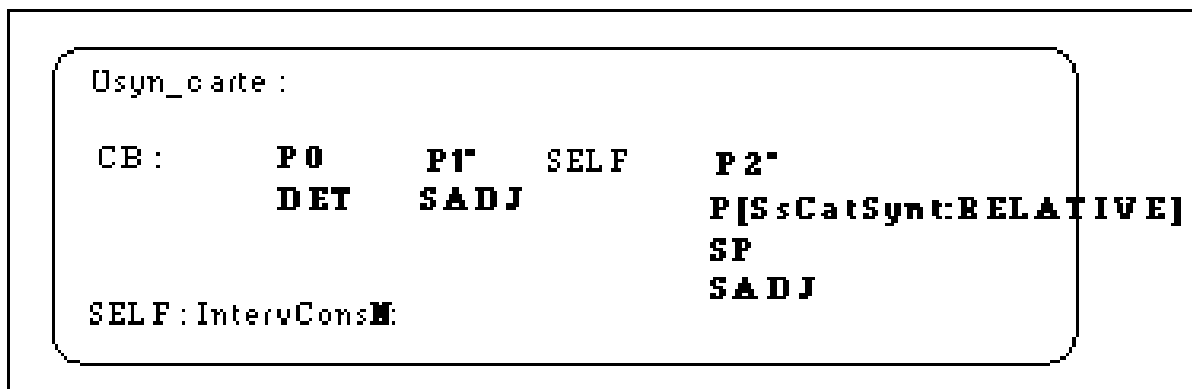
Les Positions et Syntagmes dont on veut hériter sont en gras. Les Syntagmes ou traits ajoutés lors de la composition sont dans des encadrés ronds. La perte de la répétabilité (étoile) est indiquée par l'étoile barrée.

Exemple (12) : carte bleue

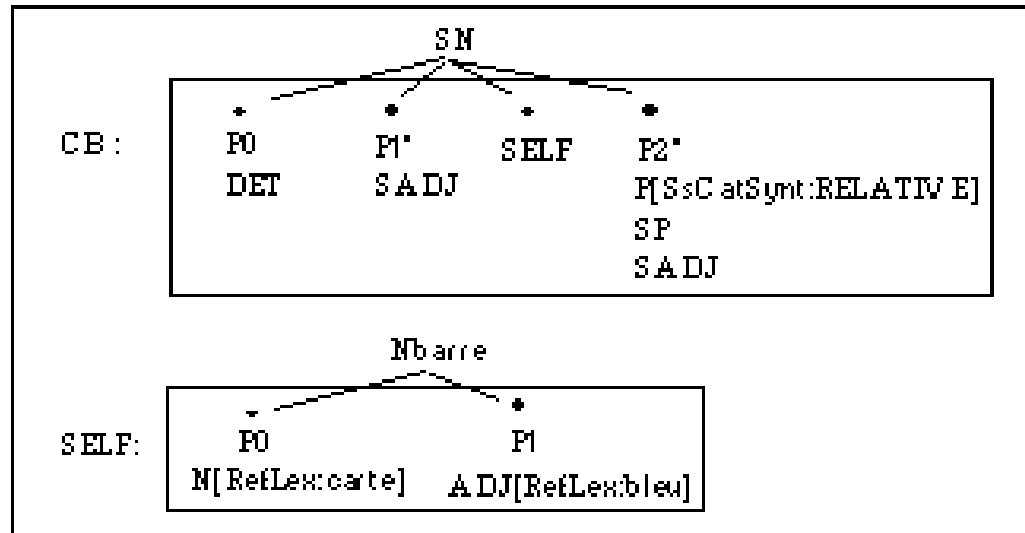
Mode de Composition INTERNE de : CARTE_BLEUE



Mode de Composition EXTERNE de : CARTE_BLEUE



Unité syntagmique composée : Carte_Bleue :

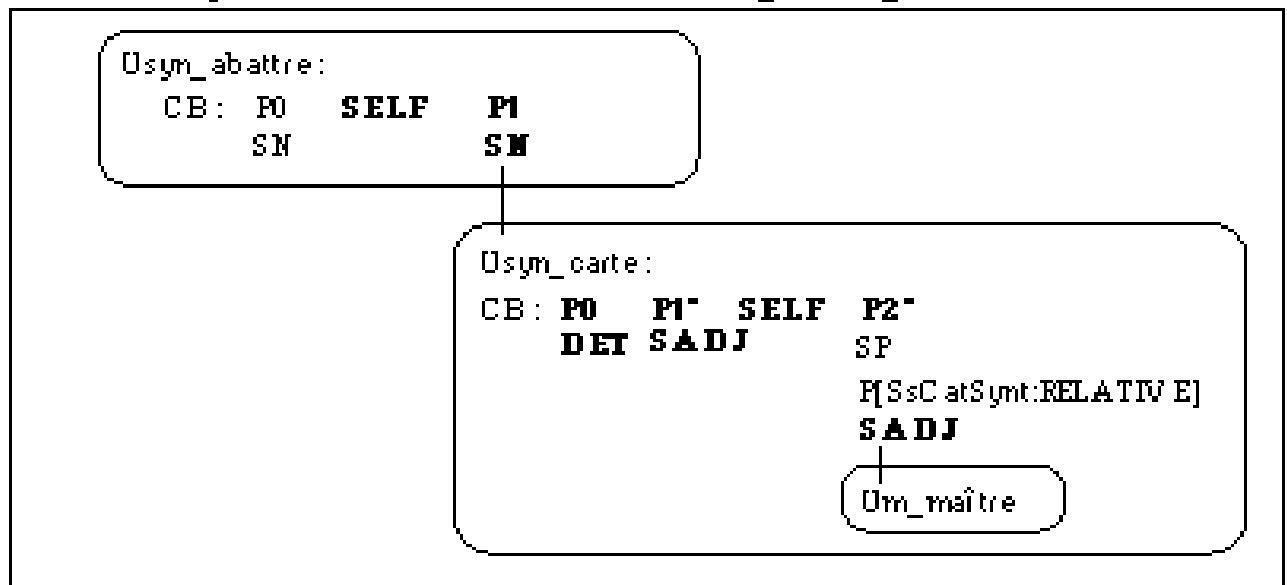


Exemple (13) : *abattre carte ma"tresse*

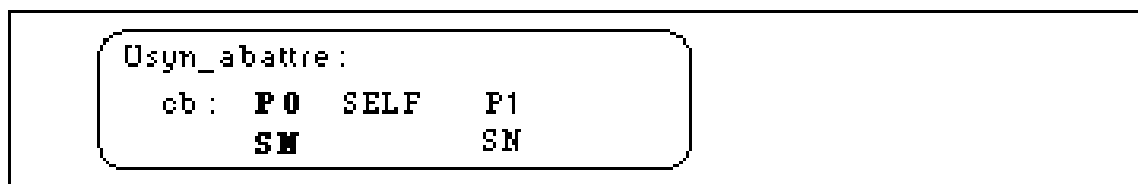
NB : Cet exemple est repris de façon plus détaillée dans la partie H-III de ce document (DTD SGML - Exemples de données balisées)

Solution 1 : 3 composants (Usyn abattre, Usyn carte et Um ma"tre)

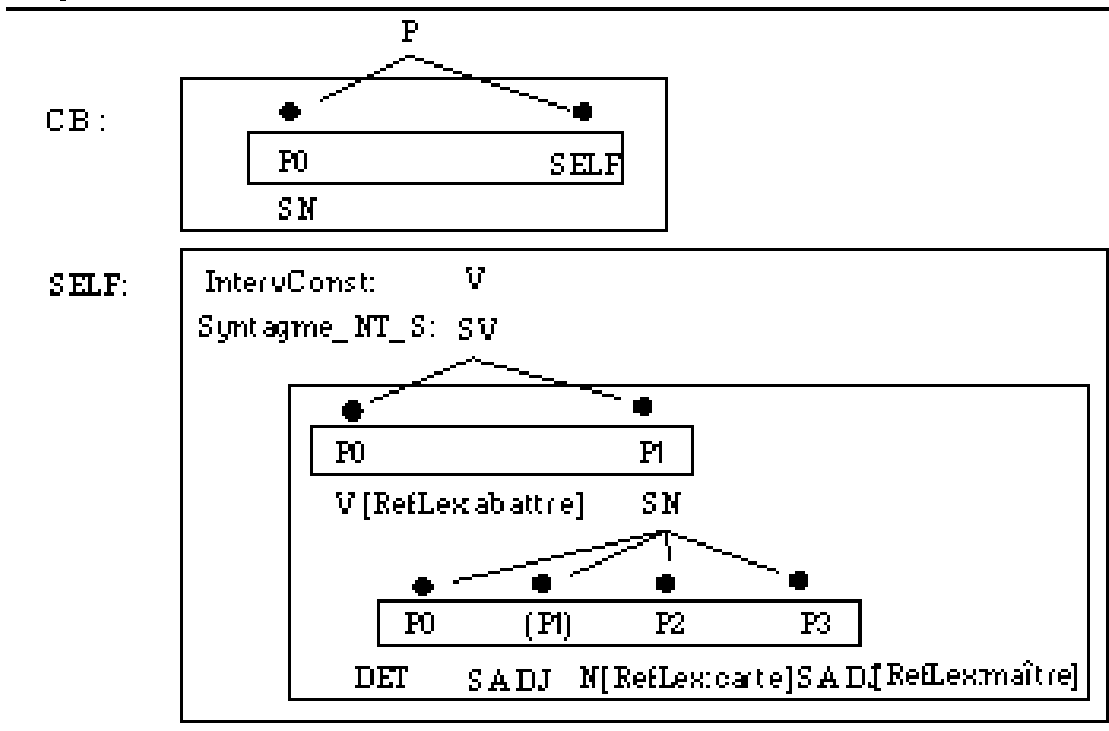
Mode de composition INTERNE de : *ABATTRE_CARTE_MAITRESSE*



Mode de composition EXTERNE de : *ABATTRE_CARTE_MAITRESSE*

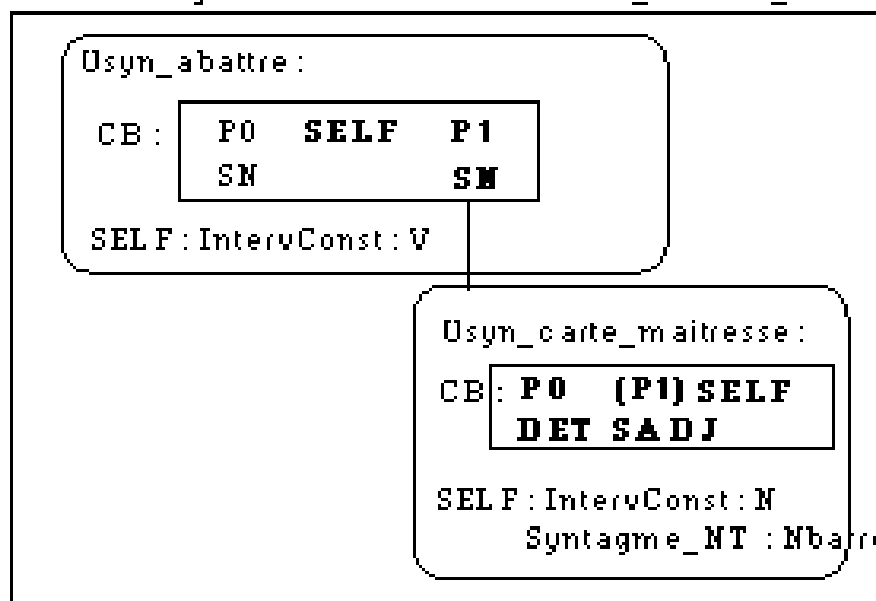


Usyn de: ABATTRE_CARTE_MAITRESSE

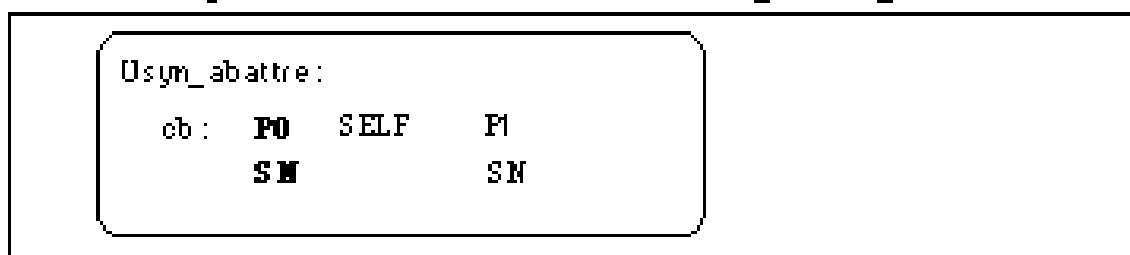


Solution 2 : 2 composants dont un composé Usyn *carte ma"tresse*.

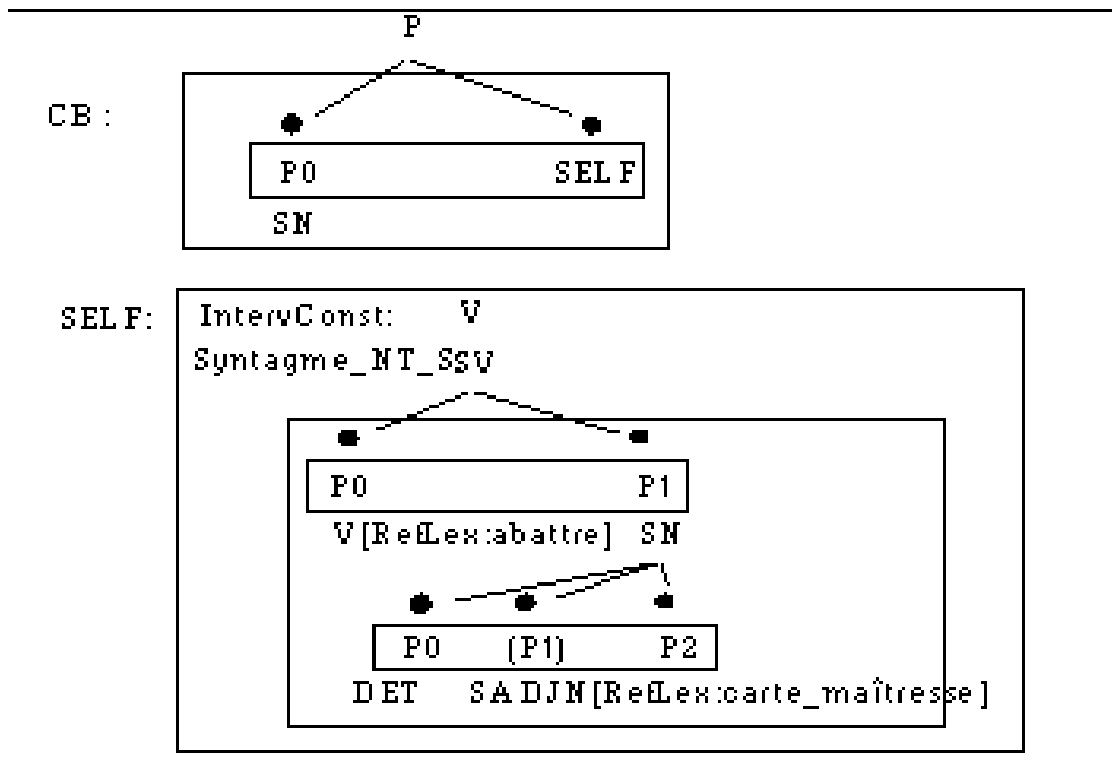
Mode de composition INTERNE de: ABATTRE_CARTE_MAITRESSE



Mode de composition EXTERNE de: ABATTRE_CARTE_MAITRESSE

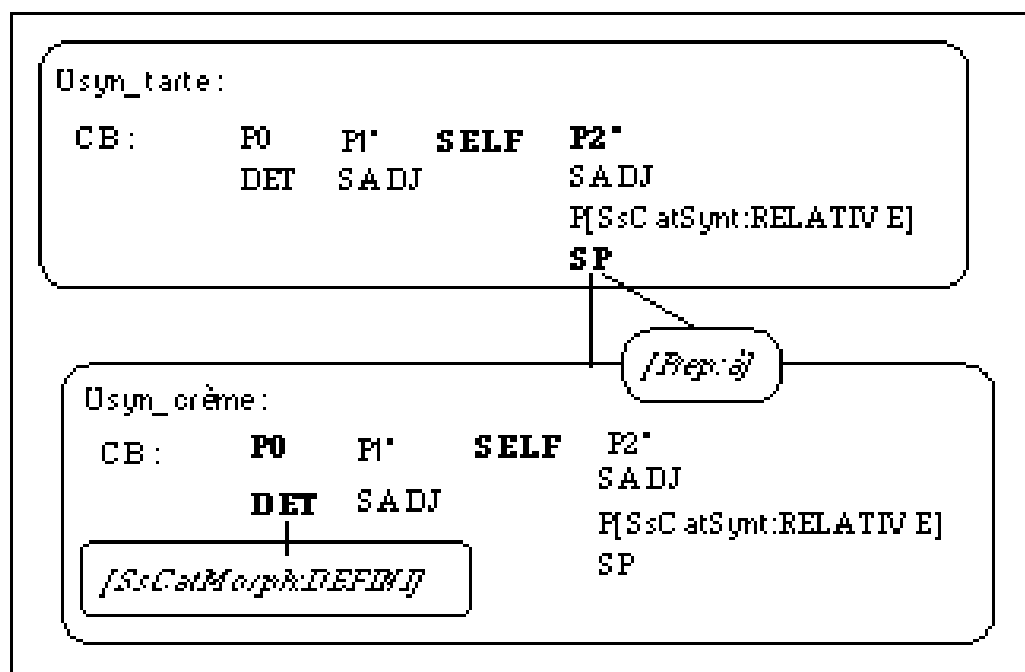


Usyn de: A BATTRE_CARTE_MAITRESSE



Exemple (14) : tarte à la crème

Mode de composition INTERNE de: TARTE_A_LA_CREME



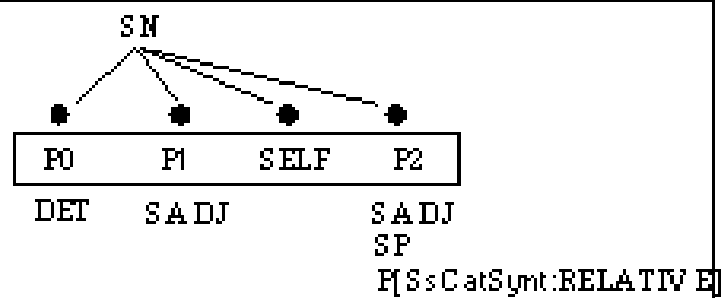
Mode de composition EXTERNE de : TARTE_A_LA_CREME

Usyn_tarte :

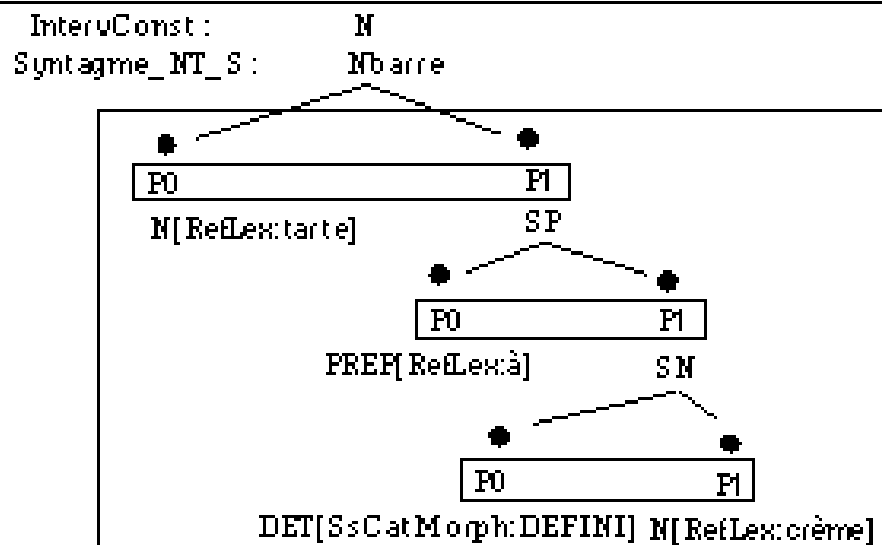
CB :	P0	P1'	SELF	P2'
	DET	SADJ		SADJ
				P[SsCatSynt:RELATIVE]
				SP

Usyn composée : TARTE_A_LA_CREME

CB :



SELF :



Exemple (15): SN avoir admiration pour SN

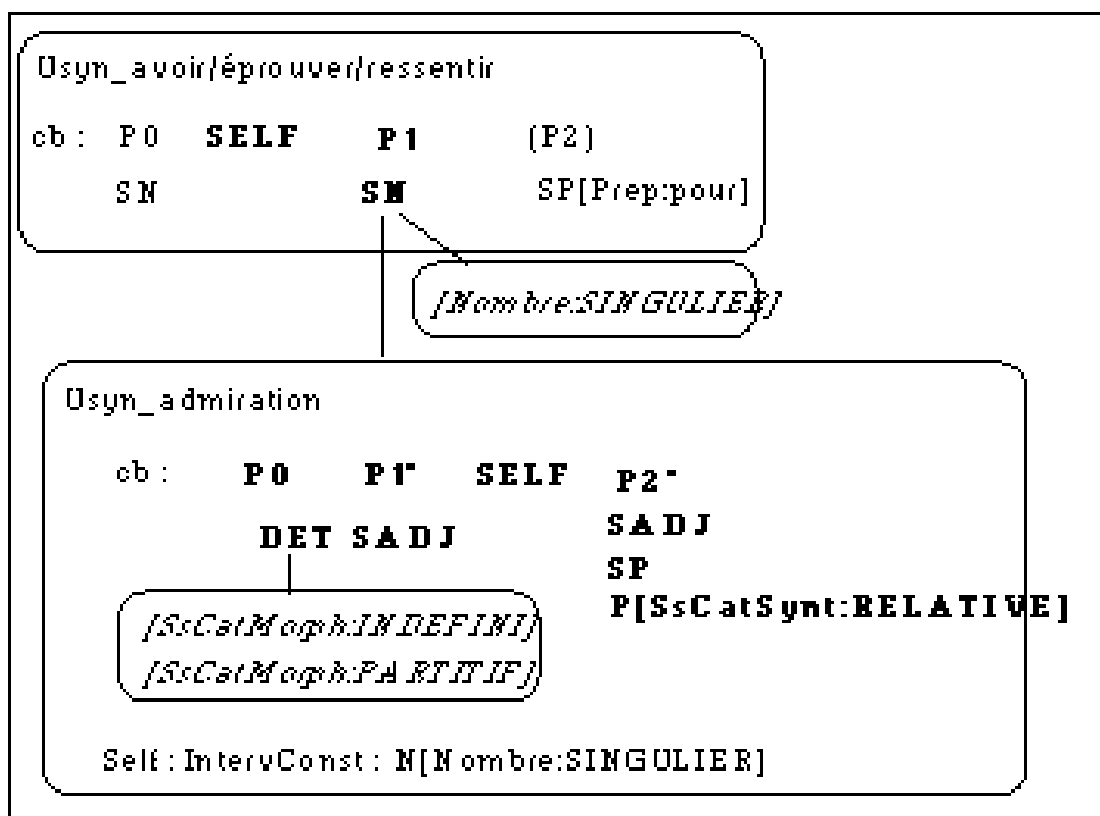
SN éprouver admiration pour SN

SN ressentir admiration pour SN

SN vouer admiration à SN

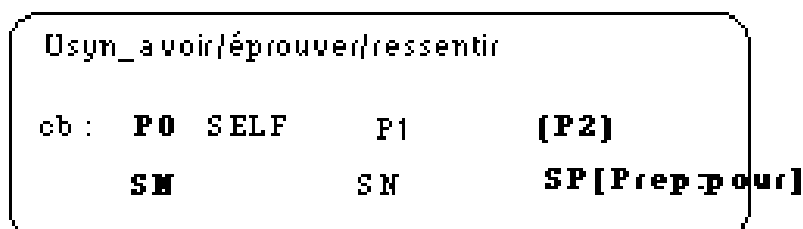
Mode de composition INTERNE partiel du Composé pour :

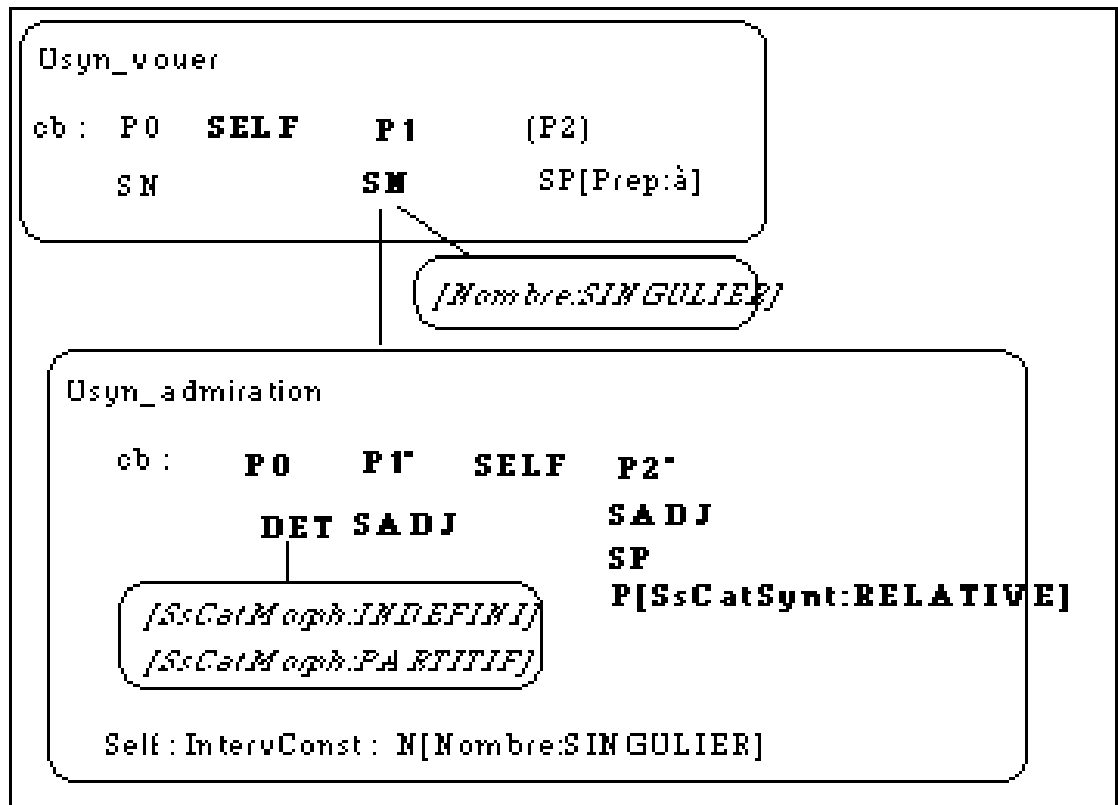
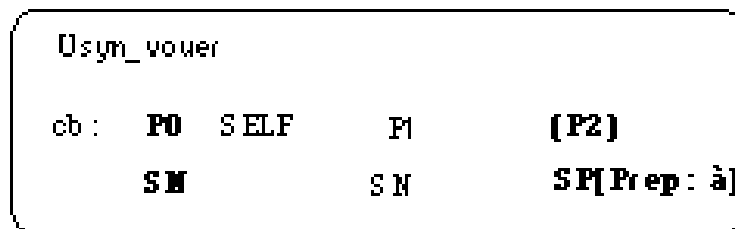
A VOIR/ÉPROUVER/RESSENTIR + ADMI.



Mode de composition EXTERNE partiel du Composé pour :

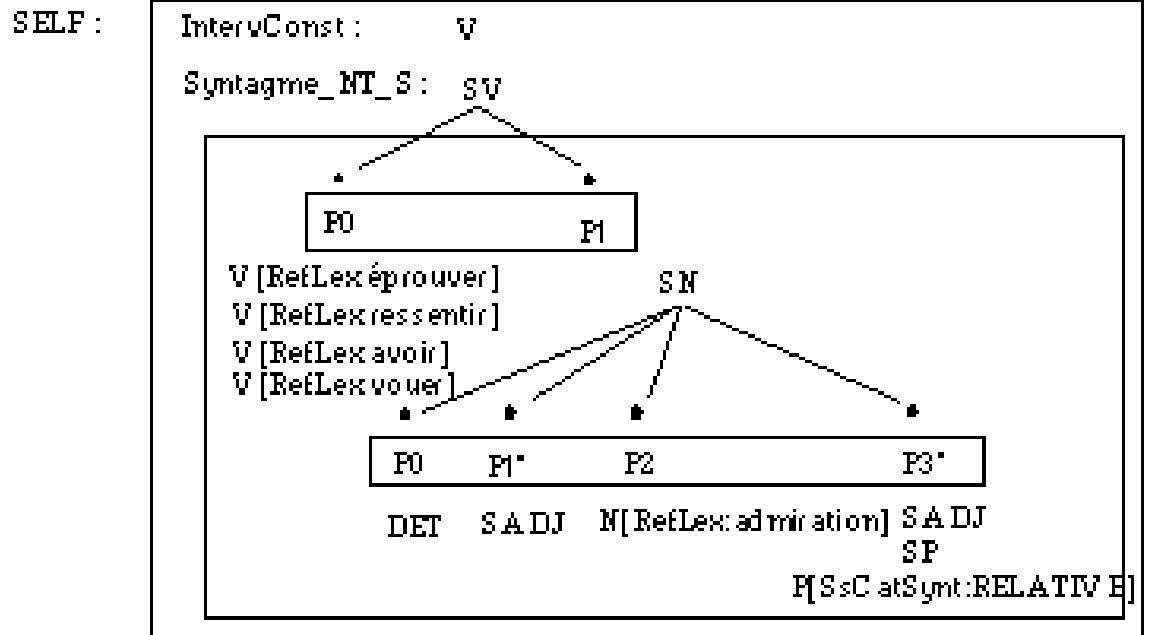
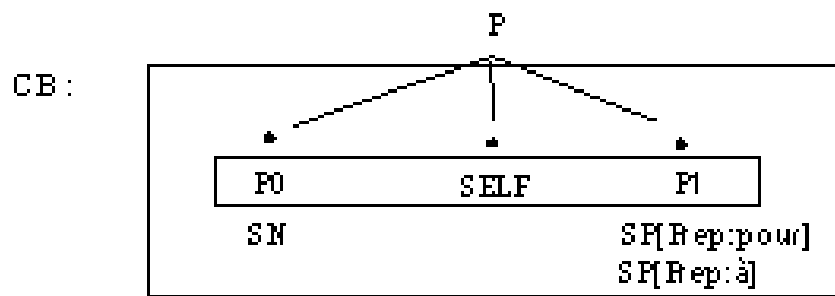
A VOIR/ÉPROUVER/RESSENTIR + ADMI.



Mode de composition INTERNE partiel du Composé pour : VOUER + ADMI*Mode de composition EXTERNE partiel du Composé pour : VOUER + ADMIRA*

La représentation adoptée ici ne rend pas immédiatement compte du codage réel de ce composé : en fait, les 4 alternatives du 1er Composant font appel formellement **au même MdC**.

Usyn composée de: *ÉPROUVER* !...*VOUER DE L'ADMIRATION*



Cond : Si (INTERNE) F0.V [RefLexvouer] Alors EXTERNE F1.SF[Prep:à]
 Cond : Si (INTERNE) ! F0.V [RefLex:vouer] Alors EXTERNE F1.SF[Prep:pour]

Exemple (16) : boîte aux lettres

boîte à lettres

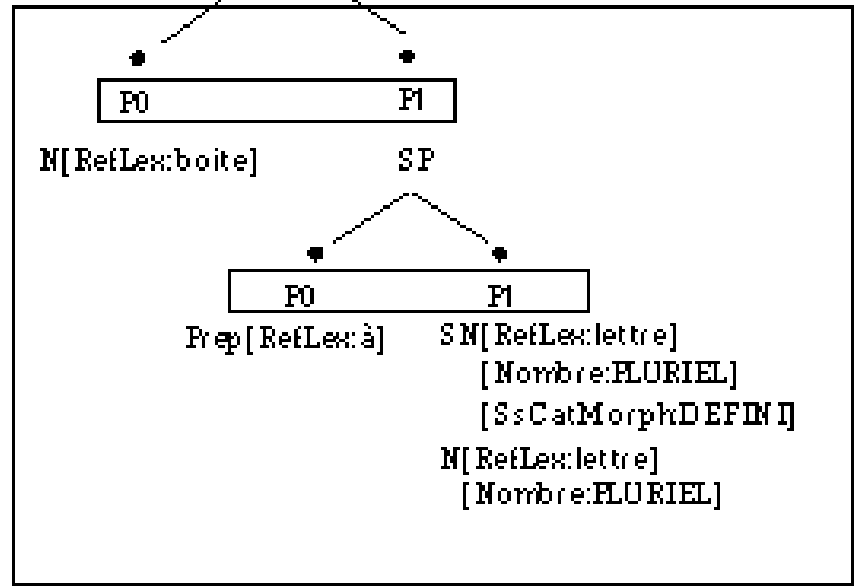
SELF

InterVConst

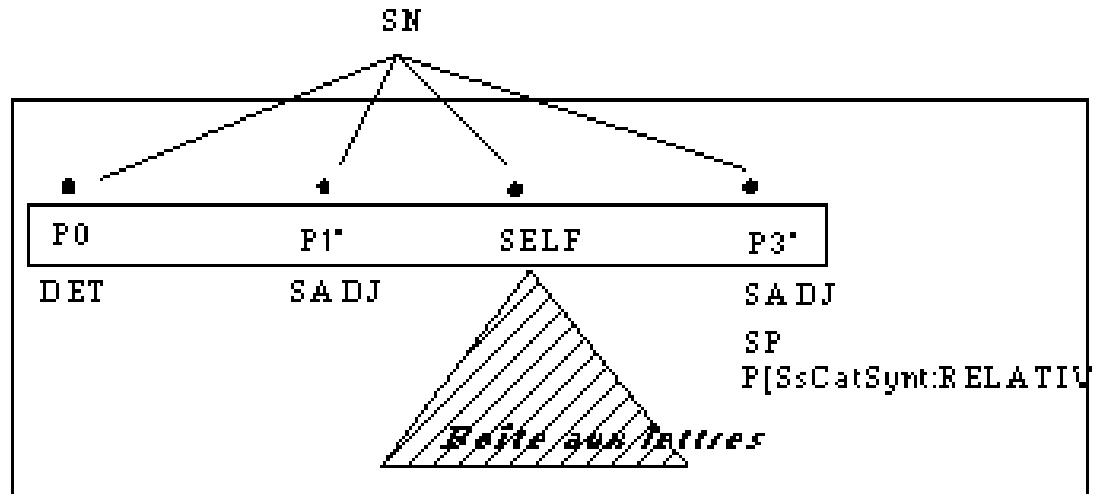
N

Syntagme_NT_S

Nbarre

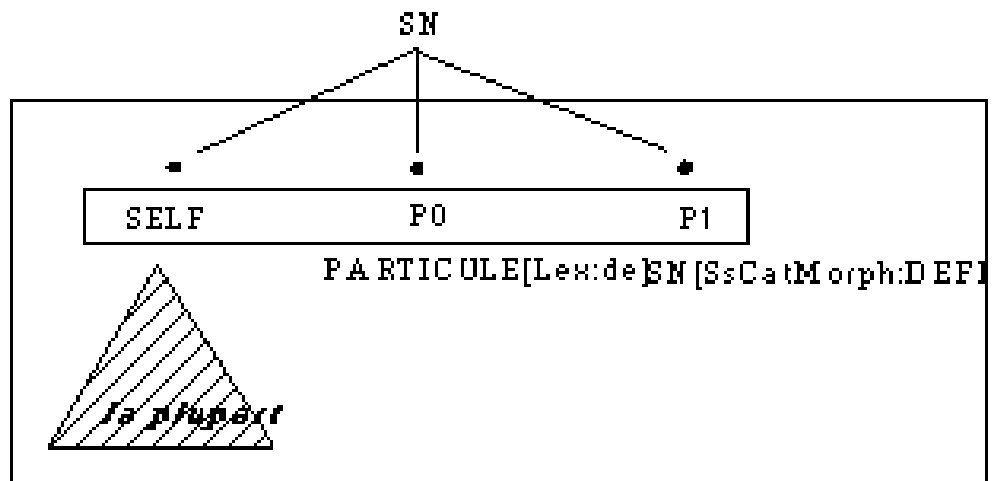


CE :

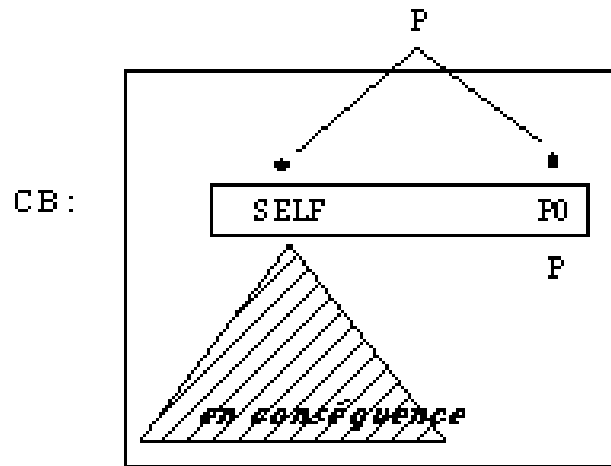


Exemple (17) : la plupart des gens

CB :



Exemple (18) : *En conséquence il n'est pas venu*



E - Bibliographie

A. Abeillé

Parsing idioms in lexicalized TAG's.

4rth Conference of the European Chapter of the Association for Computational Linguistics (EACL),

Manchester, 1989.

ACH-ACL-ALLC

Guidelines for the encoding and interchange of machine readable texts. Sperberg-McQueen & Lou Burnard (eds).

TEI P1 Draft version 1.1 Nov 1990.

M-H Antoni-Lay, Gil Francopoulo & L. Zaysser

A Generic Model for Reusable Lexicons : The GENELEX Project in

Linguistics and Literary Computing. vol. 8 n° 4.

Oxford, Sous Presse.

E. Benveniste

Problèmes de linguistique générale; tome 2. Gallimard

Paris, 1974.

G.Bès & K.Bashung & A.Lecomte

Une modélisation des entrées lexicales.

Projet EUREKA GENELEX Janv. 91.

G.Bès & M.Emorine

Une lecture des tables du Ladi en vue de la définition de la couche syntaxique de GENELEX.

Projet EUREKA GENELEX Janv. 91.

B.Boguraev & T.Briscoe

Computational lexicography for natural language processing.

Longman, 1989.

J-P Boons & A. Guillet & C.Leclère

La structure des phrases simples en français : constructions intransitives. Droz, 1976.

C. Filmore

The case for case in Universals in Linguistics Theory.

Bach. & Harems eds. Holt, Rinehart and Winston, p.1-90.

B.Fradin & J-M Marandin

Autour de la définition : de la lexicographie à la sémantique.

Langue Française 43; p.60-83.

Paris, 1979

Gazdar & Klein & Pullum & Sag

Generalized phrase structure grammar.

Harvard University Press, 1985.

M.Gross

Méthodes en syntaxe: régime des constructions complétives.

Hermann 1975.

R. Jackendoff

X-bar Syntax : a Study of Phrase Structure.

Cambridge, : MIT Press.1977

Semantic Interpretation in Generative Grammar.

Cambridge, Massachussets : MIT Press 1972.

J-C. Milner

Introduction à une science du langage

Seuil 1989.

A. Poncet-Montange

Les composés NàN.

Communication GENELEX. Paris 1990

L. Tesnière

fléments de syntaxe structurale.

Klincksieck 1959

F - MANUEL DE L'UTILISATEUR

1 : Usyn

1.1 : Généralités

L'Unité Syntaxique (**Usyn**) est le point d'entrée dans la couche syntaxique du modèle.

Les Usyn simples ne sont pas partagées par les Um.

Elles peuvent toutes porter une **CombVE**, quadruplet de valeurs d'emploi : niveau de langue (*niveaulgue*), *fréquence*, variation géographique (*vargeog*) et *datation*.

Les Usyn ont une **Description de base** obligatoire (attribut *description*) et de 0 à N **Descriptions transformées** (attribut *description_l*). Une même Description ne pourra avoir les deux statuts de base et de transformée sur la même Usyn.

Les liens de **Transformation** entre Usyn et entre Descriptions sont portés au niveau des Usyn (élément **TransfUsyn** et attribut *transfdescription_l*).

1.2 : Usyn simple

Une Usyn simple permet d'associer une Um à un ou plusieurs comportements syntaxiques et d'y attacher des informations spécifiques telles que l'attestation. Il ne sert en effet à rien, sauf à polluer le dictionnaire ou à lui retirer tout intérêt, d'enregistrer sur une entrée des propriétés virtuelles qui ne sont en fait pas attestées en langue. D'où l'importance du champ *attestation* qui garantit de ces écueils s'il est renseigné.

Une Um est en relation avec 0 à N (par exemple les affixes ou les agglutinés) ou 1 à N Usyn.

Une Usyn simple est en relation avec une et une seule Um dont elle décrit un et un seul des comportements syntaxiques. Les Usyn n'étant pas partagées, il y a au minimum autant d'Usyn simples que d'Um décrites par ces Usyn.

Une Usyn est minimalement décrite par une Description de base et 0 à N Descriptions transformées.

1.3 : Usyn composée

Une Usyn composée est un Syntagme bien formé, totalement ou partiellement lexicalisé. Elle est en relation avec les Um et/ou Usyn qui la lexicalisent et la composent (élément **Composition**).

2 : Descriptions

Une **Description** est une entité regroupant le couple **Construction** + **Self**. Ainsi les Constructions peuvent être partagées indépendamment des propriétés de Self.

Une même Description peut être partagée par différentes Usyn.

Une Description peut ne pas porter de Construction dans le cas où l'on souhaite décrire un comportement d'appelé sans préciser le contexte d'apparition.

Par contre, la présence du Self est obligatoire.

Une Description peut porter 0 à N **Conditions**, qui font également partie de sa définition.

Une même Description peut être Description de base pour une Usyn mais Description transformée pour une autre.

3 : Self

L'entité **Self** permet d'enregistrer les propriétés attachées à l'entrée décrite lorsqu'elle apparaît dans une tournure syntaxique donnée. Self ne pointe pas sur l'Um qu'il figure et peut ainsi être partagé par différentes Descriptions.

3.1 : Self simple

Les propriétés de Self lui sont attachées pour une Construction (**IntervConst**), ou reçues d'un élément extérieur à sa Construction (**ComportAppelle**). IntervConst et ComportAppelle sont des informations complémentaires non exclusives. L'un, l'autre ou les deux peuvent être renseignés, mais au moins l'un des deux l'est.

3.1.1 : Self intervenant dans la Construction (IntervConst)

Un **IntervConst** permet de spécifier et de contraindre Self en tant qu'Intervenant dans la Construction. Il regroupe :

- **des Syntagmes terminaux**

Il s'agit de couples (étiquette syntagmatique / liste de traits) décrivant la(les) réalisation(s) syntagmatique(s) de Self. Tous les types de traits restrictifs peuvent porter sur Self, à l'exception des traits lexicaux.

Ex : s'être moqué / avoir moqué

Self :

IntervConst :

syntagme_t_1 :

V[Pronominal : SE] [Aux : ETRE]

V[Aux : AVOIR]

- **une Fonction**

Fonction prise par Self en tant qu'intervenant dans la construction, dont la valeur est "TETE" si l'on décrit un comportement majeur de Self, ou toute autre fonction reçue d'un élément de sa construction si l'on décrit un de ses comportements mineurs.

GENELEX propose une liste de valeurs de fonctions qui peut être enrichie, modifiée, ou entièrement refondue :

TETE, SUJET, OBJET_DIRECT, OBJET_INDIRECT, ATTRIBUT_SUJET, ATTRIBUT_OBJET, EPITHETE_GAUCHE, EPITHETE_DROIT, SPECIFIEUR, MODIFIEUR, GENITIF.

La fonction n'est pas obligatoirement renseignée (attribut *fonction*).

Ex : La majorité des N

Self :

IntervConst :

syntagme_t_1 : N[Nombre:SINGULIER]

fonction : SPECIFIEUR

• des Rôles Thématiques

Self peut porter en propre un ou plusieurs Rôles Thématiques (**RoleTh**) dans une Construction donnée.

GENELEX propose une liste de valeurs de rôles thématiques qui peut être enrichie, modifiée, ou entièrement refondue :

AGENT, PATIENT, DESTINATAIRE, SOURCE, BUT, CAUSE, MANIERE, LOCATIF, TEMPS, INSTRUMENT, THEME.

Les rôles thématiques ne sont pas obligatoirement renseignés (attribut *roleth_l*).

3.1.2 : Self appelé par un élément externe à sa Construction (ComportAppele)

On note sur un **ComportAppele** le même type de propriétés que sur un IntervConst, mais qui sont cette fois émises sur Self par un élément non décrit dans la Construction. Contrairement à l'IntervConst, un ComportAppele ne référence qu'un seul Syntagme terminal. Tous les traits restrictifs sont autorisés à l'exclusion des traits lexicaux. Un même Self peut avoir plusieurs comportements d'appelé différents (attribut *comportappelle_l*).

Ex : Self :

comportappele_l :

syntagme_t : ADJ

fonction : ATTRIBUT_SUJET

3.2 : Self composé

La distinction entre Self intervenant dans construction externe et Self fonctionnant comme appelé s'applique aussi aux Unités syntaxiques composées. Mais la catégorie indiquée dans *IntervConst* et *ComportAppelle* correspond à la catégorie fonctionnelle (externe) du composé.

Ex : mettre en marche (VERBE)

Ce qui est spécifique aux Unités syntaxiques composées en revanche, c'est le fait que Self soit non pas atomique mais composite (multi-lexical). Pour renseigner la Structure Interne de Self, qui n'est autre qu'un Syntagme partiellement ou totalement lexicalisé, on renseigne son champ *syntagme_nt_s* par un Syntagme (**Syntagme_NT_S**) qui comprend minimalement une étiquette syntagmatique. Ce Syntagme peut aussi être réécrit le cas échéant.

Ex : mettre en marche (SV)

4 : Constructions

4.1 : Définition générale

Une **Construction** décrit le contexte ou cadre syntaxique propre à l'entrée décrite.

Elle rentre dans la définition d'une Description. S'il s'agit de la Description de base, son champ *squelettique* doit valoir NON.

Une même Construction peut être partagée par différentes Descriptions.

Une Construction est fondamentalement définie par une liste ordonnée de **Positions** (attribut *position_c_l*) qui comprend au moins un élément.

Dans une vision "syntaxiste" où les Constructions correspondent à des Syntagmes, on peut leur associer une **étiquette syntagmatique non terminale** (attribut *etiquettesynt*) et l'ensemble des traits restrictifs autorisés pour ce Syntagme (attribut *trait_l*).

Les champs *optionnalite*, *insereself* et *solidarite* participent pleinement aussi à la définition d'une Construction :

- **Optionnalité**

Le champ *optionnalite* doit toujours être renseigné, c'est-à-dire référencer un élément **Optionnalite** qui comporte minimalement un libellé. Ce libellé donne la liste ordonnée des **Positions** (telles que stockées dans l'attribut *position_c_l*) et indique par des parenthèses quelles sont les **Positions** optionnelles.

L'optionnalité de chaque **Position** ne lui est pas attachée car elle dépend de la Construction où elle apparaît.

Les **Positions** sont identifiées par un Symbole Pi. Le rang de la **Position** est une donnée relative, non attachée à la **Position**. Néanmoins, dans le modèle GENELEX, comme on référence obligatoirement une **Position** dans le contexte d'un Syntagme ou d'une Construction donnée, on l'identifiera toujours par son rang.

Ex : P0 P1 (P2)

Self ne figure pas dans cette liste.

• Inseself

- Vision atomiste

Ce champ n'est pas obligatoirement renseigné.

La liste des **Positions** ne décrivant que le schéma de complémentation de **Self**, et jamais un niveau de réécriture syntagmatique, celle-ci est toujours au même niveau syntaxique que **Self**.

- Vision syntaxiste

Dans cette vision, les listes de **Positions** sont des listes de réécriture syntagmatique.

Lorsque la liste de **Positions** correspond au niveau de réécriture de **Self**, ce champ doit obligatoirement être renseigné .

Si **Self** ne s'insère pas à ce niveau de réécriture alors ce champ est obligatoirement non renseigné. Cette liste de **Positions** est d'un niveau hiérarchique supérieur ou inférieur à **Self** et décrit, non pas la complémentation de **Self**, mais un contexte syntaxique prototypique partiellement restreint.

Le champ *inseself* indique quel est le point d'insertion de **Self** en début, milieu ou fin de la liste des **Positions**. Le numéro donné est l'indice de la **Position** DEVANT laquelle s'insère **Self**. Dans le cas où **Self** s'insère en fin de liste, l'indice est celui de la dernière Position + 1.

Ex :

P : P0 P1 P2

SUJET TETE ATTRIBUT_SUJET

P0 : PRO[SsCatMorph:IMPERSONNEL]

P1 : V[SsCatSynt:COPULE]

P2 : SADJ = (P0) SELF

P0 : ADV

Sur P, le point d'insertion de **Self** n'est pas renseigné.

Sur SADJ, il vaut 1 (0+1).

• Solidarité

Le champ texte *solidarite* est facultatif. Il indique par un tiret les paires solidaires c'est-à-dire les éléments Position, Self, ou Insertion qui constituent une suite insécable.

Ex : SELF-P1-P2

4.2 : Squelettes de Constructions

Les Modes de Transformation entre Constructions (**ModifConstruction**) utilisent des squelettes de Construction, c'est-à-dire des Constructions totalement ou partiellement remplies que l'on vient combler à partir de la Construction d'origine.

Ces squelettes de Constructions sont semblables aux Constructions à proprement parler à ceci près qu'ils doivent avoir un champ *squelettique* valant OUI, qu'ils autorisent leurs **Positions** à avoir une distribution non renseignée et qu'ils sont à mettre en regard avec un ModifConstruction qui à partir d'une Construction origine permet d'obtenir une Construction résultat.

5 : Positions

Les **Positions** sont définitoires des Constructions. Elle identifient un paradigme syntagmatique exigé par une entrée, et permettent d'associer une **Fonction** et des rôles thématiques (**RoleTh**) à ce paradigme. Typiquement un complément de verbe est défini par une **Position**.

Les **Positions** sont définies par un triplet: distribution, fonction et rôles thématiques.

Une distribution est une liste de 1 à N **Syntagmes** (attribut *syntagme_c_l*).

Fonction et rôles thématiques (attributs *fonction* et *roleth_l*) ne sont pas obligatoirement renseignés.

Par ailleurs, on indique sur chaque **Position** si elle est répétable ou non (champ *repetable*, valeur

OUI ou NON).

Enfin, une **Position** peut signaler quels syntagmes de sa distribution entretiennent une relation de Transformation (attribut *transfsyntagme_l*).

Les **Positions** sont couramment identifiées par leur rang dans la liste des **Positions** sous la Construction. Cependant, le rang de la **Position** ne lui est pas définitoire. Ca n'est qu'une valeur prise en contexte - à savoir la liste des **Positions** où elle apparaît.

Les **Positions** sont partageables d'une **Construction** à l'autre.

On verra plus loin qu'elles permettent également la réécriture des **Syntagmes**.

Seules les **Positions** qui interviennent dans les squelettes de **Construction** (champ *squelettique* valant OUI) peuvent avoir une distribution vide.

6 : Insertions

Les **Insertions** traitent les phénomènes de réentrance (pris dans un autre sens que celui des grammaires d'unification) entre l'intérieur et l'extérieur du composé, c'est-à-dire des cas où une **Position** de la **Construction** externe apparaît en surface au milieu des **Positions** de la Structure interne.

Ex : SN mettre SN en ŷuvre

Les **Insertions** n'apparaissent donc que dans les **Syntagmes** décrivant la Structure interne d'un composé (**Syntagme_NT_S**, attribut *insertion_l*). Elles sont insérées à leur place au milieu de la liste des **Positions** décrivant une Structure à un niveau N de son arbre de réécriture (place indiquée dans le champ *insereinsertion_l*).

Une Insertion est donc à la fois un pointeur vers une Position de la Construction externe et une marque du point d'insertion. Il peut y avoir plusieurs Insertions. Tout comme les Positions, elles sont numérotées à partir de 0.

Certaines **Insertions** sont obligatoires, alors qu'elles sont seulement possibles dans la plupart des cas (attribut *obligatoire* valant OUI ou NON).

7 : Conditions

7.1 : Propriétés générales

On veut pouvoir émettre des **Conditions** sur les **Positions** pour contraindre leur optionnalité ou l'ensemble de leurs occupants possibles. Pour ce faire, on se donne l'implication logique exprimée par les prédicats **Si** et **Alors**. Les implications étant orientées, on prendra toujours bien garde à émettre autant de Conditions que l'exige la couverture phénoménologique.

Les prédicats Si et Alors permettent de sélectionner une **Position** ou un Syntagme selon les besoins.

On peut exprimer la conjonction, la disjonction et la négation de Prédicats :

- La **conjonction** est exprimée par une liste de Prédicats (liste de Si , liste de Alors). Une Conjonction de Si est un ensemble de réalisations de surface à satisfaire. Une conjonction de Alors est l'ensemble des conséquences qui en résultent en surface : position(s) optionnelle(s) obligatoirement réalisée(s) ou absente(s), sélection ou exclusion d'un sous-ensemble des Syntagmes pouvant occuper une **Position**.

- La **disjonction** est exprimée par une liste de Conditions. Rappelons en effet que ces deux formules sont strictement équivalentes :

(1) Si (A ou B) Alors C

(2) (Si A alors C) et (Si B Alors C)

Dans GENELEX, on utilisera la formule (2).

- La **négation** est exprimée par le champ *negation* (OUI, NON) sur chaque Prédicat **Si** ou **Alors**. La négation d'une **Position** indique qu'elle n'est pas réalisée en surface. La négation d'un occupant de **Position** indique que •a n'est pas lui qui actualise la **Position**.

On distingue deux types de **Conditions** : Conditions sur l'**optionnalité** des Positions et Conditions sur les **réalisations** de Positions.

7.2 : Conditions sur l'optionnalité des Positions

Ces Conditions (**ConditionOpt**) permettent d'exprimer des contraintes sur l'optionnalité des **Positions**, grâce aux prédicats **SiOpt** et **AlorsOpt**. On rendra ainsi compte de phénomènes tels que :

- **Positions** optionnelles ensemble : l'absence d'une **Position** implique l'absence d'une autre.

Ex : P0 (P1) (P2)

Cond : Si !P1 Alors !P2

Cond : Si !P2 Alors !P1

- **Positions** mutuellement exclusives : l'occurrence d'une **Position** implique l'absence d'une autre et vice-versa.

Ex : P0 (P1) (P2)

Cond : Si P1 Alors !P2

Cond : Si P2 Alors !P1

- Toute autre contrainte plus ou moins complexe liée au fait qu'une **Position** optionnelle, lorsqu'elle est effectivement réalisée (ou non) en surface, peut avoir des incidences sur la réalisation effective des autres **Positions** optionnelles.

Ces **ConditionOpt** sélectionnent des **Positions** grâce à leur rang (champ *nieme_position* de **SiOpt** et **AlorsOpt**). Elles sont portées par l'élément Optionnalite qui peut comprendre zéro, une ou plusieurs ConditionOpt pour être totalement renseigné (cf. Positions optionnelles ensemble ou mutuellement exclusives). Rappelons que Construction et Syntagme_NT_C portent un champ *optionnalite*.

7.3 : Conditions sur les réalisations de Position

Certaines réalisations de **Positions** sont inter-conditionnées, c'est-à-dire que si P_i est réalisée de telle ou telle manière, alors P_j ne peut être réalisée que de telle ou telle autre. Ainsi, chaque réalisation de **Position** peut exclure ou au contraire sélectionner un sous-ensemble des **Syntagmes** pouvant occuper une autre **Position**.

Sélection et exclusion mutuelles devront toujours être exprimées par une double implication.

Ex : **Si** $P_i == \text{SyntagmeX}$ **Alors** $P_j = \text{SyntagmeY}$

Si $P_j == \text{SyntagmeY}$ **Alors** $P_i = \text{SyntagmeX}$

Un prédicat sélectionne :

- un Syntagme ou une **Position** de la **Construction** externe (valeur EXTERNE du champ *portee*),
- un **Syntagme** ou une **Position** du **Syntagme** structurel du **Self** (valeur INTERNE du champ *portee*), si l'Usyn est composée,
- une réalisation du Self en tant qu'intervenant de Construction (valeur INTERVENANT du champ *portee*).

La sélection ou l'exclusion se fait :

- pour un Syntagme gr%oce au **CheminSyntagme** :

celui-ci sélectionne une Position et un de ses Syntagmes (attributs *nieme_position* et *syntagme*) puis descend éventuellement plus bas dans la réécriture gr%oce à un lien récursif sur un autre CheminSyntagme.

- pour une Position gr%oce au **CheminPosition** :

celui-ci sélectionne une Position (attribut *nieme_position* de **PositionBut**) après être

éventuellement descendu dans la réécriture par un `CheminSyntagme`.

- pour une réalisation de **Self** en tant qu'intervenant de Construction grâce au **SelectIntervConst** :

celui-ci sélectionne un des Syntagmes référencés par l'`IntervConst` (attribut `syntagme_t`).

Ces Conditions sont portées par les Descriptions.

8 : Syntagmes

Les **Syntagmes** interviennent :

- dans la caractérisation de **Self**,
- dans la distribution d'une **Position**,

Les **Syntagmes** sont partageables.

Un **Syntagme** est défini par :

- son **étiquette syntagmatique** (attribut `etiquettesynt`),
- la liste (éventuellement vide) de ses **traits restrictifs** (attribut `trait_l`),
- s'il s'agit d'un Syntagme non terminal, la liste (éventuellement vide) des **Positions** décrivant sa réécriture (en vision syntaxiste) au premier niveau (attribut `position_c_l` ou `position_s_l`)
- un attribut `listepositions` (`OUVERTE`, `FERMEE`) qui indique si la liste des Positions de réécriture est fermée ou ouverte, en d'autres termes si la réécriture du Syntagme est totalement ou partiellement spécifiée. Lorsque la liste de Positions est vide il s'agit évidemment d'une liste `OUVERTE`.

On distingue deux sous-catégorisations des Syntagmes :

1. Syntagmes terminaux (**Syntagme_T**) dont l'étiquette syntagmatique est une catégorie morpho-syntaxique et Syntagmes non terminaux (**Syntagme_NT**). Les Syntagmes terminaux n'autorisent aucun des traits `Prep`, `Conj`, `ProRel`, `ProIntrog`, `Introd` de sorte que le seul trait lexical autorisé est le trait **Lex**. Cette règle s'applique par miroir aux traits de la famille `RefLex` (voir \boxtimes sur les Traits).
2. Syntagmes entrant dans la définition d'une Position de Construction (**Syntagme_C**) et Syntagmes entrant dans la définition d'une Position de Structure (**Syntagme_S**). Seuls les `Syntagme_S` acceptent des Insertions dans leur liste de Positions ainsi que des traits lexicaux de la famille `RefLex` sur leurs Syntagmes. Dans les deux cas, la liste de réécriture est une liste ordonnée. L'ordre des éléments dans la liste correspond à l'ordre canonique. Ces éléments pourront par la suite être référencés par leur rang dans la liste, indépendamment de leur type.

Ces deux distinctions peuvent bien sûr se conjuguer (**Syntagme_T**, **Syntagme_NT_C**, **Syntagme_NT_S**).

Les étiquettes syntagmatiques proposées par GENELEX sont les suivantes:

- **Étiquettes de Syntagmes non terminaux** : P, SN, Nbarre, SV, SADI, SADI, SP.
- **Étiquettes de Syntagmes terminaux** : NOM, ADJECTIF, ADVERBE, VERBE, PREPOSITION, CONJONCTION, PARTICULE, DETERMINANT, INTERJECTION, PRONOM.
- **catégorie e** : dit "élément vide" (empty category) en grammaire générative, applicable à un Syntagme terminal, qui permet d'exprimer les traces et peut donc supporter tous les traits restrictifs à l'exception des traits de sous-catégorie morphologique ou syntaxique.

Il s'agit d'une liste minimale qui garantit la généralité. La pauvreté de cette liste est compensée par la richesse des traits restrictifs qui peuvent être associés à l'étiquette syntagmatique. Toutefois cette liste peut être enrichie de nouvelles valeurs par l'utilisateur, mais elle perdra alors en généralité.

Les Syntagmes peuvent être décrits comme des arbres syntaxiques de profondeur N. Un arbre syntaxique s'exprime dans GENELEX par un enchaînement de listes de Positions dans les termes suivants : un Syntagme est réécrit au premier niveau par une liste de Positions, chaque Syntagme occupant une Position pouvant à son tour être réécrit par une liste de Positions, etc. Dans le cas où un Syntagme est réécrit par une liste de Positions, toutes les remarques données plus haut pour les champs *optionnalite*, *inseeself* et *solidarite* de la Construction valent.

9 : Traits

Combinés aux différents Syntagmes, les **traits restrictifs** devraient permettre en l'état de spécifier aussi précisément que possible n'importe quel occupant de Position

Ex: P[SsCatSynt:RELATIVE] => relative

P[Mode:INFINITIF] => infinitive

P[SsCatSynt:COORDONNE] => phrase coordonnée

P[SsCatSynt:SUBORDONNEE] => subordonnée

P[SsCatSynt:COMPLETIVE]

[Mode:SUBJONCTIF] => complétive

On donne aux traits une apparence graphique (interface utilisateur) qui ne préjuge en rien de leur structuration interne. Les traits sont notés entre crochets carrés, à l'intérieur desquels on spécifie d'abord le nom du trait, puis après deux points, sa valeur. Seuls quelques traits appellent une

notation plus complexe, il s'agit de Aux, et de tous les traits de la famille des RefLex.

Ex: Trait_Genre

[Genre : FEMININ]

Trait_Aux

[Aux : ETRE [Mode : PARTICIPE] [Temps : PRESENT]]

Trait_RefLex

[Reflex : 1.3]

Le nom du Trait figurant dans l'apparence est déduit de son type ou spécifié lors de la création du Trait.

Ex: Trait_Personne

nom : Personne

Trait_Libre

nom : couleur

Trait_Bin

nom : animé

9.1 : Traits lexicaux

9.1.1 : Famille des traits Lex

Ces traits donnent des informations de lexicalisation en spécifiant une Um par sa graphie et/ou par son identifiant (ce qui lève tout problème d'homographie). Cette référence directe (identifiant SGML) ou indirecte (graphie) à une Um est faite à travers deux champs:

valeur : *string*

um : *pointeur*

NB: On autorise un écart (fondé sur l'écart entre catégorie morfo-syntaxique et catégorie fonctionnelle) entre la catégorie du Syntagme qui supporte Trait_Lex et la catégorie de l'Um référée

par Trait_Lex.

Ex : Um id:um04 CatGram:ADJECTIF Umg:courageux

NOM[Lex:courageux]

A l'intérieur de cette famille, on doit encore distinguer les introducteurs et le trait Lex à proprement parler.

1. Introducteurs: **Prep, Conj, ProRel, ProIntrog, Introd**

Ces traits lexicalisent des introducteurs de Syntagmes non terminaux:

- Prep pour la préposition recatégorisant un SN, introduisant une Phrase ou étant le premier élément de réécriture d'un Syntagme Prépositionnel.
- ProRel pour le pronom relatif introduisant une Phrase.
- ProIntrog pour le pronom interrogatif introduisant une Phrase.
- Conj pour la conjonction introduisant une Phrase.
- Introd pour tout autre introducteur de Syntagme non terminal que nous n'aurions pas prévu.

2. Lex

Ce trait lexicalise:

- une feuille syntagmatique, s'il s'applique à un Syntagme de catégorie terminale
- la tête d'un Syntagme, s'il s'applique à un Syntagme de catégorie non terminale. Puisqu'on s'est par ailleurs donné les introducteurs, on admettra de fait que les têtes syntagmatiques sont:

N pour SN

N pour SP

ADJ pour SADJ

ADV pour SADV

V pour SV

V pour P si l'on admet que $P=V'''$ dans la théorie X-barre.

L'attribut *saturesynt* du trait **Lex** indique si l'élément lexicalisant est ou non (en incluant un éventuel introducteur), la seule feuille de ce syntagme. En conséquence, cet attribut :

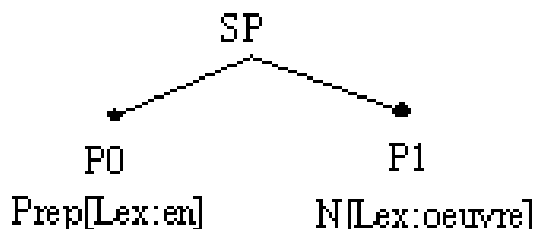
- doit toujours valoir *OUI* (valeur par défaut) si le syntagme est de catégorie terminale,
- peut valoir *OUI* ou *NON* pour les syntagmes de catégorie non terminale pour lesquels il a été conçu.

Ex : en Œuvre

Solution 1 :

SP [P_{prep}:en][Lex:oeuvre[saturesynt:OUI]]

Solution 2 :



9.1.2 : Famille des traits RefLex

Ces traits sont définis en miroir des traits de la famille Lex. Dans cette famille de traits on trouve donc : **RefLex**, **RefPrep**, **RefProRel**, **RefProIntrog**, **RefConj** et **RefIntrod**.

Ce qui les distingue, c'est :

- le mode d'accès aux unités,
- le type des unités accédées qui sont lexicalisantes (Um) et/ou structurantes (Usyn),
- le fait qu'ils ne sont utilisés qu'à l'intérieur des Structures décrivant l'intérieur des composés syntaxiques.

Les attributs *nieme_cposition* et *nieme_cposant* réfèrent par leurs indices des Um ou des Usyn contenues dans une liste à deux niveaux : liste des alternatives (**Composition**) portée par l'Usyn composée, liste des composants (**R_ComposeUm**, **R_ComposeUsyn**).

nieme_cposition : *integer*

nieme_cposant : *integer*

Ex : ((avoir admiration pour)(porter admiration à))

RefLex[2.1] => porter

9.2 : Traits morphologiques

• Trait_Mode

valeur : INDICATIF, SUBJONCTIF, CONDITIONNEL, PARTICIPE, INFINITIF, IMPERATIF.

Ex : [Mode : INFINITIF]

• Trait_Temps

valeur : PRESENT, IMPARFAIT, FUTUR, PASSE, PASSE_SIMPLE, COMPOSE.

Ex : [Temps : PASSE]

Le trait de temps permet d'exprimer les restrictions de temps liées à certaines tournures.

• Trait_Personne

valeur : 1, 2, 3

Ex : [Personne : 1]

• Trait_Genre

valeur : masculin, féminin, neutre.

Ex : [Genre : FEMININ]

• Trait_Nombre

valeur : singulier, pluriel.

Ex : [Nombre : SINGULIER]

- **Trait_NombrePosseur**

valeur : SINGULIER_POSSEUR, pluriel_POSSEUR.

Ex : [NombrePosseur : SINGULIER_POSSEUR]

9.3 : Traits morpho-syntaxiques

- **Trait_Accord**

valeur : I, J, K, L, NON_I, NON_J, NON_K, NON_L.

Ex : [Accord:I]

cf. Coref

- **Trait_SsCatMorph**

valeur : PROPRE, COMMUN, POSSESSIF, DEMONSTRATIF, PARTITIF, DEFINI, INDEFINI, INTERROGATIF, CARDINAL, ORDINAL, RELATIF, PERSONNEL_FORT, PERSONNEL_FAIBLE, IMPERSONNEL, EXCLAMATIF, QUALIFICATIF, COORDINATION, SUBORDINATION, COMPLETIF, COMPARATIF_EGALITE, COMPARATIF_SUPERIORITE, COMPARATIF_INFERIORITE, SUPERLATIF_ABSOLU, SUPERLATIF_SUPERIORITE, SUPERLATIF_INFERIORITE.

Ex : [SsCatMorph:DEFINI]

- **Trait_Aux**

valeur : ETRE, AVOIR.

temps : PRESENT, IMPARFAIT, FUTUR, PASSE, PASSE_SIMPLE, COMPOSE.

mode : INDICATIF, SUBJONCTIF, CONDITIONNEL, PARTICIPE, INFINITIF, IMPERATIF.

Ce trait permet d'associer à un verbe donné (l'entrée décrite ou bien un verbe dans le contexte de l'entrée) son ou ses auxiliaires autorisés pour un emploi.

Ex : *se lever* (être levé) // *lever* (avoir levé)

V[Lex:lever] V[Lex:lever]

[Aux : ETRE] [Aux : AVOIR]

[Pronominal : SE]

L'apparition du Trait_Aux en conjonction avec un Trait_Temps portant la valeur COMPOSE indique que l'auxiliaire est nécessairement présent dans l'emploi que l'on décrit.

Ex : être arrivé socialement

V[Lex : arriver]

[Aux : ETRE]

[Temps : COMPOSE]

Les attributs temps et mode du Trait_Aux précisent lorsque c'est nécessaire le temps et le mode de l'auxiliaire lui même.

Ex : étant donné

V[Lex : donner]

[Temps : COMPOSE]

[Aux : ETRE [Mode : PARTICIPE] [Temps : PRESENT]]

• Trait_Pronominal

valeur : SE, LE, LA, LES, EN, Y, SE_LE, SE_LA, SE_LES, SE_Y, SE_EN.

Ce trait permet d'associer à un verbe donné (l'entrée décrite ou bien un verbe dans le contexte de l'entrée) la particule préverbale NON REFERENTIELLE (Cf. "vrais" pronominaux) qui doit être associée à un emploi.

Ex : s'en aller

V[Lex : aller]

[Pronominal : SE_EN]

• Trait_Passif

valeur : PLUS, MOINS.

Ex : [Passif:PLUS]

9.4 : Traits syntaxiques

• Trait_Neg

valeur : LIBRE, NE, NE_PAS, NE_JAMAIS, NE_PLUS, NE_GUERE, NE_POINT, NE_MAIS, NE_QUE, NE_PAS_QUE, NE_JAMAIS_QUE, NE_PLUS_QUE, NE_GUERE_QUE, NE_RIEN_QUE.

La présence d'un Trait_Neg indique que l'emploi décrit est à la forme négative. On peut de plus préciser dans le champ valeur une restriction sur la lexicalisation de la Négation

Ex : [Neg:NE_PAS]

• Trait_SsCatSynt

valeur : RELATIVE, COMPLETIVE, COORDONNE, SUBORDONNEE, EXCLAMATIVE, INTERROGATIVE_DRI, INTERROGATIVE_DRD, TEMPS, LIEU, MANIERE, DEGRE, COPULE, DET_VIDE, DETERMINATIF, QUANTIFIEUR, COMPARATIF, SUPERLATIF.

Ex : [SsCatSynt:COMPLETIVE]

• Trait_Tournure

valeur : INTERROGATIVE, EXCLAMATIVE.

Ex : [Tournure:INTERROGATIVE]

9.5. : Traits syntaxico-sémantiques

• Trait_Coref

valeur : I, J, K, L, NON_I, NON_J, NON_K, NON_L

Ex : [Coref:I]

Les indices de coréférence pourront être utilisés à tout moment. Ils pourront comme tout trait s'appliquer à n'importe quel occupant de Position. Mais la coréférence doit toujours être résoluble : si un trait de valeur I est présent, il existe au moins un autre trait de valeur I ou NON_I lui répondant.

Les traits Coref ne forcent pas la co-réalisation des Syntagmes les portant ; si l'on souhaite imposer cette co-réalisation, on le fera comme d'habitude par l'usage de Conditions.

Lorsqu'un trait de Coref porte sur la catégorie P, il désigne implicitement le sujet de P.

Ex : *Jean dit partir en vacances*

cb : P0 SELF P1

P0 : SN[Coref:I]

...

P1 : P[Mode:INFINITIF][Coref:I]

...

9.6 : Traits sémantiques

- **Trait_Aspect**

valeur : STATIF, PROCESSIF, RESULTATIF.

Ex : [Aspect:STATIF]

- **Trait_Libre**

nom : *string*

valeur : *string*

Ce type de trait pourra être exploité pour spécifier des classes ou des familles sémantiques.

Ex : [couleur:rouge]

[classe:vêtement]

- **Trait_Bin**

nom : *string*

valeur : PLUS, MOINS.

Ce type de trait permet par exemple d'exprimer les "conditions dénotationnelles".

Ex : [animé:PLUS]

10 : Transformations

Les attributs *appellation*, *commentaire*, *exemple* portés par les **Transf** glosent les transformations elles-mêmes et non l'élément à transformer.

10.1 : TransfUsyn et TransfDescription

Le processus de Transformation décrit par les TransfUsyn et les TransfDescription est à la base le même. On rend compte de trois phénomènes :

1 : modification des réalisations de Self. (attribut *modifintervconst*)

Ex : Self : IntervConst : V

--passivation-->

Self : IntervConst : V[Aux:ETRE]

[Temps:COMPOSE]

[Passif:PLUS]

2 : modification de la Construction (attribut *modifconstruction*).

Ex : P0[RoleTh:AGENT] P1[RoleTh:THEME]

SN SN

--passivation-->

P0[RoleTh:THEME] P1[RoleTh:AGENT]

SN SP[Prep:par]

3 : restructuration du Syntagme structurel décrivant l'intérieur d'un composé (attribut *modifsyntagme_nt*)

Ex : pour les beaux yeux de SN

-> pour ses beaux yeux

Mais dans le cas des **TransfDescription**, on travaille sous la même **Usyn**, alors que dans celui des **TransfUsyn** on travaille sous deux **Usyn** différentes, ceci pour des raisons linguistiques.

1. **TransfUsyn** : Transformations opérant entre deux Unités Syntaxiques

- issues de la même **Um**,

Ex : neutralité

- issues d'**Um** différentes

Ex : dérivation syntaxique

2. **TransfDescription** : Transformation opérant entre deux Descriptions (c'est-à-dire deux couples **Self / Construction**).

Ex : passivation

Formellement par ailleurs, **TransfUsyn** et **TransfDescription** sont toutes deux pointées par l'**Usyn** origine. Mais :

- **TransfUsyn** pointe sur l'**Usyn** résultat,
- **TransfDescription** pointe d'une part sur la Description d'origine (de base ou transformée pour pouvoir rendre compte des changements de Transformation) d'autre part sur la Description résultat.

De plus, les contraintes sur la cardinalité des listes ne sont pas les mêmes :

- une **Usyn** peut porter 0 à N **TransfUsyn** dont elle est l'origine.
- la cardinalité de la liste de **TransfDescription** (*transfdescription_l*) est supérieure ou égale (car une même Description peut résulter de deux Transformations différentes) à celle des Descriptions transformées (*description_l*). Et si N est le nombre de Descriptions transformées, alors le nombre de **TransfDescription** est inférieur ou égal au nombre de combinaisons possibles de deux Descriptions parmi N+1 (puisque l'on doit aussi compter la base).

10.2 : **TransfUsyn**

TransfUsyn met en relation une **Usyn** origine, donnée dans le contexte d'appel de **TransfUsyn**, et une **Usyn** résultat pointée par l'attribut *usyn_resultat* de **TransfUsyn**.

TransfUsyn permet éventuellement de donner des informations sur le PASSAGE de l'origine au résultat en renseignant un **ModifDescription** sur la Description de base de l'Usyn origine .

10.3 : TransfDescription

TransfDescription met en relation de transformation une **Description** origine et une **Description** résultat, respectivement pointées par les attributs *description_origine* et *description_resultat*.

En ce qui concerne la Transformation de la **Construction** contenue dans la **Description**, on peut choisir entre un mode descriptif (usage minimal) ou calculatoire (usage maximal) :

- en mode descriptif, la **Construction** pointée par la Description résultat est totalement décrite : le **ModifConstruction** met alors en correspondance des Positions ou des Syntagmes à des niveaux hiérarchiques quelconques entre **Construction** origine et **Construction** résultat.
- en mode calculatoire, la **Construction** pointée par la **Description** résultat est déclarée *squelettique* : la **Construction** résultat à proprement parler n'est pas enregistrée. Seul un squelette de résultat l'est. Celui-ci est habillé avec des éléments ou attributs provenant de la Construction origine ou donnés dans le **ModifConstruction**. Il faut remarquer qu'en mode calculatoire on ne peut pas enregistrer de chaîne de transformations.

10.4 : TransfPosition

TransfPosition met en relation de transformation une **Position** origine et une **Position** résultat, respectivement pointées par deux **CheminPosition**.

Pour transformer une Position (**TransfPosition**) en mode calculatoire, on identifie une Position origine (premier CheminPosition) et une Position du squelette (deuxième CheminPosition) entre lesquelles on veut établir une relation de Transformation. Le nombre de TransfPosition est inférieur ou égal au nombre de Positions du squelette.

10.5 : TransfSyntagme

Les Syntagmes occupant une même Position ou caractérisant Self peuvent entretenir des relations de Transformation, les plus typiques étant la pronominalisation et la réduction de complétive en infinitive.

Ex : P0 *aimer* P1

P1 : SN

PRONOM[Lex:le]

[SsCatMorph:PERSONNEL_FAIBLE]

P[Mode:INFINITIF]

PRONOM [Lex : ça]

Ces propriétés transformationnelles n'étant valides que dans le contexte d'une Position (ça n'est pas une propriété du Syntagme proprement dit), elles sont portées par la Position et non par le Syntagme, la Position indiquant quels sont les couples de Syntagmes qui, l'occupant, sont reliés par une relation de Transformation.

Si N est le nombre de Syntagmes entrant dans une même distribution, alors le nombre de TransfSyntagme est inférieur ou égal au nombre de combinaisons possibles de deux Syntagmes parmi N.

TransfSyntagme met en relation de transformation un **Syntagme** origine et un **Syntagme** résultat, respectivement pointés par les attributs *syntagme_origine* et *syntagme_resultat* et permet éventuellement de donner des informations sur le PASSAGE de l'origine au résultat en renseignant un ModifSyntagme sur l'origine (attribut *modifsyntagme*).

TransfSyntagme sert également à décrire l'éventuelle transformation subie par un Syntagme lors d'une Transformation de la Construction o• il appara"t. Dans ce cas, *syntagme_origine* pointe sur un Syntagme de la Construction origine et *syntagme_resultat* pointe sur un Syntagme de la Construction résultat.

11 : Modifs

Les éléments dont le nom commence par Modif servent à modifier les valeurs d'attributs d'un élément dit "origine" (avant la modification). Un couple Modif + élément origine (connu en contexte) doit permettre de déduire un élément "résultat" (après application des modifications spécifiées).

Tous ou seulement certains attributs d'un élément peuvent être modifiés suivant un protocole qui dépend du type de l'attribut :

- attributs IDREF(S), CDATA et NUMBER(S). Lorsqu'ils ne sont pas renseignés dans le Modif, ils sont hérités tels quels de l'origine. Lorsqu'ils sont renseignés, la nouvelle valeur écrase l'ancienne.

- **Cas particulier** pour certains IDREFS, le mécanisme par double liste (**retire + ajoute**) : lorsqu'aucun élément n'est pointé dans le Modif ni par le retire ni par l'ajoute, on hérite de celui (ceux) de l'origine. Sinon, on obtient une nouvelle valeur d'attribut par retrait et ajout d'élément(s) sur la base de la valeur de l'attribut origine.

- attribut à valeurs énumérées. Lorsque la valeur est HERITAGE dans le Modif, ces champs sont hérités tels quels de l'origine. Toute autre valeur écrase l'ancienne valeur.

11.1 : ModifDescription

Etant donné une **Description** regroupant par définition une **Construction** et un **Self**, **ModifDescription** permet de modifier :

- la **Construction**, par appel de **ModifConstruction** (attribut *modifconstruction*)
- l'**IntervConst** de **Self**, par appel de **ModifIntervConst** (attribut *modifintervconst*)
- le **Syntagme** figurant la Structure interne d'un **Self** composé, par appel de **ModifSyntagme_NT** (attribut *modifsyntagme_nt*).

11.2 : ModifConstruction

Etant donné une **Construction**, **ModifConstruction** permet de la modifier complètement, c'est-à-dire de :

- réécrire sa glose, qui recouvre les attributs *appellation*, *commentaire*, *exemple* ;
- modifier son *etiquettesynt*, son *optionnalite* et sa *solidarite* ;
- spécifier les traits restrictifs à lui retirer et/ou ajouter (*ajoute_trait_l*, *retire_trait_l*) ;
- changer le point d'insertion de **Self** (attribut *inseeself*) ;
- supprimer (attribut *retire_position_l*) ou modifier (élément **TransfPosition**) des **Positions**. Dans ce cas, il faudra s'assurer que les attributs *optionnalite*, *inseeself* et *solidarite* sont cohérents avec la nouvelle liste de **Positions**.

11.3 : ModifPosition

Permet de modifier complètement une **Position**, c'est-à-dire :

- sa glose, qui recouvre les attributs *appellation*, *commentaire*, *exemple* ;
- ses valeurs de *repetabilite*, *fonction* et *roleth_l* ;
- sa distribution, en :
 - lui retirant et/ou ajoutant des **Syntagmes**,
(attributs *retire_syntagme_l* et *ajoute_syntagme_l*),
 - lui retirant et/ou ajoutant des **Transformations** entre ces **Syntagmes**,
(attributs *retire_transfsyntagme_l* et *ajoute_transfsyntagme_l*).
- modifiant des syntagmes l'occupant par Transformation sur ces **Syntagmes** (attribut

transfsyntaxme_l).

11.4 : ModifSyntaxme

Etant donné un **Syntaxme** donné par le contexte d'appel de **ModifSyntaxme**, **ModifSyntaxme** permet :

1. Pour n'importe quel type de Syntaxme (**Syntaxme_T**, **Syntaxme_NT**, **Syntaxme_NT_S**) :
 - de spécifier les **Traits** restrictifs à lui retirer et/ou ajouter (attributs *retire_trait_l* et *ajoute_trait_l*),
2. Pour un Syntaxme non terminal (**Syntaxme_NT**) :
 - de modifier son étiquettesynt, son *optionnalite* et sa *solidarite* ;
 - de supprimer (attribut *retire_position_l*) ou modifier (élément **TransfPosition**) des **Positions** de réécriture s'il y en a. Dans ce cas, il faudra s'assurer que les attributs *optionnalite*, *inseself* et *solidarite* sont cohérents avec la nouvelle réécriture.
3. Pour un Syntaxme de Construction (**Syntaxme_NT_C**) :
 - de changer le point d'insertion de **Self** (attribut *inseself*) ;
4. Pour un Syntaxme de Structure (**Syntaxme_NT_S**) :
 - de supprimer (attribut *retire_insertion_l*) ou modifier (élément **TransfInsertion**) les **Insertions** s'il y en a.
 - de changer le point d'insertion des Insertions s'il y en a (attribut *insereinsertion_l*).

Un **ModifSyntaxme** fait virtuellement une copie du **Syntaxme** à modifier. En conséquence, si deux **ModifSyntaxme** travaillent sur le même **Syntaxme**, il faut comprendre qu'ils permettent de produire deux **Syntaxmes** résultats sur la base d'un **Syntaxme** origine.

11.5 : ModifIntervConst

Sachant que **Self** est défini par un seul **IntervConst** qui décrit ses caractéristiques en tant qu'intervenant dans la Construction, **ModifIntervConst** permet de :

- modifier sa *fonction* et ses *rôles thématiques*,
- retirer et/ou d'ajouter des **Syntaxmes** à ceux qui définissent **Self**.
- modifier ces **Syntaxmes** par **Transformation**.

12 : R_Compose et Composition

R_Compose et **Composition** rendent compte de la lexicalisation des **Usyn** composées. Les **R_Compose** spécifient les composants, les **Compositions** spécifient les alternatives de lexicalisation. Chaque **Composition** contient donc une liste de **R_Compose**.

Ex : [(avoir admiration)

(éprouver admiration)

(porter admiration)]

3 Compositions comprenant chacune 2 R_Composes

Ces deux niveaux de liste sont utilisés dans les traits de la famille RefLex qui référencent indirectement les éléments de ces listes :

- soit dans le Syntagme structurel interne du Self,
- soit dans le Mode de composition MdC

selon la syntaxe suivante :

```
[RefLex:[nieme_composition:i][nieme_composant:j]]
```

Les listes sont numérotées à partir de 1, le 0 sur nieme_composition étant réservé à la sélection de l'ensemble des alternatives d'un élément de la liste.

Ex : RefLex[1.1] = avoir

RefLex[0.1] = avoir/éprouver/porter

Les composants pointés par R_Compose peuvent être de deux types :

- composant Um,
- composant Usyn et son éventuel MdC.

A chacun des types de composants correspond un sous-type de R_Compose : R_ComposeUm et R_ComposeUsyn.

L'attribut *type* indique si le composant est appelant seulement, appelant et appelé ou appelé seulement. La tête du composé est seulement appelante.

Un composant de type Um est toujours appelé.

13 : Mode de Composition

13.1 : MdC

Le Mode de Composition (**MdC**) permet de spécifier l'organisation mutuelle des Usyn composantes, en indiquant quel composant sature quelle **Position** d'un autre composant, et d'émettre des contraintes d'inhibition, de filtrage ou d'héritage des propriétés de l'Usyn composante pour ce composé.

Etant donné un composant de type Usyn, on peut lui associer un MdC pour pouvoir hériter totalement ou partiellement de son schéma de complémentation ou de son contexte d'occurrence ; ce MdC permet de :

- spécifier a minima les **Positions** dont on veut hériter et filtrer éventuellement ces **Positions** et leurs occupants ;
- spécifier la lexicalisation de ces **Positions** par d'autres composants (seulement dans le cas d'un comportement d'appelant) grâce aux traits de la famille RefLex sur les occupants de Position.
- filtrer éventuellement les propriétés du Self composant.

Pour un composé syntaxique sur lequel on veut spécifier toutes ces informations, il y a autant de MdC que de composants de type Usyn.

13.2 : HeritePosition

Toute **Position** non référencée par **HeritePosition** est considérée inhibée. **HeritePosition** est nécessaire pour indiquer qu'une **Position** est héritée lors de la composition, et spécifie :

- quelle est la **Position** concernée (élément **CheminPosition**),
- • cette **Position** est héritée, c'est-à-dire à l'*INTERIEUR* ou à l'*EXTERIEUR* du composé (attribut *destination*).
- comment cette **Position** est héritée, c'est-à-dire avec ou sans filtrage (attributs *optionnel* et *modifposition*).

Ces modifications sur la **Position** origine peuvent affecter tous les attributs qui la définissent ainsi que son *optionnalité*. Elles sont exprimées sur **HeritePosition** par :

- l'attribut *optionnel* qui a pour valeur *HERITAGE*, *OUI*, *NON*. La valeur *HERITAGE* indique qu'on doit hériter de la valeur d'optionnalité définie pour la **Position** dans le champ optionnalité de sa Construction. Les valeurs *OUI* et *NON* écrasent les valeurs d'optionnalité de la **Position** origine.
- l'attribut *modifposition* qui réfère à l'élément **ModifPosition**. Celui-ci factorise les modifications à porter à tous les attributs d'une **Position** sachant que tout ce qui n'est pas

modifié est hérité tel quel.

Lorsqu'une **Position** est héritée à l'intérieur ET à l'extérieur du composé, il faut lui associer deux **HeritePosition**, l'un pour l'intérieur, l'autre pour l'extérieur.

Lorsqu'une **Position** n'est héritée qu'à l'intérieur du composé, on dit dans la vision atomiste qu'elle est saturée par le composant qui vient l'occuper.

Si toutes les **Positions** d'un niveau supérieur à Self (en vision syntaxiste avec arbre de réécriture) sont inhibées, alors le niveau o• elles apparaissent est inhibé, sans dire quoi que soit de l'inhibition des niveaux qui lui sont inférieurs. Si toutes les **Positions** d'un niveau inférieur à Self sont inhibées, alors le niveau o• elles apparaissent est inhibé et tous les niveaux encore inférieurs à ce niveau sont inhibés.

13.3 : FiltreSelf

Par défaut, l'**IntervConst** et le **Syntaxme_NT_S** du **Self** de l'**Usyn** composante sont hérités TELS QUELS lors du processus de composition et il n'y a alors aucun héritage sur Self à spécifier.

Si l'on veut en hériter par filtrage (pour les modifier sensiblement lorsque le composant remplit le composé), il faut définir un **FiltreSelf**.

FiltreSelf permet donc de spécifier l'héritage de **Self** par filtrage, ce qui consiste à reprendre ses **IntervConst** et son **Syntaxme_NT_S** (s'il est lui-même un composé) pour les modifier respectivement par **ModifIntervConst** et **ModifSyntaxme_NT**.

14 : Structure (Syntaxme_NT_S)

Une **Structure** est portée par l'élément **Self**. Elle sert à décrire la structuration interne d'un composé.

Les **Structures** partagent toutes les caractéristiques des **Syntaxmes**, à ceci près que :

- leurs listes de **Positions** de réécriture sont a priori entièrement spécifiées, le champ *listepositions* devrait donc toujours valoir FERMEE
- elles contiennent en plus une liste d'**Insertions** (Insertion de **Positions** de la structure externe dans la structure interne),
- les traits lexicaux portés par les **Syntaxmes** occupants les **Positions** sont de la famille Lex ou **RefLex**.
- leur champ *solidarite* contient éventuellement la référence aux **Insertions** mais exclut Self.

Tout comme les **Positions**, les **Insertions** sont numérotées à partir de 0. La place des **Insertions** au milieu des **Positions** est notée dans le champ *insereinsertion_l*

- Vision atomiste

Ce champ n'est pas renseigné.

- Vision syntaxiste

Le champ *insereinsertion_l* indique quel est le point d'insertion de chaque *Insertion* en début ou milieu de la liste des **Positions**. Le numéro donné pour chacune en respectant leur ordre canonique est l'indice de la **Position** DEVANT laquelle elles s'insèrent. Si plusieurs Insertions sont placées devant la même **Position**, c'est qu'elles se suivent dans cet ordre.

15 : Appellation, commentaire, exemple

Les champs *appellation*, *commentaire* et *exemple* permettent au lexicographe de gloser l'élément décrit, d'en accélérer l'accès et d'en faciliter la maintenance. L'*appellation* permet d'associer à un élément GENELEX le nom qui est en usage dans la tradition grammairienne, le monde linguistique ou encore l'équipe lexicographique. Ainsi une construction pourra porter l'*appellation* "transitif direct". Le *commentaire* permet d'indiquer dès le niveau le plus haut les particularités d'un élément sans avoir à le parcourir entièrement. Il permet aussi de noter toutes les informations relatives au travail lexicographique lui-même. Ainsi le statut "en cours", "à valider", "validé" d'un élément pourra être enregistré à cet endroit, tout comme les justifications éventuelles du choix de codage. Enfin l'*exemple* devrait parler de lui-même...

G - SCHEMAS ENTITES-RELATIONS

CONVENTIONS graphiques particulières

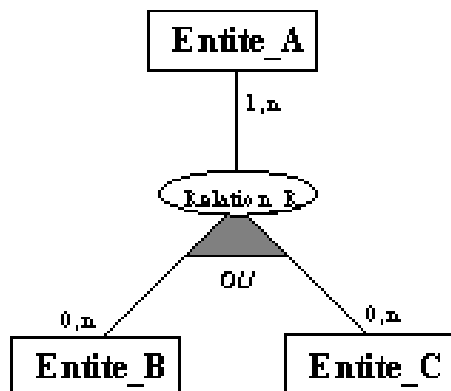
utilisées dans les Schémas Entités/Attributs/Relations de ce document:



Une Entité ainsi encadrée (en grisé) est décrite ailleurs dans les schémas de ce document.

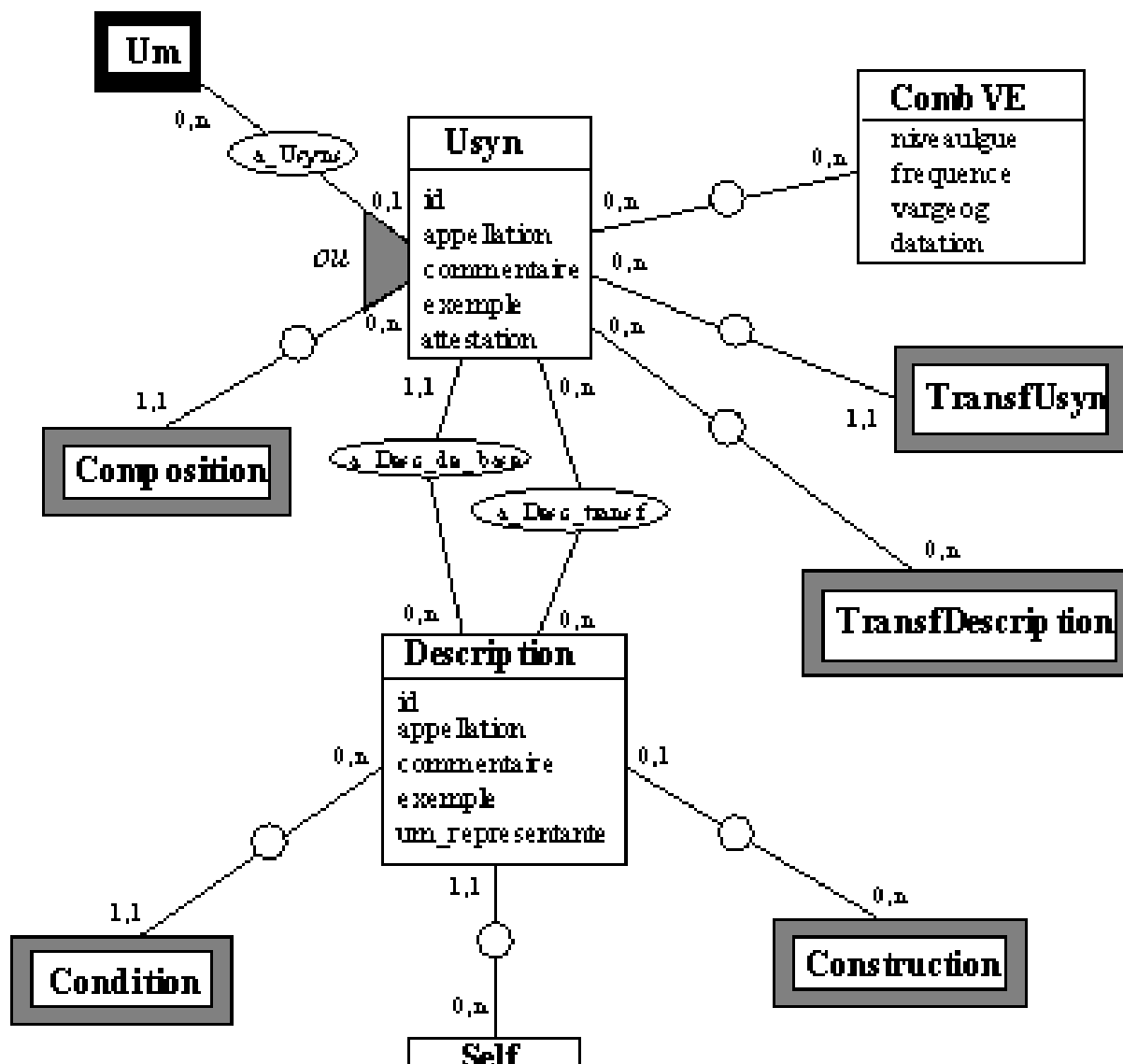


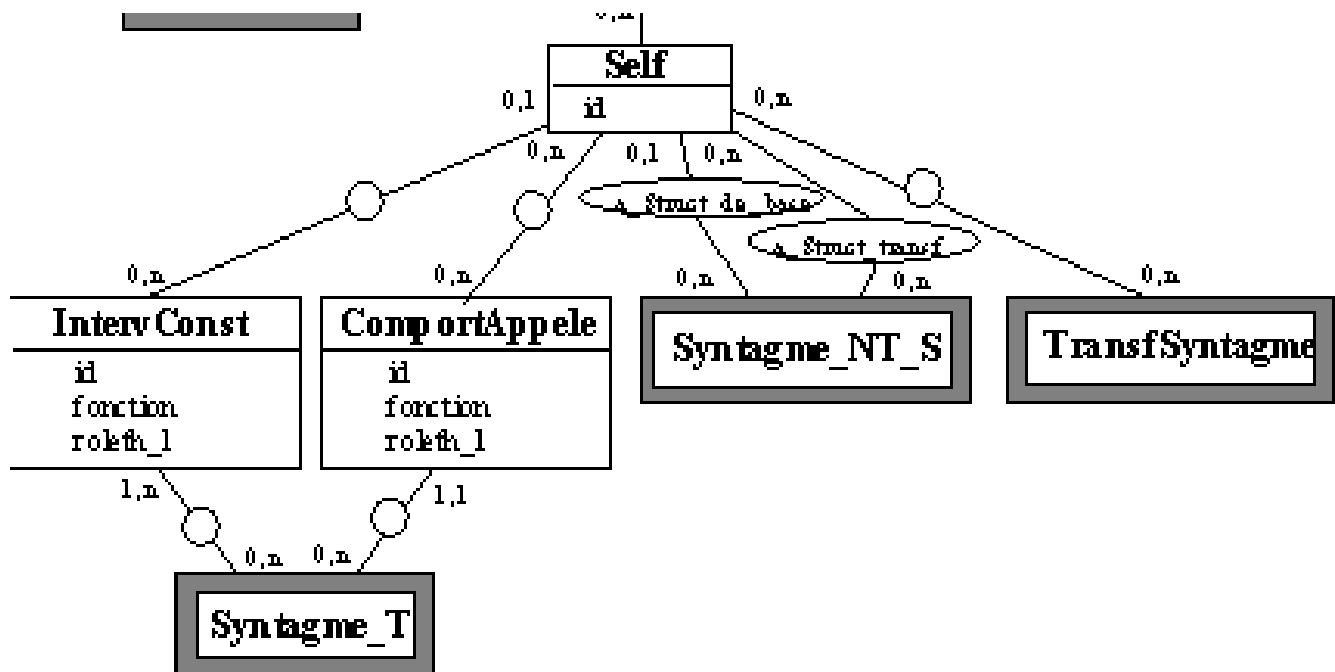
Une Entité ainsi encadrée (en noir) est décrite dans les schémas d'un autre document GENELEX (soit dans le document MORPHOLOGIE soit dans un futur document, SEMANTIQUE ou autre).



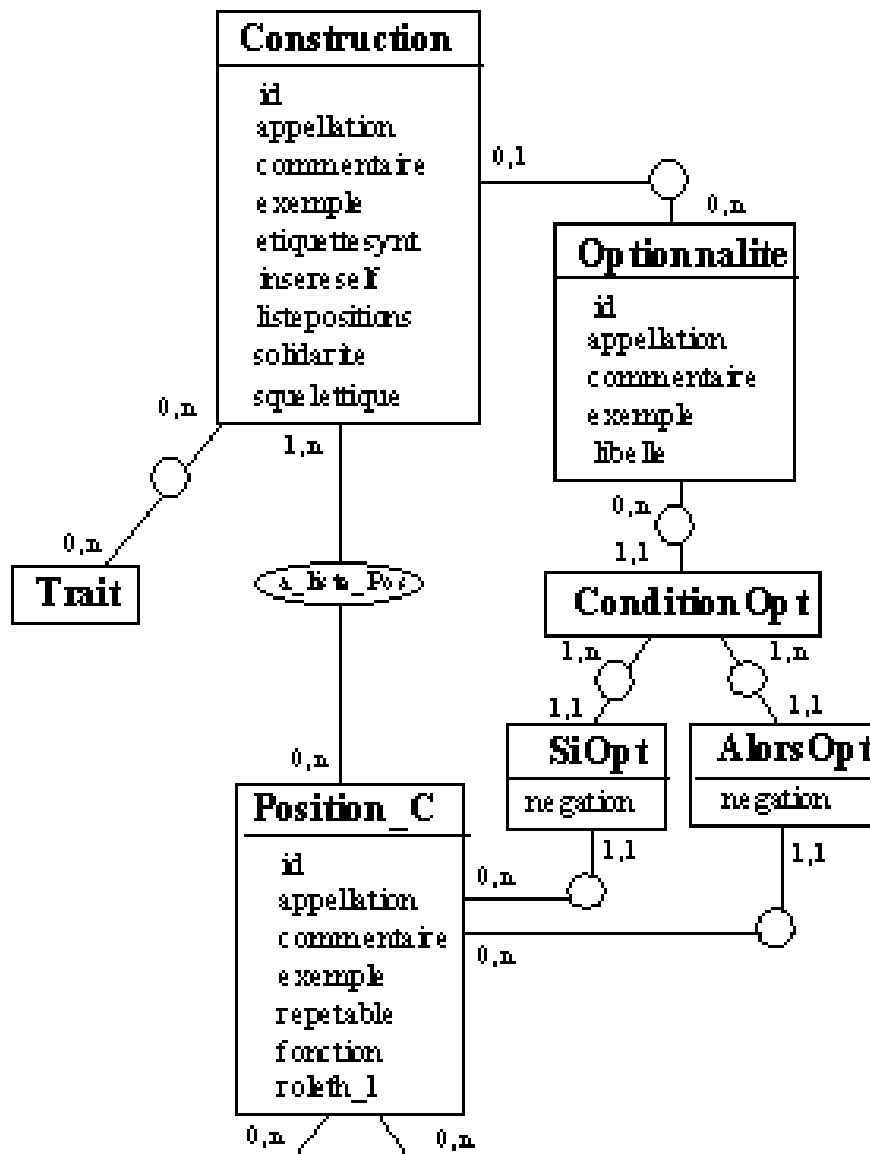
Le triangle grisé marqué "OU" signifie ici que l'Entité Entité_A est en Relation (Relation_R) avec une ou plusieurs Entités qui peuvent être des Entité_B OU des Entité_C.

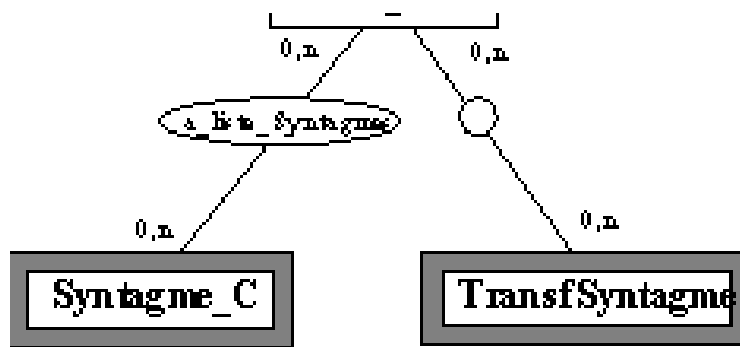
1 : Unité Syntaxique, Description et Self



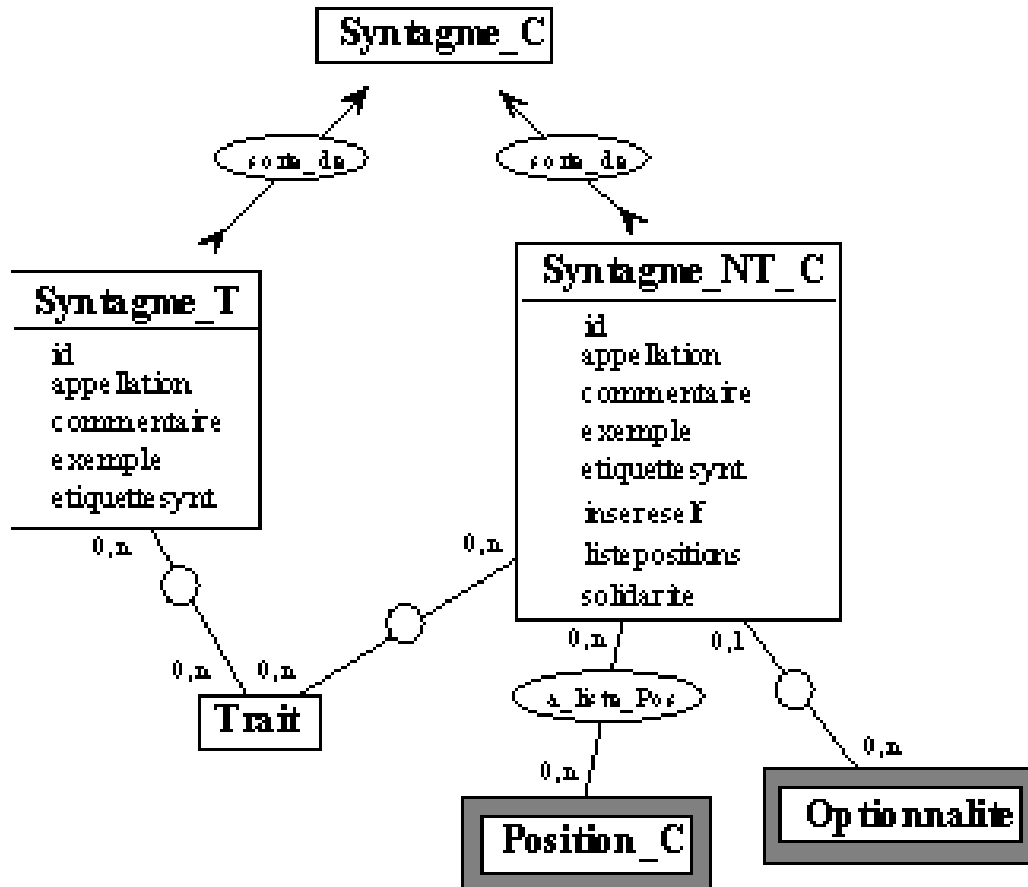


2 : Construction, Position_C et Optionnalite

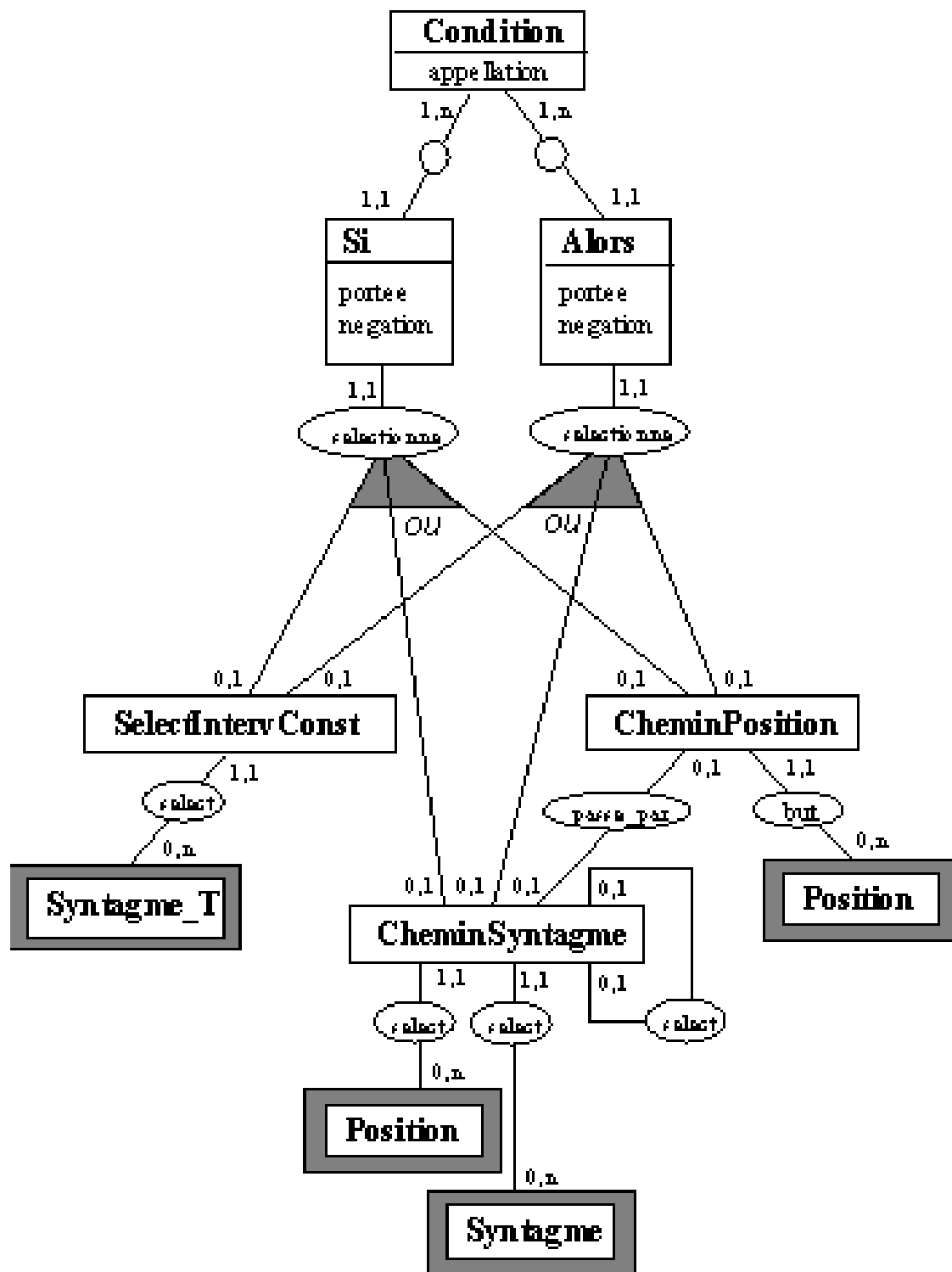




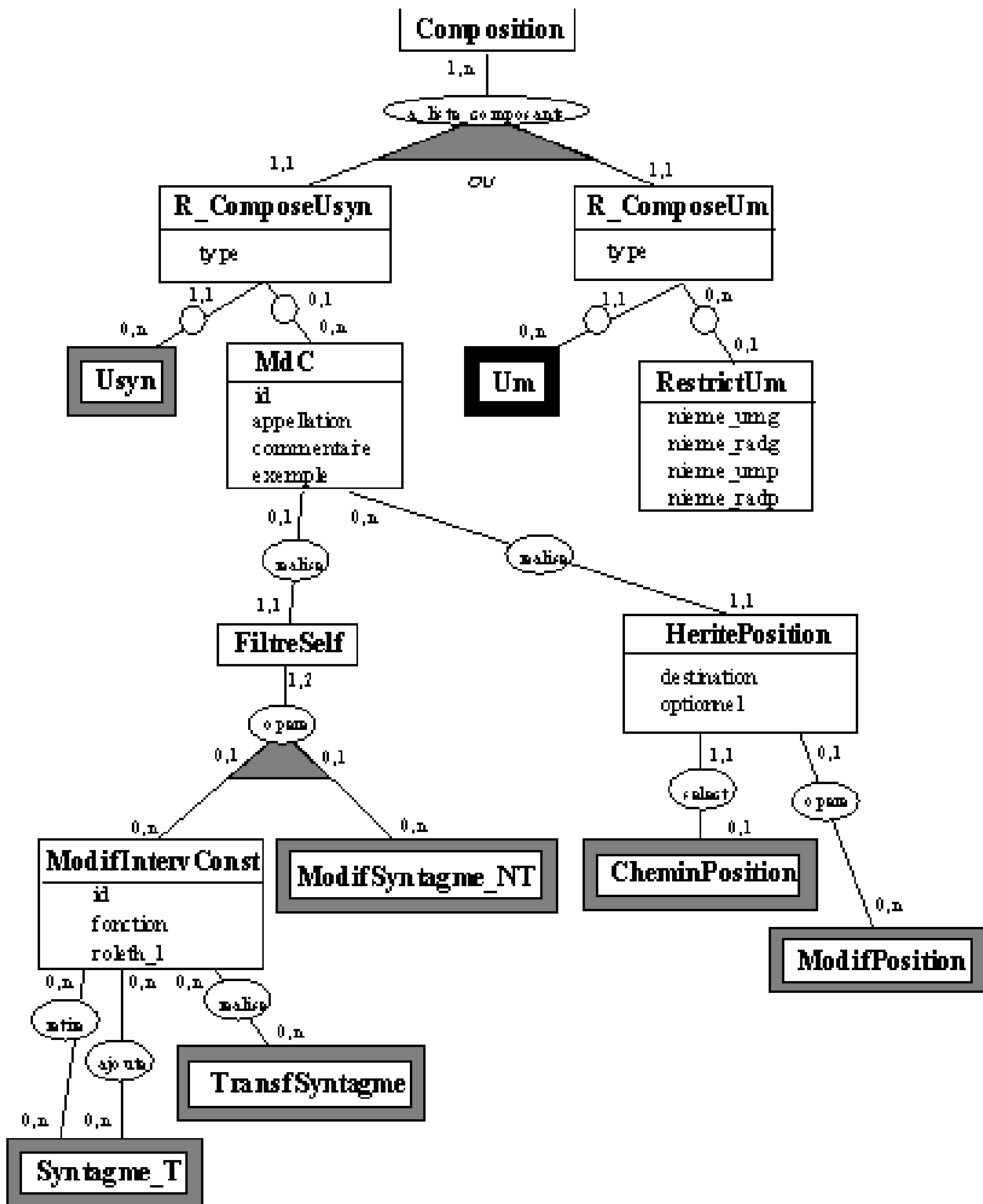
3 : Syntagme_T et Syntagme_NT_C



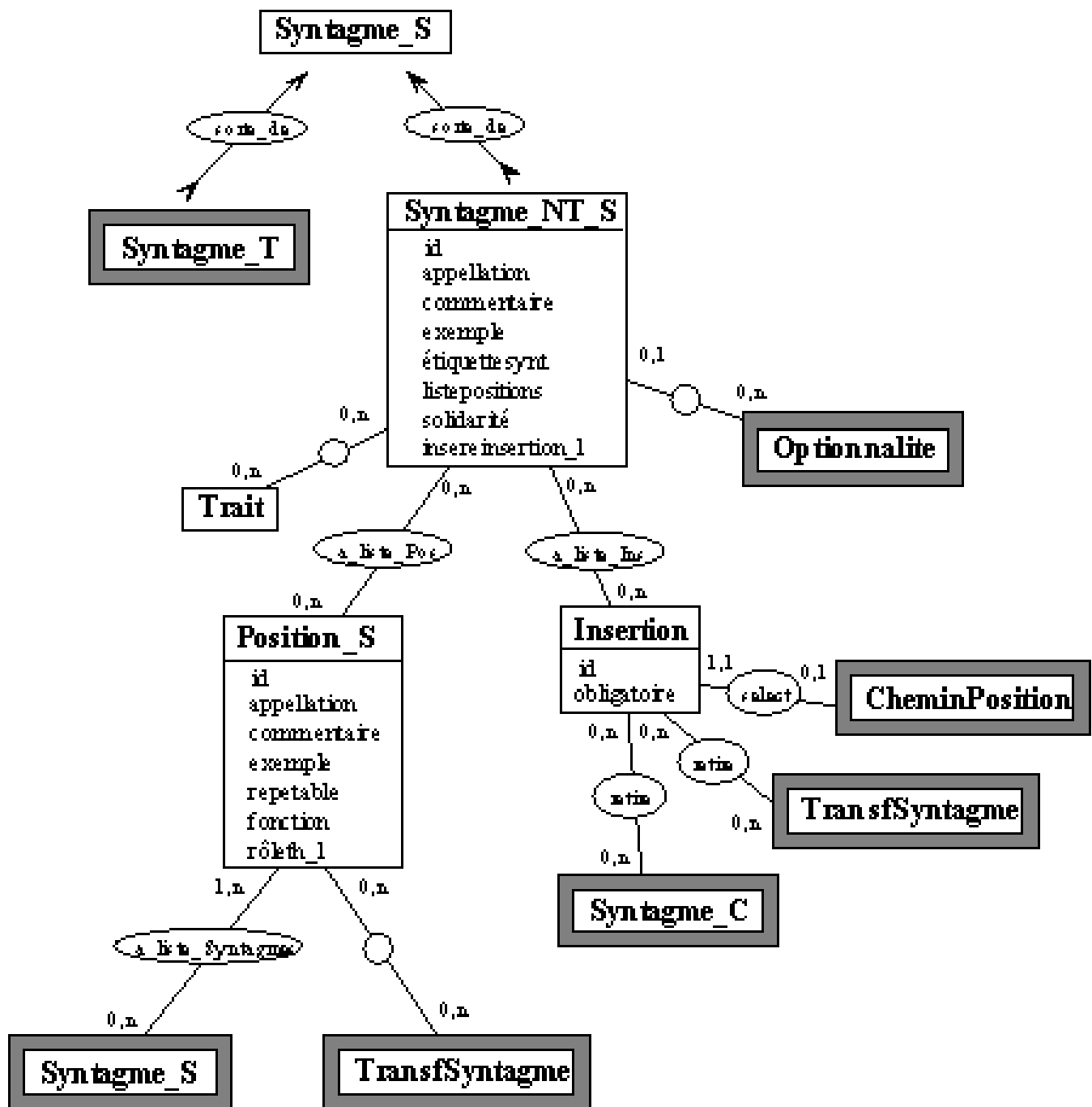
4 : Condition, CheminPosition, CheminSyntagme



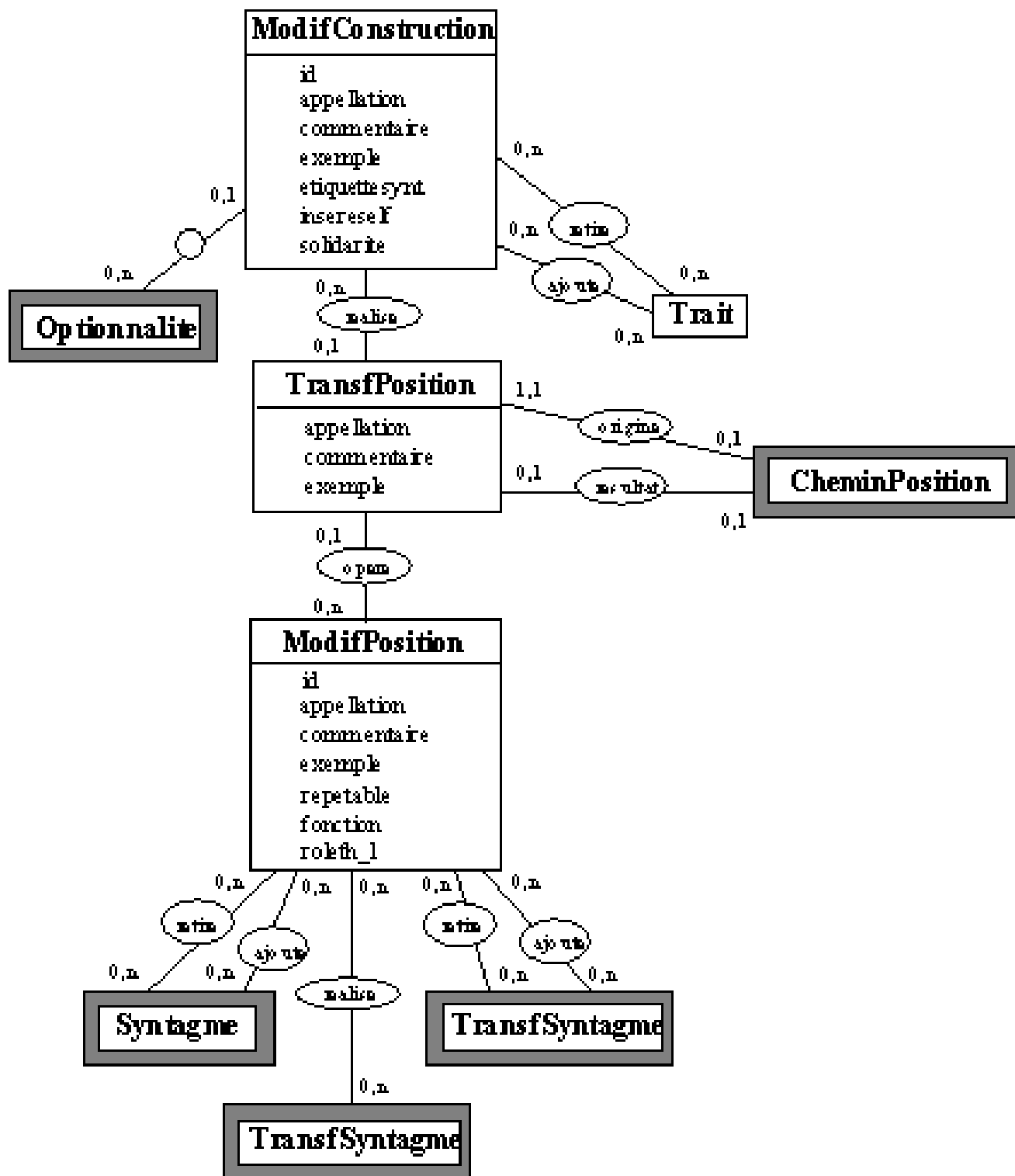
5 : Composition, MdC et ModifIntervConst



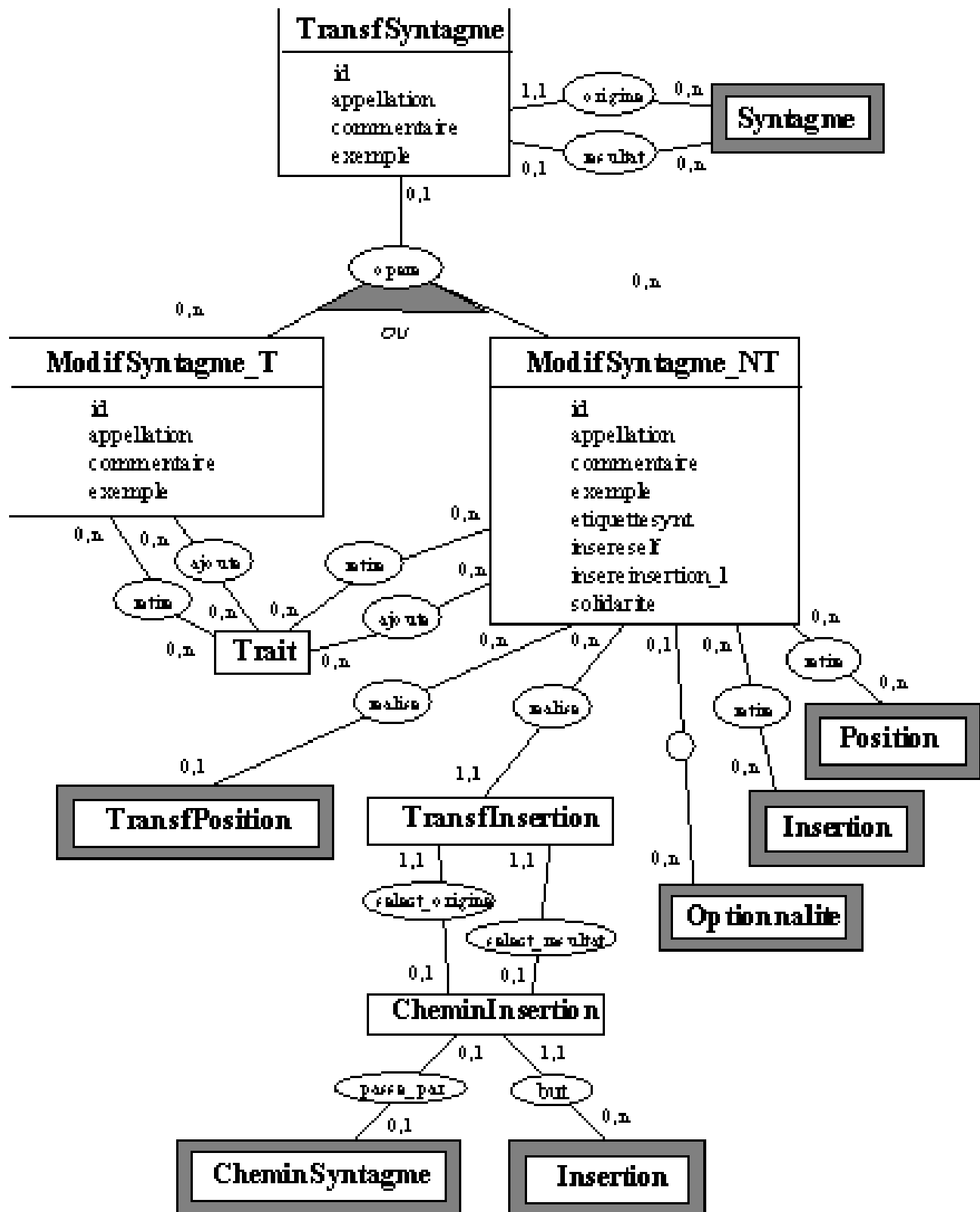
6 : Syntagme_NT_S, Position_S et Insertion



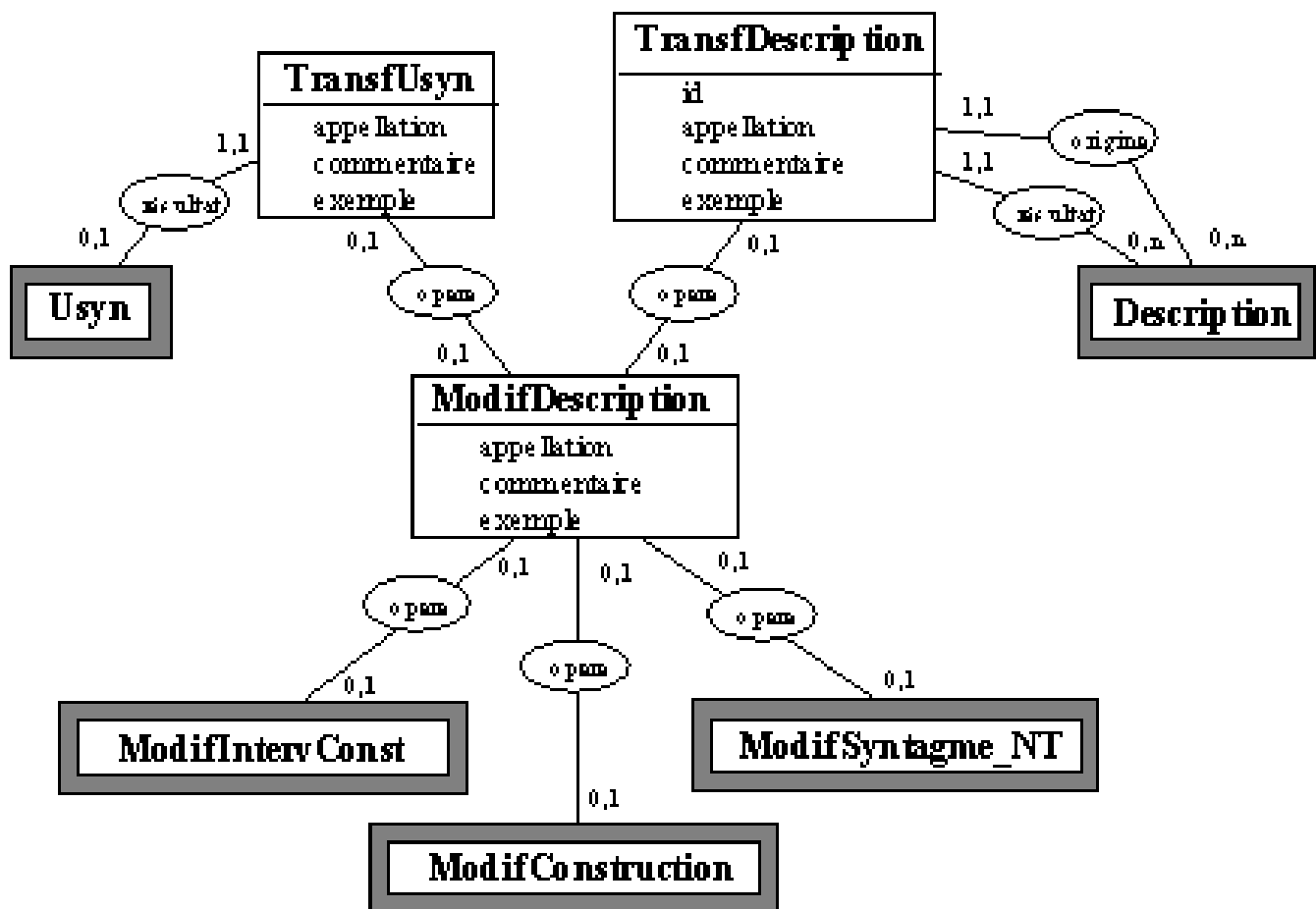
7 : ModifConstruction, TransfPosition et ModifPosition



8 : TransfSyntagme, ModifSyntagme_T et ModifSyntagme_NT



9 : TransfUsyn, TransfDescription et ModifDescription



H - DTD SGML

I : Introduction - Traduction du Modèle Conceptuel

Le modèle conceptuel GENELEX a été largement exprimé au travers de modèles Entité-Attribut-Relation (Merise).

Beaucoup de contraintes d'intégrité sont exprimées dans ce formalisme : typage des objets, typage des relations, cardinalité des relations, etc. Cependant, ce modèle n'étant pas fait pour exprimer des règles - à moins de complications extrêmes-, certaines contraintes ont dû être exprimées dans le document d'accompagnement (restriction sur les combinatoires de valeurs). Il s'ensuit que le modèle Conceptuel de GENELEX combine l'utilisation du formalisme Entité-Attribut-Relation (EAR) et de commentaires en langage naturel.

Une DTD (Définition de Type de Document) SGML est un modèle physique du type grammair qui décrit le marquage des données.

Lors du passage du Modèle Conceptuel à la DTD GENELEX, nous nous sommes efforcés de traduire de manière systématique les modèles EAR et avons tenté d'exprimer formellement la plupart des contraintes d'intégrité décrites en langage naturelle.

Certaines règles de traduction du formalisme EAR vers SGML ont été mises en oeuvre:

- (i) Les Entités EAR deviennent des Eléments SGML.
- (ii) Les Attributs d'Entités EAR deviennent des Attributs d'Eléments.

Lorsque les valeurs d'un Attribut sont exclusives les unes des autres et qu'elles constituent un vocabulaire fermé, ces valeurs sont représentées sous forme d'Attributs SGML listés.

(iii) Les Relations non attribuées pointant sur une Entité EAR non partagée sont traduites par des liens de hiérarchie entre les Eléments de la DTD. Leur cardinalité est traduite par les indicateurs d'occurrences SGML: ? + *

(iv) Les Relations non Attribuées pointant sur une Entité EAR partagée sont exprimées par des liens de référence entre les Eléments.

(v) Les Relations Attribuées sont traduites par des Eléments SGML attribués mis en relation - hiérarchie ou référence - avec les Eléments traduisant les Entités EAR.

Un fichier de contraintes ("syntaxe.ctr") a été créé pour faciliter la lecture des références croisées. Ces contraintes apparaissent comme des commentaires et seront donc ignorées par un parser SGML ; elles expriment dans une syntaxe intuitive le typage des références.

II : DTD GENELEX commentée

1 : DTD *genelex.dtd*

```
<!--Consortium GENELEX @(#) genelex.dtd 3.0 -->

<!-- *****A L'ADRESSE DES UTILISATEURS *****

Vos remarques concernant la DTD seront etudiees par le consortium
GENELEX. Celui-ci assurera la diffusion de la nouvelle version qui
pourrait en decouler.

***** -->

<!DOCTYPE Genelex [

<!ENTITY % ISolat1 PUBLIC "ISO 8879-1986//ENTITIES Added Latin 1//EN">

%ISolat1

<!ENTITY % CustEnt PUBLIC "-//GLX-TEAM//ENTITIES Custom Entity Set//FR">
```

```
<!ENTITY % MorpEnt PUBLIC "-//GLX-TEAM//ENTITIES Morphologie Entity Set//FR">
```

```
<!ENTITY % SyntEnt PUBLIC "-//GLX-TEAM//ENTITIES Syntaxe Entity Set//FR">
```

```
%CustEnt
```

```
%MorpEnt
```

```
%SyntEnt
```

```
<!--
```

Un document Genelex est compose de plusieurs parties:

- la description morphologique
- la description syntaxique
- ...

Pour selectionner la partie souhaitee, il suffit de specifier le mot

cle approprie (INCLUDE or IGNORE) dans les declarations d'entite

suivantes :

```
-->
```

```
<!ENTITY % isMor "INCLUDE" >
```

```
<!ENTITY % isSyn "INCLUDE" >
```

```
<!ELEMENT Genelex - O ( GenelexMorpho? & GenelexSyntaxe? & CombVE*)>
```

```
<!ATTLIST Genelex
```

```
nom CDATA #REQUIRED
```

```
langue CDATA #REQUIRED
```

```
version CDATA #IMPLIED
```

```
date_creation1 CDATA #IMPLIED
```

```
date_creationglx CDATA #IMPLIED
```

```
date_modif CDATA #IMPLIED
```

```
propriete CDATA #IMPLIED
```

```

copyright CDATA #IMPLIED

integrite (SANS_B|%pBooleen) SANS_B

certification CDATA #IMPLIED>

<!-- ***** -->

<!ENTITY % pGlose

"appellation CDATA #IMPLIED

exemple CDATA #IMPLIED

commentaire CDATA #IMPLIED">

<!-- D'une maniere generale dans tout le fichier :

- appellation : permet de nommer de facon comprehensible

et si possible univoque l'objet

- exemple : permet d'illustrer l'emploi (citation d'auteur,

exemple de dictionnaire ou de linguiste)

- commentaire : champ libre pour l'utilisateur -->

<!-- ***** -->

<!ELEMENT CombVE - O EMPTY>

<!ATTLIST CombVE

id ID #REQUIRED

datation (SANS_D|%pDatation) SANS_D

niveaulgue (SANS_NL|%pNiveauLgue) SANS_NL

frequence (SANS_F|%pFrequence) SANS_F

vargeog CDATA #IMPLIED>

<![ %isMor [

<!ENTITY % GLXmor PUBLIC "-//GLX-TEAM//DTD Description Morphologie//FR">

<!ENTITY % MorpCtr PUBLIC "-//GLX-TEAM//DTD Contraintes Morphologie//FR">

%GLXmor

```

```

%MorpCtr
]]>
<![ %isSyn [
<!ENTITY % GLXsyn PUBLIC "-//GLX-TEAM//DTD Description Syntaxe//FR">
<!ENTITY % SyntCtr PUBLIC "-//GLX-TEAM//DTD Contraintes Syntaxe//FR">
%GLXsyn
%SyntCtr
]]>
]>

```

2 : DTD *syntaxe.dtd*

```

<!--Consortium GENELEX @(#) syntaxe.dtd 4.0 -->

```

```

<!--

```

```

***** A L'ADRESSE DES UTILISATEURS *****

```

Vos remarques concernant la DTD seront etudiees par le consortium

GENELEX. Celui-ci assurera la diffusion de la nouvelle version qui

pourrait en decouler.

```

*****

```

```

-->

```

```

<!ELEMENT GenelexSyntaxe - O (

```

```

Usyn+ &

```

```

Description+ &

```

```

Self+ &

```

```

IntervConst* &

```

ComportAppele* &

Optionnalite* &

Construction* &

Position_C* &

Position_S* &

Insertion* &

Syntagme_T* &

Syntagme_NT_C* &

Syntagme_NT_S* &

MdC* &

TransfDescription* &

ModifConstruction* &

ModifPosition* &

TransfSyntagme* &

ModifSyntagme_T* &

ModifSyntagme_NT* &

ModifIntervConst* &

Trait_Lex* &

Trait_Introd* &

Trait_Prep* &

Trait_Conj* &

Trait_ProRel* &

Trait_ProIntrog* &

Trait_RefLex* &

Trait_RefIntrod* &

Trait_RefPrep* &

Trait_RefConj* &

Trait_RefProRel* &

Trait_RefProIntrog* &

Trait_Mode* &

Trait_Temps* &

Trait_Personne* &

Trait_Genre* &

Trait_Nombre* &

Trait_NombrePosseur* &

Trait_SsCatMorph* &

Trait_SsCatSynt* &

Trait_Aux* &

Trait_Pronominal* &

Trait_Neg* &

Trait_Accord* &

Trait_Passif* &

Trait_Tournure* &

Trait_Coref* &

Trait_Aspect* &

Trait_Bin* &

Trait_Libre* &

RoleTh* &

Fonction*)>

<!-- ***** -->

<!-- ***** UNITE SYNTAXIQUE ET DESCRIPTION ***** -->

```

<!-- ***** -->

<!ELEMENT Usyn - O (Composition* & TransfUsyn*)>

<!ATTLIST Usyn

id ID #REQUIRED

%pGlose

attestation CDATA #IMPLIED

combve IDREF #IMPLIED

description IDREF #REQUIRED

description_l IDREFS #IMPLIED

transfdescription_l IDREFS #IMPLIED>

<!-- Le champ attestation permet de preciser la source de l'emploi releve
(nom ou titre du dictionnaire, du texte d'auteur, ou de l'article
de linguistique) -->

<!-- L'attribut description note la description de base,
la liste description_l enregistre les descriptions transformees,
la liste transfdescription_l note les transformations entre les
descriptions associees a l'Usyn ; ces transformations peuvent operer
entre la description de base et les transformees mais aussi sur les
transformees entre elles. -->

<!ELEMENT Description - O (Condition*) >

<!ATTLIST Description

id ID #REQUIRED

%pGlose

um_representante CDATA #IMPLIED

self IDREF #REQUIRED

```



```
construction IDREF #IMPLIED>
```

```
<!-- ***** -->
```

```
<!-- ***** SELF ***** -->
```

```
<!-- ***** -->
```

```
<!ELEMENT Self - O EMPTY>
```

```
<!ATTLIST Self
```

```
id ID #REQUIRED
```

```
syntagme_nt_s IDREF #IMPLIED
```

```
syntagme_nt_s_l IDREFS #IMPLIED
```

```
transfsyntagme_l IDREFS #IMPLIED
```

```
intervconst IDREF #IMPLIED
```

```
comportappele_l IDREFS #IMPLIED>
```

```
<!-- Le champ syntagme_nt_s n'est instancie que pour les Usyn composees,
```

```
il en exprime alors la structure interne, eventuellement reduite
```

```
a l'etiquette syntagmatique, par un Syntagme_NT_S avec ou sans
```

```
reecriture.
```

```
Ex : mettre en marche SV
```

```
Les champs syntagme_nt_s_l et transfsyntagme_l ne concernent de
```

```
meme que les Usyn composees et servent a noter d'eventuelles
```

```
transformations sur la structure interne.
```

```
Le champ intervconst donne les realisations de Self intervenant
```

```
dans la construction externe :
```

```
- en tant qu'occupant de la construction si celle-ci decrit
```

```
un contexte d'apparition dans lequel s'inscrit le Self,
```

```
- en tant que predicat associe a une construction decrivant un schema
```

de complementation.

La liste comportappelle_l permet d'indiquer les alternatives

de comportement de Self en tant qu'appelle par un element

non decrit dans sa construction -->

```
<!ELEMENT IntervConst - O EMPTY>
```

```
<!ATTLIST IntervConst
```

```
id ID #REQUIRED
```

```
fonction IDREF #IMPLIED
```

```
roleth_l IDREFS #IMPLIED
```

```
syntagme_t_l IDREFS #REQUIRED>
```

```
<!-- L'element IntervConst comporte des syntagmes terminaux.
```

On peut y exprimer des variations de realisation de Self :

Ex : N + [Nombre:PLURIEL]

Ex : V

V + [Pronominal:SE]

V + [Pronominal:SE_EN]

Pour les mots simples, on autorise un ecart entre la categorie

fonctionnelle et la categorie grammaticale de l'unite morphologique.

Ex : description du comportement adjectival de

l'Unite Morphologique du NOM abricot

Pour les composes syntaxiques, on rend compte de la categorie

fonctionnelle (externe) du compose qui peut differer de son etiquette

de syntagme structurel interne

Ex : mettre en oeuvre (VERBE / SV)

On peut de plus indiquer sur cet element une fonction et des roles

thematiques : ce sont les valeurs portees par le Self lorsqu'il

```

s'insere dans la construction. -->

<!ELEMENT ComportAppele - O EMPTY>

<!ATTLIST ComportAppele

id ID #REQUIRED

fonction IDREF #IMPLIED

roleth_1 IDREFS #IMPLIED

syntagme_t IDREF #REQUIRED>

<!-- ComportAppele note un comportement de Self en tant qu'appelle par
un element non decrit dans sa construction.

Un comportement regroupe :

- une etiquette syntagmatique terminale et une liste de traits,
exprimant la categorie fonctionnelle d'appelle et les traits
restrictifs associes :

cette association est equivalente a un syntagme terminal

Ex : PREP + [SsCatSynt:LIEU]

(si le Self porte un IntervConst, l'etiquette syntagmatique
du syntagme du ComportAppele doit etre la meme que l'etiquette
de l'un des syntagmes dudit IntervConst)

- une fonction en tant qu'appelle

- une liste de roles thematiques -->

<!-- ***** -->

<!-- ***** CONSTRUCTION ***** -->

<!-- ***** -->

<!ENTITY % pConstSyntNT

"%pGlose

```

```

etiquettesynt (SANS_E
| %pEtiquetteSynt_NT
| %pEtiquetteSynt_cust) SANS_E

solidarite CDATA #IMPLIED

optionnalite IDREF #IMPLIED">

<!-- L'attribut solidarite indique par des tirets les paires de positions
solidaires ;

Ex : P0 SELF-P1-P2 -->

<!ELEMENT Optionnalite - O (ConditionOpt*)>
<!ATTLIST Optionnalite
id ID #REQUIRED
%pGlose
libelle CDATA #REQUIRED>

<!ELEMENT ConditionOpt - O (SiOpt+ , AlorsOpt+)>
<!ELEMENT (SiOpt|AlorsOpt) - O EMPTY>
<!ATTLIST (SiOpt|AlorsOpt)
negation (%pBooleen) NON
nieme_position NUMBER #REQUIRED>

<!-- L'optionnalite est exprimee :
- d'une part par un libelle indiquant pour chaque position par
des parentheses si son effacement est possible dans une
realisation de la construction ; toutes les positions de la
construction ou du syntagme sont rappelees dans ce champ,
le point d'insertion du Self n'y apparait pas

Ex : P0 (P1) P2 (P3)

- d'autre part par des conditions exprimant d'eventuelles

```

interdependances entre les positions -->

<!ELEMENT Construction - O EMPTY>

<!ATTLIST Construction

id ID #REQUIRED

%pConstSyntNT

squelettique (%pBooleen) NON

listepositions (%pTypeListPos) FERMEE

inereself NUMBER #IMPLIED

trait_l IDREFS #IMPLIED

position_c_l IDREFS #REQUIRED>

<!-- L'attribut squelettique indique si l'element est un squelette

de construction - donc a mettre en regard avec un

ModifConstruction pour obtenir une Construction a part entiere.

L'attribut listepositions note si la liste position_c_l est

a comprendre comme une liste FERMEE - toutes les positions

sont donnees - ou OUVERTE.

L'attribut inereself indique lorsqu'il est renseigne le point

d'insertion de Self dans la liste de positions : avant la

position de rang correspondant a sa valeur.

La liste position_c_l renvoie a des Position_C.

Il s'agit d'une liste ordonnee, l'ordre des elements dans

la liste correspond a l'ordre canonique (valeur initiale = 0),

et les elements pourront par la suite etre references par

leur rang dans cette liste -->

<!-- ***** -->

```

<!-- ***** POSITION ET INSERTION ***** -->

<!-- ***** -->

<!ENTITY % pPosition

"%pGlose

repetable (SANS_B|%pBooleen) SANS_B

fonction IDREF #IMPLIED

roleth_1 IDREFS #IMPLIED">

<!-- L'attribut repetable indique si une position peut etre

realisable plusieurs fois -->

<!ELEMENT (Position_C|Position_S) - O EMPTY>

<!ATTLIST Position_C

id ID #REQUIRED

%pPosition

syntagme_c_1 IDREFS #IMPLIED

transfsyntagme_1 IDREFS #IMPLIED>

<!ATTLIST Position_S

id ID #REQUIRED

%pPosition

syntagme_s_1 IDREFS #REQUIRED

transfsyntagme_1 IDREFS #IMPLIED>

<!-- L'attribut syntagme_c_1 reference les occupants possibles

de Position_C :

- syntagme terminal (Syntagme_T)

- syntagme non terminal dont la reecriture est ou non

decrete (Syntagme_NT_C) -->

```

```

<!-- L'attribut syntagme_s_l reference les occupants possibles
de Position_S :
- syntagme terminal (Syntagme_T)
- syntagme non terminal dont la reecriture est ou non
decrete (Syntagme_NT_S) -->
<!ELEMENT Insertion - O (CheminPosition)>
<!ATTLIST Insertion
id ID #REQUIRED
obligatoire (SANS_B|%pBooleen) SANS_B
retire_syntagme_c_l IDREFS #IMPLIED
retire_transfsyntagme_l IDREFS #IMPLIED>
<!-- L'insertion dans un syntagme structurel est utilisee uniquement
pour représenter le cas d'une insertion renvoyant a une position
d'une position decrite dans la construction syntaxique externe :
l'element CheminPosition donne acces a cette position.
Les attributs de l'Insertion indiquent les eventuelles modifications
a appliquer a la position referencee : retrait de syntagmes et
transformations entre syntagmes. La repetabilite reste celle de
la position externe referencee.
Ex : le compose "mettre en oeuvre" a dans sa construction externe
une position objet direct contenant par exemple un syntagme
nominal et un pronom personnel :
mettre en oeuvre un processus, le mettre en oeuvre
l'insertion n'est ici possible que pour le SN :
mettre un processus en oeuvre
L'attribut obligatoire indique si, lors d'une realisation de la

```

position externe referencee, le phenomene d'insertion est

obligatoire ou facultatif -->

```
<!-- ***** -->
```

```
<!-- ***** CONDITION ***** -->
```

```
<!-- ***** -->
```

```
<!ELEMENT Condition - O (Si+ , Alors+)>
```

```
<!ATTLIST Condition
```

```
appellation CDATA #IMPLIED>
```

```
<!-- Les Conditions permettent d'exprimer des contraintes mutuelles entre
```

```
realisations de positions.
```

```
Ex : si P0 == P[SsCatSynt:COMPLETIVE]
```

```
alors P1 != P[SsCatSynt:COMPLETIVE]
```

Les listes de Predicats (Si et Alors) permettent d'exprimer

des conjonctions sur ces predicats.

Les disjonctions sont exprimees au moyen de la liste de Conditions

portee par la Description (liste "et"). -->

```
<!ELEMENT (Si|Alors) - O (CheminPosition|CheminSyntagme
```

```
|SelectIntervConst)>
```

```
<!ATTLIST (Si|Alors)
```

```
portee (%pPortee) EXTERNE
```

```
negation (%pBooleen) NON>
```

```
<!-- Un predicat selectionne :
```

```
- un syntagme ou une position de la construction externe
```

```
(champ portee EXTERNE),
```

```
- un syntagme ou une position du syntagme structurel du Self
```


(champ portee INTERNE),

- une realisation du Self en tant qu intervenant de construction (champ portee INTERVENANT).

La selection se fait

- pour un syntagme grace au CheminSyntagme,
 - pour une position grace au CheminPosition,
 - pour une realisation en tant qu intervenant grace au SelectIntervConst.

Un predicat porte eventuellement une negation qui exprime l inhibition du syntagme ou de la position pointee. -->

```
<!ELEMENT CheminSyntagme - O (CheminSyntagme?)>
```

```
<!ATTLIST CheminSyntagme
```

```
nieme_position NUMBER 0
```

```
syntagme IDREF #REQUIRED>
```

```
<!-- Cet element permet de selectionner un syntagme particulier.
```

La recursivite est utilisee pour descendre dans une eventuelle reecriture. L'element aboutit toujours a un syntagme.

Les positions sont referencees par nieme_position qui indique leur rang dans la liste ou elles apparaissent ;

la valeur 0 reference le premier element de la liste. -->

```
<!ELEMENT CheminPosition - O (CheminSyntagme?,PositionBut)>
```

```
<!-- Pour une construction ou un syntagme donne, cet element permet de selectionner une de ses positions - eventuellement apparaissant dans une reecriture de syntagme -.
```

L'element PositionBut indique la position selectionnee ;

si celle-ci apparait dans une reecriture de syntagme,

```

on utilise l'element CheminSyntagme pour atteindre

ledit syntagme -->

<!ELEMENT PositionBut - O EMPTY>

<!ATTLIST PositionBut

nieme_position NUMBER 0>

<!ELEMENT SelectIntervConst - O EMPTY>

<!ATTLIST SelectIntervConst

syntagme_t IDREF #REQUIRED>

<!-- ***** -->

<!-- ***** SYNTAGMES TERMINAUX ET NON TERMINAUX ***** -->

<!-- ***** -->

<!ELEMENT Syntagme_T - O EMPTY>

<!ATTLIST Syntagme_T

id ID #REQUIRED

%pGlose

etiquettesynt (SANS_E

| %pEtiquetteSynt_T

| %pEtiquetteSynt_cust) SANS_E

trait_l IDREFS #IMPLIED>

<!ELEMENT Syntagme_NT_C - O EMPTY>

<!ATTLIST Syntagme_NT_C

id ID #REQUIRED

%pConstSyntNT

listepositions (%pTypeListPos) FERMEE

insereself NUMBER #IMPLIED

```

```

trait_l IDREFS #IMPLIED

position_c_l IDREFS #IMPLIED>

<!-- La liste position_c_l renvoie a des Position_C.

Il s'agit d'une liste ordonnee, l'ordre des elements dans

la liste correspond a l'ordre canonique et les elements pourront

par la suite etre references par leur rang dans cette liste. -->

<!ELEMENT Syntagme_NT_S - O EMPTY>

<!ATTLIST Syntagme_NT_S

id ID #REQUIRED

%pConstSyntNT

listepositions (%pTypeListPos) FERMEE

insereinsertion_l NUMBERS #IMPLIED

trait_l IDREFS #IMPLIED

position_s_l IDREFS #IMPLIED

insertion_l IDREFS #IMPLIED>

<!-- La liste position_s_l renvoie a des Position_S ; la liste insertion_l

renvoie a des Insertions . Il s'agit de listes ordonnees, leurs

elements pourront par la suite etre references par leur rang

d'apparition dans leur liste.

La liste d'entiers insereinsertion_l indique les points ou s'insereent

parmi les Positions les Insertions presentes dans la liste

insertion_l -->

<!-- Le Syntagme_T est un occupant de position terminal (etiquette reduite

a la Categorie grammaticale ou a "e" pour la trace)

Le Syntagme_NT_C/S est un occupant de position non terminal,

```

on utilisera sa liste de positions (#IMPLIED) si l'on veut préciser sa reécriture.

Le Syntagme_NT_S sert de plus pour décrire la structure interne d'une unité composée (champ syntagme_nt_s du Self).

La liste trait_l renvoie à des traits restrictifs et permet ainsi de spécifier sur un syntagme un ensemble de contraintes :

contraintes lexicales, morphologiques, morpho-syntaxiques, syntaxico-sémantiques, voire sémantiques.

Le champ appellation permettra de noter le nom usuel du syntagme.

Ex : Syntagme_NT_C

```
etiquettesynt = "P"
```

```
trait_l -> [Mode:INFINITIF]
```

```
appellation = "phrase infinitive"
```

La catégorie "e" permet de noter les traces pour les tenants de la grammaire générative et de les considérer comme des syntagmes fantômes auxquels on peut associer les restrictions nécessaires.

Ex : Syntagme_T

```
etiquettesynt = "e"
```

```
trait_l -> [Personne:3][Nombre:SINGULIER]
```

```
appellation = "elt vide" -->
```

```
<!-- ***** -->
```

```
<!-- ***** COMPOSITION ***** -->
```

```
<!-- ***** -->
```

```
<!ELEMENT Composition - O (R_ComposeUm|R_ComposeUsyn)+>
```

```
<!-- Les éléments Composition portés par l'unité syntaxique
```

```
enregistrent les listes alternatives de lexicalisations :
```

Ex : (avoir admiration pour)

(eprouver admiration pour)

(eprouver admiration envers)

(porter admiration a) -->

<!-- La liste contenue de R_ComposeUm et R_ComposeUsyn donne la
liste des composants pour une alternative de composition donnee. -->

<!-- On referera dans la suite les composants par des traits RefLex
portant deux valeurs d indices :

- l indice de la compositon dans la liste de compositions

- l indice du composant dans la liste des composants

Ex. : [RefLex:1,2]:

Ce mecanisme sera utilise :

- soit dans le syntagme structurel interne de Self,

- soit dans le mode de calcul MdC. -->

<!ENTITY % pCompose

"type (%pTypeComposant) APPELE">

<!ELEMENT R_ComposeUm - O (RestrictUm*)>

<!ATTLIST R_ComposeUm

%pCompose

um IDREF #REQUIRED>

<!ELEMENT R_ComposeUsyn - O EMPTY>

<!ATTLIST R_ComposeUsyn

%pCompose

usyn IDREF #REQUIRED

mdc IDREF #IMPLIED>

<!-- Dans le cas ou l'on veut noter la formation du compose selon

l'Usyn composante :

- lexicalisation de ses positions par d'autres composants

(seulement dans le cas d'un comportement d'appelant)

et/ou

- heritage de ses positions avec eventuellement ajout de

restrictions,

on associe un MdC decrivant l'ensemble de ces phenomenes.

NB : La lexicalisation d'une position d'une Usyn appelante par un

autre composant est faite par l'introduction d'un Syntagme

portant un trait [RefLex:nieme_cposition,nieme_cposant] dans

le MdC de l'Usyn. Ce Syntagme peut de plus porter d'autres

traits restrictifs ; on devra dans ce cas s'assurer que ces

traits sont compatibles avec le MdC et le Self de l'Usyn

fonctionnant en tant que composant appele.

Ex : Le MdC de l'appelante indique :

```
SN[RefLex:3,1][SsCatSynt:DET_VIDE]
```

L'appellee (3,1) ne devra pas dans son MdC imposer

une lexicalisation de sa position Determinant. -->

```
<!ELEMENT MdC - O (HeritePosition* & FiltreSelf?)>
```

```
<!ATTLIST MdC
```

```
id ID #REQUIRED
```

```
%pGlose>
```

<!-- Le MdC decrit le mode de composition d'une Usyn Composee sur une

de ses Usyn composantes. Il permet de specifier :

- les contraintes emises par l'Usyn composee sur les Usyn

composantes,

- l'organisation mutuelle des Usyn composantes, en indiquant quel composant occupe quelle position d'un autre composant.

Le MdC s applique a l usyn composante et permet de

- heriter des positions de sa construction
- filtrer son Self

Une position non referencee par le MdC est inhibee. Lorsqu aucun filtre n est precise sur le Self, celui-ci est par default herite. -->

```
<!ELEMENT HeritePosition - O (CheminPosition)>
```

```
<!ATTLIST HeritePosition
```

```
destination (%pDestination) EXTERIEUR
```

```
optionnel (HERITAGE| %pBooleen) HERITAGE
```

```
modifposition IDREF #IMPLIED>
```

```
<!-- L'attribut destination indique si la position heritee se retrouve dans l'exterieur ou l'interieur du compose.
```

```
L'attribut optionnel permet de noter d'eventuelles modifications de l'optionnalite de la position heritee. -->
```

```
<!ELEMENT FiltreSelf - O EMPTY>
```

```
<!ATTLIST FiltreSelf
```

```
modifintervconst IDREF #IMPLIED
```

```
modifsyntagme_nt IDREF #IMPLIED>
```

```
<!-- Le FiltreSelf doit au moins realiser une des 2 operations :
```

- Modification de l'IntervConst
- Modification du syntagme structurel interne.-->

```
<!-- ***** -->
```

```
<!-- ***** TRANSFORMATIONS ***** -->
```

```
<!-- ***** -->
```

```
<!-- Il existe trois types de transformations :
```

- TransfUsyn : transformations operant entre deux Unites Syntaxiques
- issues de la meme Um

```
Ex : neutralite
```

- issues d'Um differentes

```
Ex : derivation syntaxique
```

- TransfDescription : transformations operant entre deux Descriptions
(c'est-a-dire deux couples Self/Construction).

```
Ex : passivation
```

- TransfSyntagme : transformations operant entre des occupants
de Position, c'est-a-dire des Syntagmes, terminaux ou non.

```
Ex : pronominalisation -->
```

```
<!ELEMENT TransfUsyn - O (ModifDescription?)>
```

```
<!ATTLIST TransfUsyn
```

```
%pGlose
```

```
usyn_resultat IDREF #REQUIRED>
```

```
<!-- Les TransfUsyn sont pointees par les Usyn origine et indiquent  
les Usyn resultat
```

```
Elles s'appliquent entre la Description de base de l'Usyn origine  
et la Description de base de l'Usyn resultat-->
```

```
<!ELEMENT TransfDescription - O (ModifDescription?)>
```

```
<!ATTLIST TransfDescription
```

```
id ID #REQUIRED
```


%pGlose

description_origine IDREF #REQUIRED

description_resultat IDREF #REQUIRED>

<!-- Les TransfDescriptions sont portees par l'Usyn et operent entre
les Descriptions de cette Usyn.

L'origine d'une TransfDescription peut etre la Description de base
mais aussi une Description transformee de l'Usyn. -->

<!ELEMENT ModifDescription - O EMPTY>

<!ATTLIST ModifDescription

%pGlose

modifconstruction IDREF #IMPLIED

modifintervconst IDREF #IMPLIED

modifsyntagme_nt IDREF #IMPLIED>

<!-- Les ModifDescriptions peuvent rendre compte de trois phenomenes :

- transformation sur la construction
- transformation sur les realisations de Self associees
a la construction externe en tant qu'occupant ou predicat
(champ intervconst de Self)

Ex : $V \rightarrow V[\text{Passif:PLUS}]$

- transformation sur le syntagme structurel interne decrivant
un compose

Ex : pour les beaux yeux de SN

-> pour ses beaux yeux

Dans le premier cas, on peut toujours choisir entre un mode
calculatoire ou descriptif :

- en mode calculatoire, la Construction pointee par la

Description resultat est declaree squelettique : la construction resultat elle meme est a construire a partir de ce squelette en lui appliquant les modifications indiquees sur ModifConstruction

- en mode descriptif, la Construction pointee par la

Description resultat est totalement decrite : le

ModifConstruction indique alors des correspondances, a des degres de finesse variable, entre Construction origine et Construction resultat.

Dans les deux autres cas, les resultats - IntervConst et Syntagme_NT_S - sont totalement decrits : on est en mode descriptif -->

```
<!ELEMENT ModifConstruction - O (TransfPosition*)>
```

```
<!ATTLIST ModifConstruction
```

```
id ID #REQUIRED
```

```
%pGlose
```

```
etiquettesynt (HERITAGE
```

```
|%pEtiquetteSynt_NT
```

```
|%pEtiquetteSynt_cust) HERITAGE
```

```
inreself NUMBER #IMPLIED
```

```
solidarite CDATA #IMPLIED
```

```
optionnalite IDREF #IMPLIED
```

```
retire_trait_1 IDREFS #IMPLIED
```

```
ajoute_trait_1 IDREFS #IMPLIED>
```

```
<!-- En mode descriptif, cet element permet de noter - eventuellement
```

de facon partielle - des informations sur le passage de la construction origine a la construction resultat ; il peut minimalement s'agir de mises en correspondance de positions par le contenu TransfPosition*.

En mode calculatoire, cet element permet de construire la construction resultat a partir de la construction origine et du squelette : le squelette est habille avec des elements ou attributs provenant de la construction origine ou precises sur le ModifConstruction.

Si les attributs etiquettesynt, insereself, solidarite, optionnalite et ceux groupes par pGlose

- sont renseignes : ils remplissent (voire ecrasent)

les attributs correspondants du squelette

- ne sont pas renseignes : les attributs correspondants

et non renseignes sur la construction squelette sont herites de la construction origine

Les traits de la Construction origine sont herites moyennant des retraits et ajouts notes ici.

La liste contenue de TransfPositions exprime la formation des positions de la construction resultat a partir des positions du squelette et des positions de la construction origine. -->

```
<!ELEMENT TransfPosition - O (CheminPosition , CheminPosition?)>
```

```
<!ATTLIST TransfPosition
```

```
%pGlose
```

```
modifposition IDREF #IMPLIED>
```

```
<!-- Selection de la Position d'origine
```

Le premier element contenu CheminPosition pointe vers une position de la construction a l'origine de la transformation.

Selection de la Position resultat

La position resultat est eventuellement selectionnee par le second element contenu CheminPosition

L'attribut ModifPosition indique les modifications a operer pour passer de la position origine a la position resultat -->

```
<!ELEMENT ModifPosition - O EMPTY>
```

```
<!ATTLIST ModifPosition
```

```
id ID #REQUIRED
```

```
%pGlose
```

```
repetable (HERITAGE| %pBooleen) HERITAGE
```

```
fonction IDREF #IMPLIED
```

```
roleth_1 IDREFS #IMPLIED
```

```
retire_syntagme_1 IDREFS #IMPLIED
```

```
retire_transfsyntagme_1 IDREFS #IMPLIED
```

```
ajoute_syntagme_1 IDREFS #IMPLIED
```

```
ajoute_transfsyntagme_1 IDREFS #IMPLIED
```

```
transfsyntagme_1 IDREFS #IMPLIED>
```

```
<!-- Le fonctionnement de ModifPosition est similaire a celui du
ModifConstruction ; on aura en particulier la meme alternative
entre un fonctionnement calculatoire - la position resultat est
a construire a partir de la position pointee sur le squelette
et les informations notees ici - et un fonctionnement descriptif
- la position resultat est deja totalement renseignee. -->
```

```

<!ELEMENT TransfSyntagme - O EMPTY>

<!ATTLIST TransfSyntagme

id ID #REQUIRED

%pGlose

syntagme_origine IDREF #REQUIRED

syntagme_resultat IDREF #IMPLIED

modifsyntagme IDREF #IMPLIED>

<!-- Les TransfSyntagme notent les relations de transformation
existant entre :

- des syntagmes occupant une meme position - Syntagmes
terminaux ou non -, les syntagmes a l'origine et au
resultat de la transformation sont alors tous deux
nécessairement renseignés,

- des syntagmes mis en correspondance lors d'une
transformation de plus haut niveau ; dans ce cas,
le syntagme resultat ne sera pas nécessairement
renseigné si on est a l'interieur d'une transformation
de Construction fonctionnant en mode calculatoire. -->

<!ENTITY % pModifSyntagme

"%pGlose

retire_trait_l IDREFS #IMPLIED

ajoute_trait_l IDREFS #IMPLIED">

<!ELEMENT ModifSyntagme_T - O EMPTY>

<!ATTLIST ModifSyntagme_T

id ID #REQUIRED

```

```

%pModifSyntagme>

<!ELEMENT ModifSyntagme_NT - O ((TransfPosition|TransfInsertion)*)>

<!ATTLIST ModifSyntagme_NT

id ID #REQUIRED

%pModifSyntagme

etiquettesynt (HERITAGE

| %pEtiquetteSynt_NT

| %pEtiquetteSynt_cust) HERITAGE

inereself NUMBER #IMPLIED

insereinsertion_l NUMBERS #IMPLIED

retire_position_l NUMBERS #IMPLIED

retire_insertion_l NUMBERS #IMPLIED

solidarite CDATA #IMPLIED

optionnalite IDREF #IMPLIED>

<!-- Modifier un Syntagme consiste a :

- retirer ou ajouter des traits,

- et dans le cas d'un syntagme non terminal :

- changer l'etiquette,

- changer l'optionnalite, la solidarite, et

les eventuels points d'insertion notes.

- modifier les positions et insertions de

l'eventuelle reecriture -->

<!ELEMENT TransfInsertion - O (CheminInsertion, CheminInsertion)>

<!-- L'element TransfInsertion met en rapport une insertion du

syntagme de structure origine et l'insertion correspondante

du syntagme de structure resultat. -->

```

```

<!ELEMENT CheminInsertion - O (CheminSyntagme?,InsertionBut)>

<!ELEMENT InsertionBut - O EMPTY>

<!ATTLIST InsertionBut
nieme_insertion NUMBER 0>

<!-- L attribut nieme_insertion selectionne une insertion par son rang
dans la liste insertion_l ou elle apparait -->

<!ELEMENT ModifIntervConst - O EMPTY>

<!ATTLIST ModifIntervConst
id ID #REQUIRED
fonction IDREF #IMPLIED
roleth_l IDREFS #IMPLIED
retire_syntagme_t_l IDREFS #IMPLIED
ajoute_syntagme_t_l IDREFS #IMPLIED
transfsyntagme_l IDREFS #IMPLIED>

<!-- L'element ModifIntervConst permet de noter les modifications subies,
en transformation ou en composition, par le contenu IntervConst
du Self. -->

<!-- ***** -->

<!-- ***** TRAITS RESTRICTIFS ***** -->

<!-- ***** -->

<!-- Les traits restrictifs permettent de specifier les syntagmes
auxquels ils sont combines

Ex : P[Conj:que][Mode:SUBJONCTIF] =>completive

P[SsCatSynt:RELATIVE] =>relative

P[Mode:INFINITIF] =>infinitive

```

P[SsCatSynt:COORDONNE] =>phrase coordonnee

P[SsCatSynt:SUBORDONNEE] =>subordonnee

SN[Nombre:PLURIEL]

SN[Coref:I] -->

<!--***** Traits LEXICAUX *****-->

<!--*****-->

<!ENTITY % pTrait_Lexical

"id ID #REQUIRED

valeur CDATA #IMPLIED

um IDREF #IMPLIED">

<!-- Ces traits permettent d'exprimer une restriction d'ordre lexical sur

un Syntagme,

- soit en entrant dans le champ "valeur" une chaine de caracteres

representant la graphie - ce moyen ne desambiguise pas

les homographes -,

- soit en referant une unite morphologique (par son identifiant).

On autorise un ecart (fonde sur l'ecart entre categorie

morpho-syntaxique et categorie fonctionnelle) entre la categorie

du syntagme qui supporte le trait lexical et la categorie de l'Um

referee par ce trait lexical

Ex : NOM[Lex:courageux[um:UM04]]

avec en Morphologie :

Um[id:UM04;catgram:ADJECTIF[umg:courageux]] -->

<!ELEMENT Trait_Lex - O EMPTY>

<!ATTLIST Trait_Lex

%pTrait_Lexical


```

saturesynt (%pBooleen) OUI>

<!ELEMENT Trait_Introd - O EMPTY>

<!ATTLIST Trait_Introd

%pTrait_Lexical>

<!ELEMENT Trait_Prep - O EMPTY>

<!ATTLIST Trait_Prep

%pTrait_Lexical>

<!ELEMENT Trait_Conj - O EMPTY>

<!ATTLIST Trait_Conj

%pTrait_Lexical>

<!ELEMENT Trait_ProRel - O EMPTY>

<!ATTLIST Trait_ProRel

%pTrait_Lexical>

<!ELEMENT Trait_ProIntrog - O EMPTY>

<!ATTLIST Trait_ProIntrog

%pTrait_Lexical>

<!ENTITY % pTrait_RefLexical

" id ID #REQUIRED

nieme_cposition NUMBER 0

nieme_cposant NUMBER 1">

<!-- Ces traits referent par leurs coordonnees (nieme_cposition
indique le rang dans la liste de Composition portee par l Usyn
et nieme_cposant le rang dans la liste mixte de R_ComposeUm/Usyn
portee par la Composition) l'Um ou Usyn entrant dans la
composition de l'Unite. La valeur 0 sur nieme_cposition refere

```

tous les composants de rang nieme_cpsant independamment des

compositions -->

<!ELEMENT Trait_RefLex - O EMPTY>

<!ATTLIST Trait_RefLex

%pTrait_RefLexical

saturesynt (%pBooleen) OUI>

<!ELEMENT Trait_RefIntrod - O EMPTY>

<!ATTLIST Trait_RefIntrod

%pTrait_RefLexical>

<!ELEMENT Trait_RefPrep - O EMPTY>

<!ATTLIST Trait_RefPrep

%pTrait_RefLexical>

<!ELEMENT Trait_RefConj - O EMPTY>

<!ATTLIST Trait_RefConj

%pTrait_RefLexical>

<!ELEMENT Trait_RefProRel - O EMPTY>

<!ATTLIST Trait_RefProRel

%pTrait_RefLexical>

<!ELEMENT Trait_RefProIntrog - O EMPTY>

<!ATTLIST Trait_RefProIntrog

%pTrait_RefLexical>

<!--***** Traits MORPHOLOGIQUES*****-->

<!--*****-->

<!ELEMENT Trait_Mode - O EMPTY>

<!ATTLIST Trait_Mode

```
id ID #REQUIRED

valeur (%pMode) INDICATIF>

<!ELEMENT Trait_Temps - O EMPTY>

<!ATTLIST Trait_Temps

id ID #REQUIRED

valeur (%pTemps|COMPOSE) PRESENT>

<!-- Le trait de temps permet d'exprimer les restrictions de temps liees
a certaines tournures.

Ex : etre arrive socialement

V[Lex:arriver]

[Temps:COMPOSE]

[Aux:ETRE] -->

<!ELEMENT Trait_Personne - O EMPTY>

<!ATTLIST Trait_Personne

id ID #REQUIRED

valeur (%pPersonne) 3>

<!ELEMENT Trait_Genre - O EMPTY>

<!ATTLIST Trait_Genre

id ID #REQUIRED

valeur (%pGenre) MASCULIN>

<!ELEMENT Trait_Nombre - O EMPTY>

<!ATTLIST Trait_Nombre

id ID #REQUIRED

valeur (%pNombre) SINGULIER>

<!ELEMENT Trait_NombrePasseur - O EMPTY>

<!ATTLIST Trait_NombrePasseur
```

```

id ID #REQUIRED

valeur (%pNombrePasseur) SINGULIER_POSSEUR>

<!--***** Traits MORPHO-SYNTAXIQUES *****-->
<!--*****-->

<!ELEMENT Trait_SsCatMorph - O EMPTY>

<!ATTLIST Trait_SsCatMorph

id ID #REQUIRED

valeur (%pSsCatGram) COMMUN>

<!-- Les valeurs des traits de sous-categorie morphologique
sont predefinies -->

<!ELEMENT Trait_SsCatSynt - O EMPTY>

<!ATTLIST Trait_SsCatSynt

id ID #REQUIRED

valeur (%pSsCatSynt
| %pSsCatSynt_cust) COORDONNE>

<!-- Les traits de sous-categorie syntaxique peuvent porter sur des
categories terminales ou non terminales et sont definis a facon,
en plus de certaines valeurs deja proposees dans le modele GENELEX.

Ex : un KILO de pommes

N[SsCatSynt:DETERMINATIF]

SN[SsCatSynt:DET_VIDE] -->

<!ELEMENT Trait_Aux - O EMPTY>

<!ATTLIST Trait_Aux

id ID #REQUIRED

valeur (%pAux) AVOIR

```

temps (SANS_T|%pTemps

|COMPOSE) SANS_T

mode (SANS_M|%pMode) SANS_M>

<!-- Ce trait permet d'associer a un verbe donne (l'entree decrite ou bien un verbe dans le contexte de l'entree) l'auxiliaire qui correspond a un emploi et qui doit lui etre associe.

Ex : se lever (etre leve) // lever (avoir leve)

V[Lex:lever] V[Lex:lever]

[Aux:ETRE] [Aux:AVOIR]

[Pronominal:SE]

L'apparition du Trait_Aux en conjonction avec un Trait_Temps portant la valeur COMPOSE indique que l'auxiliaire est necessairement present dans l'emploi que l'on decrit.

Ex : etre arrive

V[Lex:arriver]

[Aux:ETRE]

[Temps:COMPOSE]

Les attributs temps et mode precisent lorsque c'est necessaire le temps et le mode de l'auxiliaire lui-meme.

Ex : etant donne

V[Lex:donner]

[Temps:COMPOSE]

[Aux:ETRE[Mode:PARTICIPE][Temps:PRESENT]] -->

<!ELEMENT Trait_Pronominal - O EMPTY>

<!ATTLIST Trait_Pronominal

id ID #REQUIRED

valeur (%pPronominal) SE>

<!-- Ce trait permet d'associer a un verbe donne (l'entree decrite
ou bien un verbe dans le contexte de l'entree) la particule preverbale
NON REFERENTIELLE qui correspond a un emploi et qui doit lui etre
associee (Cf. "vrais" pronominaux).

Ex : s'en aller

V[Lex:aller]

[Pronominal:SE_EN] -->

<!ELEMENT Trait_Neg - O EMPTY>

<!ATTLIST Trait_Neg

id ID #REQUIRED

valeur (%pNeg) LIBRE>

<!-- La presence d'un Trait_Neg indique que l'emploi decrit est a la forme
negative ; on peut de plus preciser dans le champ valeur une restriction
sur la lexicalisation de la negation -->

<!ELEMENT Trait_Accord - O EMPTY>

<!ATTLIST Trait_Accord

id ID #REQUIRED

valeur (%pIJKL) I>

<!ELEMENT Trait_Passif - O EMPTY>

<!ATTLIST Trait_Passif

id ID #REQUIRED

valeur (%pBin) PLUS>

<!ELEMENT Trait_Tournure - O EMPTY>

<!ATTLIST Trait_Tournure

id ID #REQUIRED

valeur (%pTournure) INTERROGATIVE>

<!--***** Traits SYNTAXICO-SEMANTIQUES *****-->

<!--*****-->

<!ELEMENT Trait_Coref - O EMPTY>

<!ATTLIST Trait_Coref

id ID #REQUIRED

valeur (%pIJKL) I>

<!-- La coreference doit toujours etre resoluble : si un trait de valeur I est present, il existe au moins un autre trait de valeur I ou NON_I lui repondant.

Les traits Coref ne forcent pas la co-realisation des syntagmes les portant ; si on souhaite imposer cette co-realisation, on le fera comme d'habitude par l'usage de Conditions -->

<!--***** Traits SEMANTIQUES *****-->

<!--*****-->

<!ELEMENT Trait_Aspect - O EMPTY>

<!ATTLIST Trait_Aspect

id ID #REQUIRED

valeur (%pAspect) PROCESSIF>

<!ELEMENT Trait_Bin - O EMPTY>

<!ATTLIST Trait_Bin

id ID #REQUIRED

nom CDATA #REQUIRED

valeur (%pBin) PLUS>

```

<!-- Ce type de trait permet par exemple d'exprimer les
"conditions denotationnelles",
Ex : anime, humain -->

<!ELEMENT Trait_Libre - O EMPTY>

<!ATTLIST Trait_Libre

id ID #REQUIRED

nom CDATA #REQUIRED

valeur CDATA #REQUIRED>

<!-- Ce type de trait pourra etre exploite pour specifier des classes
ou des familles semantiques

Ex : nom : classe

valeur : vetement -->

<!-- Les Traits Bin et Libre pourront de plus etre utilises pour d'eventuels
autres traits non predefinis dans cette DTD -->

<!--***** ROLE THEMATIQUE *****-->

<!--*****-->

<!ELEMENT RoleTh - O EMPTY>

<!ATTLIST RoleTh

id ID #REQUIRED

valeur (%pRoleTh_cust) AGENT>

<!--***** FONCTION *****-->

<!--*****-->

<!ELEMENT Fonction - O EMPTY>

<!ATTLIST Fonction

id ID #REQUIRED

valeur (%pFonction_cust) TETE>

```



```
<!-- Les entites suffixees par _cust sont definies dans un fichier
"custom.ent" propre a l'utilisateur ; celui-ci peut ainsi ajouter
les valeurs d'attributs dont il souhaite disposer -->
```

3 : Entités *morpho.ent*

```
<!-- Consortium GENELEX @(#) morpho.ent 3.0 -->
```

```
<!-- *****A L'ADRESSE DES UTILISATEURS *****
```

Vos remarques concernant la DTD seront etudiees par le consortium GENELEX. Celui-ci assurera la diffusion de la nouvelle version qui pourrait en decouler.

```
***** -->
```

```
<!ENTITY % pBooleen "OUI|NON" >
```

```
<!ENTITY % pDatation "ARCHAIQUE|VIEILLI|MODERNE" >
```

```
<!ENTITY % pNiveauLgue "FAMILIER|VULGAIRE|ARGOTIQUE|POPULAIRE
|LITTERAIRE|SAVANT|STANDARD" >
```

```
<!ENTITY % pFrequence "RARE|COURANT" >
```

```
<!ENTITY % pCatGram "NOM|ADJECTIF|ADVERBE|VERBE|PREPOSITION
|CONJONCTION|INTERJECTION|DETERMINANT|PRONOM
|PARTICULE" >
```

```
<!ENTITY % pSsCatGram "PROPRE|COMMUN|POSSESSIF|DEMONSTRATIF
|PARTITIF|DEFINI|INDEFINI|CARDINAL|ORDINAL
```

```
|EXCLAMATIF|QUALIFICATIF|INTERROGATIF
```

```
|RELATIF|COMPLETIF|COORDINATION|SUBORDINATION
```

```
|PERSONNEL_FORT|PERSONNEL_FAIBLE|IMPERSONNEL
```

```

|COMPARATIF_EGALITE|COMPARATIF_SUPERIORITE

|COMPARATIF_INFERIORITE

|SUPERLATIF_SUPERIORITE|SUPERLATIF_INFERIORITE

|SUPERLATIF_ABSOLU">

<!ENTITY % pMode "INDICATIF|SUBJONCTIF|CONDITIONNEL|IMPERATIF

|INFINITIF|PARTICIPE" >

<!ENTITY % pTemps "PRESENT|IMPARFAIT|PASSE_SIMPLE|FUTUR|PASSE" >

<!ENTITY % pPersonne "1|2|3" >

<!ENTITY % pGenre "MASCULIN|FEMININ|NEUTRE" >

<!ENTITY % pNombre "SINGULIER|PLURIEL" >

<!ENTITY % pNombrePosseur "SINGULIER_POSSEUR|PLURIEL_POSSEUR" >

<!ENTITY % pTypaff "PREFIXE|SUFFIXE|INFIXE" >

<!ENTITY % pStatut "%pTypaff|BASE" >

<!ENTITY % pTypeBref "ABREVIATION|SIGLE|ACRONYME" >

<!ENTITY % pSeparg "TIRET|APOSTROPHE|ESPACE|JOINTURE|TIRET_ESPACE

|TIRET_JOINTURE|TIRET_ESPACE_JOINTURE

|APOSTROPHE_JOINTURE" >

<!ENTITY % pSeparp "LIAISON_t|LIAISON_z|LIAISON_k

|LIAISON_n|LIAISON_r|FRONTIERE_MOT" >

```

4 : Entités *syntaxe.ent*

```

<!--Consortium GENELEX @(#) syntaxe.ent 4.0 -->

```

```

<!-- *****A L'ADRESSE DES UTILISATEURS *****

```

Vos remarques concernant la DTD seront étudiées par le consortium GENELEX. Celui-ci assurera la diffusion de la nouvelle version qui

pourrait en découler.

*****-->

<!ENTITY % pEtiquetteSynt_T "%pCatGram|e">

<!ENTITY % pEtiquetteSynt_NT "P|Nbarre|SN|SV|SADJ|SADV|SP">

<!ENTITY % pSsCatSynt "RELATIVE|COMPLETIVE

|COORDONNE|SUBORDONNEE

|INTERROGATIVE_DRI|INTERROGATIVE_DRD

|COMPARATIF|SUPERLATIF

|TEMPS|LIEU|MANIERE|DEGRE|QUANTITE

|COPULE|DET_VIDE|DETERMINATIF">

<!ENTITY % pIJKL "I|J|K|L|NON_I|NON_J|NON_K|NON_L">

<!ENTITY % pAux "ETRE|AVOIR">

<!ENTITY % pPronominal "SE|LE|LA|LES|Y|EN|SE_LE|SE_LA|SE_LES

|SE_Y|SE_EN">

<!ENTITY % pNeg "LIBRE|NE_PAS|NE_PLUS|NE_JAMAIS|NE

|NE_GUERE|NE_POINT|NE_MAIS

|NE_QUE|NE_PAS_QUE|NE_PLUS_QUE

|NE_JAMAIS_QUE|NE_GUERE_QUE|NE_RIEN_QUE">

<!ENTITY % pAspect "PROCESSIF|RESULTATIF|STATIF">

<!ENTITY % pBin "PLUS|MOINS">

<!ENTITY % pTournure "INTERROGATIVE|EXCLAMATIVE">

<!ENTITY % pPortee "EXTERNE|INTERNE|INTERVENANT">

<!ENTITY % pTypeComposant "APPELANT|APPELANT_APPELE|APPELE"

<!ENTITY % pDestination "EXTERIEUR|INTERIEUR">

<!ENTITY % pTypeListPos "OUVERTE|FERMEE">

5 : Entités *custom.ent*

```

<!-- Consortium GENELEX @(#) custom.ent 3.0 -->

<!-- *****A L'ADRESSE DES UTILISATEURS *****

Vos remarques concernant la DTD seront etudiees par le consortium
GENELEX. Celui-ci assurera la diffusion de la nouvelle version qui
pourrait en decouler.

*****-->

<!-- Ce fichier est accessible a l'utilisateur qui peut y definir
des valeurs supplementaires d'attribut en remplaçant le texte
"_VALEURS_A_DEFINIR_" de l'entite par la liste des valeurs dont
il veut disposer : "VALEUR1|VALEUR2|...|VALEURn" -->

<!-- Pour l'instant, cette possibilite est retenue seulement pour
l'etiquette syntagmatique, la fonction, le role thematique et les
sous-categorisations syntaxiques.

Pour l'etiquette syntagmatique et la sous-categorisation, une liste
minimale de valeurs est definie dans le fichier "syntaxe.ent".

On pourra envisager d'etendre ce fonctionnement a d'autres traits. -->

<!ENTITY % pEtiquetteSynt_cust "_VALEURS_A_DEFINIR_" >

<!ENTITY % pSsCatSynt_cust "_VALEURS_SC_A_DEFINIR_" >

<!ENTITY % pFonction_cust

"TETE|SUJET|OBJET_DIRECT|OBJET_INDIRECT|ATTRIBUT_SUJET

|ATTRIBUT_OBJET|EPITHETE_GAUCHE|EPITHETE_DROIT

|SPECIFIEUR|MODIFIEUR|GENITIF" >

```

```
<!ENTITY % pRoleTh_cust
"AGENT | PATIENT | DESTINATAIRE | SOURCE | BUT | CAUSE | MANIERE
| LOCATIF | TEMPS | INSTRUMENT | THEME" >
```

6 : Contraintes *syntaxe.ctr*

```
<!--Consortium GENELEX @(#) syntaxe.ctr 4.0 -->
<!--CONTRAİNTE Usyn
combve TYPE CombVE
(description
|description_1) TYPE Description
transfdescription_1 TYPE TransfDescription -->
<!--CONTRAİNTE Description
self TYPE Self
construction TYPE Construction -->
<!--CONTRAİNTE Self
(syntagme_nt_s
|syntagme_nt_s_1) TYPE Syntagme_NT_S
transfsyntagme_1 TYPE TransfSyntagme
intervconst TYPE IntervConst
comportappele_1 TYPE ComportAppele -->
<!--CONTRAİNTE IntervConst
fonction TYPE Fonction
roleth_1 TYPE RoleTh
syntagme_t_1 TYPE Syntagme_T -->
```

```
<!--CONTRAINTE ComportAppele
fonction TYPE Fonction
roleth_l TYPE RoleTh
syntagme_t TYPE Syntagme_T -->
<!--CONTRAINTE Construction
optionnalite TYPE Optionnalite
trait_l TYPE (Trait_Lex|Trait_Introd
|Trait_Prep|Trait_Conj
|Trait_ProRel
|Trait_ProIntrog
|Trait_Mode|Trait_Temps
|Trait_Personne|Trait_Genre
|Trait_Nombre
|Trait_NombrePosseur
|Trait_SsCatMorph
|Trait_SsCatSynt
|Trait_Aux|Trait_Pronominal
|Trait_Neg|Trait_Accord
|Trait_Passif|Trait_Tournure
|Trait_Coref|Trait_Aspect
|Trait_Bin|Trait_Libre)
position_c_l TYPE Position_C -->
<!--CONTRAINTE Position_C
fonction TYPE Fonction
roleth_l TYPE RoleTh
syntagme_c_l TYPE (Syntagme_T|Syntagme_NT_C)
```

```

transfsyntagme_l TYPE TransfSyntagme -->

<!--CONTRAINTE Position_S

fonction TYPE Fonction

roleth_l TYPE RoleTh

syntagme_s_l TYPE (Syntagme_T|Syntagme_NT_S)

transfsyntagme_l TYPE TransfSyntagme -->

<!--CONTRAINTE Insertion

retire_syntagme_c_l TYPE (Syntagme_T|Syntagme_NT_C)

retire_transfsyntagme_l TYPE TransfSyntagme -->

<!--CONTRAINTE CheminSyntagme

syntagme TYPE (Syntagme_T|Syntagme_NT_C
|Syntagme_NT_S) -->

<!--CONTRAINTE SelectIntervConst

syntagme_t TYPE Syntagme_T -->

<!--CONTRAINTE Syntagme_T

trait_l TYPE (Trait_Lex|Trait_RefLex
|Trait_Mode|Trait_Temps
|Trait_Personne|Trait_Genre
|Trait_Nombre
|Trait_NombrePosseur
|Trait_SsCatMorph
|Trait_SsCatSynt
|Trait_Aux|Trait_Pronominal
|Trait_Neg|Trait_Accord
|Trait_Passif|Trait_Tournure

```

|Trait_Coref|Trait_Aspect

|Trait_Bin|Trait_Libre) -->

<!--CONTRAINTE Syntagme_NT_C

optionnalite TYPE Optionnalite

trait_l TYPE (Trait_Lex|Trait_Introd

|Trait_Prep|Trait_Conj

|Trait_ProRel

|Trait_ProIntrog

|Trait_Mode|Trait_Temps

|Trait_Personne|Trait_Genre

|Trait_Nombre

|Trait_NombrePosseur

|Trait_SsCatMorph

|Trait_SsCatSynt

|Trait_Aux|Trait_Pronominal

|Trait_Neg|Trait_Accord

|Trait_Passif|Trait_Tournure

|Trait_Coref|Trait_Aspect

|Trait_Bin|Trait_Libre)

position_c_l TYPE Position_C -->

<!--CONTRAINTE Syntagme_NT_S

optionnalite TYPE Optionnalite

trait_l TYPE (Trait_Lex|Trait_Introd

|Trait_Prep|Trait_Conj

|Trait_ProRel

|Trait_ProIntrog

|Trait_RefLex

|Trait_RefIntrod

|Trait_RefPrep|Trait_RefConj

|Trait_RefProRel

|Trait_RefProIntrog

|Trait_Mode|Trait_Temps

|Trait_Personne|Trait_Genre

|Trait_Nombre

|Trait_NombrePosseur

|Trait_SsCatMorph

|Trait_SsCatSynt

|Trait_Aux|Trait_Pronominal

|Trait_Neg|Trait_Accord

|Trait_Passif|Trait_Tournure

|Trait_Coref|Trait_Aspect

|Trait_Bin|Trait_Libre)

position_s_l TYPE Position_S

insertion_l TYPE Insertion -->

<!--CONTRAINTE R_ComposeUm

um TYPE (Um_S|Um_Agg|Um_C) -->

<!--CONTRAINTE R_ComposeUsyn

usyn TYPE Usyn

mdc TYPE MdC -->

<!--CONTRAINTE HeritePosition

modifposition TYPE ModifPosition -->

```
<!--CONTRAINTE FiltreSelf
modifintervconst TYPE ModifIntervConst
modifsyntagme_nt TYPE ModifSyntagme_NT -->
<!--CONTRAINTE TransfUsyn
usyn_resultat TYPE Usyn -->
<!--CONTRAINTE TransfDescription
(description_origine
|description_resultat) TYPE Description -->
<!--CONTRAINTE ModifDescription
modifconstruction TYPE ModifConstruction
modifintervconst TYPE ModifIntervConst
modifsyntagme_nt TYPE ModifSyntagme_NT -->
<!--CONTRAINTE ModifConstruction
optionnalite TYPE Optionnalite
(retire_trait_l
|ajoute_trait_l) TYPE (Trait_Lex|Trait_Introd
|Trait_Prep|Trait_Conj
|Trait_ProRel
|Trait_ProIntrog
|Trait_Mode|Trait_Temps
|Trait_Personne|Trait_Genre
|Trait_Nombre
|Trait_NombrePosseur
|Trait_SsCatMorph
|Trait_SsCatSynt
```

```
|Trait_Aux|Trait_Pronominal
```

```
|Trait_Neg|Trait_Accord
```

```
|Trait_Passif|Trait_Tournure
```

```
|Trait_Coref|Trait_Aspect
```

```
|Trait_Bin|Trait_Libre) -->
```

```
<!--CONTRAINTE TransfPosition
```

```
modifposition TYPE ModifPosition -->
```

```
<!--CONTRAINTE ModifPosition
```

```
fonction TYPE Fonction
```

```
roleth_l TYPE RoleTh
```

```
(retire_syntagme_l
```

```
|ajoute_syntagme_l) TYPE (Syntagme_T|Syntagme_NT_C
```

```
|Syntagme_NT_S)
```

```
(retire_transfsyntagme_l
```

```
|ajoute_transfsyntagme_l
```

```
|transfsyntagme_l) TYPE TransfSyntagme -->
```

```
<!--CONTRAINTE TransfSyntagme
```

```
(syntagme_origine
```

```
|syntagme_resultat) TYPE (Syntagme_T|Syntagme_NT_C
```

```
|Syntagme_NT_S)
```

```
modifsyntagme TYPE (ModifSyntagme_T
```

```
|ModifSyntagme_NT) -->
```

```
<!--CONTRAINTE ModifSyntagme_T
```

```
(retire_trait_l
```

```
|ajoute_trait_l TYPE (Trait_Lex|Trait_RefLex
```

```
|Trait_Personne|Trait_Genre
```

```
|Trait_Nombre  
  
|Trait_NombrePosseur  
  
|Trait_SsCatMorph  
  
|Trait_SsCatSynt  
  
|Trait_Aux|Trait_Pronominal  
  
|Trait_Neg|Trait_Accord  
  
|Trait_Passif|Trait_Tournure  
  
|Trait_Coref|Trait_Aspect  
  
|Trait_Bin|Trait_Libre) -->  
  
<!--CONTRAINTE ModifSyntagme_NT  
  
(retire_trait_l  
  
|ajoute_trait_l) TYPE (Trait_Lex|Trait_Introd  
  
|Trait_Prep|Trait_Conj  
  
|Trait_ProRel  
  
|Trait_ProIntrog  
  
|Trait_RefLex  
  
|Trait_RefIntrod  
  
|Trait_RefPrep|Trait_RefConj  
  
|Trait_RefProRel  
  
|Trait_RefProIntrog  
  
|Trait_Mode|Trait_Temps  
  
|Trait_Personne|Trait_Genre  
  
|Trait_Nombre  
  
|Trait_NombrePosseur  
  
|Trait_SsCatMorph
```

```

|Trait_SsCatSynt

|Trait_Aux|Trait_Pronominal

|Trait_Neg|Trait_Accord

|Trait_Passif|Trait_Tournure

|Trait_Coref|Trait_Aspect

|Trait_Bin|Trait_Libre)

optionnalite TYPE Optionnalite -->

<!--CONTRAINTE ModifIntervConst

fonction TYPE Fonction

roleth_l TYPE RoleTh

(retire_syntagme_t_l

|ajoute_syntagme_t_l) TYPE Syntagme_T

transfsyntagme_l TYPE TransfSyntagme -->

<!--CONTRAINTE (Trait_Lex|Trait_Introd|Trait_Prep|Trait_Conj

|Trait_ProRel|Trait_ProIntrog)

um TYPE (Um_S|Um_Agg|Um_C) -->

```

III : Exemples de données balisées

L'exemple de balisage SGML donné ici reprend de façon un peu plus détaillée les deux manières de coder l'Usyn composée "*abattre carte ma"tresse"*", telles qu'on peut les trouver dans l'exemple (13) de la partie D de ce document (Annexes aux Unités Syntaxiques Composées).

```

<GENELEX nom="ABATTRE CARTE MAITRESSE" langue="FRANCAIS">

<GENELEXMORPHO>

<!-- ***** -->

<!-- ***** -->

<!-- Les Unites Morphologiques -->

```

```
<!-- ***** -->

<!-- ***** -->

<UM_S id="UM2556" catgram="VERBE" usyn_l="USYN2556">

<UMG>abattre</>

</>

<UM_S id="UM4570" catgram="NOM" sscatgram="COMMUN" usyn_l="USYN4570">

<UMG>carte</>

</>

<UM_S id="UM6734" catgram="ADJECTIF" sscatgram="QUALIFICATIF"
usyn_l="USYN6734">

<UMG>ma&icirc;tre</>

</>

</GENELEXMORPHO>

<GENELEXSYNTAXE>

<!-- ***** -->

<!-- Fonctions -->

<!-- ***** -->

<FONCTION id="FT1" valeur="TETE">

<FONCTION id="FT2" valeur="SUJET">

<FONCTION id="FT3" valeur="OBJET_DIRECT">

<FONCTION id="FT7" valeur="EPITHETE_GAUCHE">

<FONCTION id="FT9" valeur="SPECIFIEUR">

<!-- ***** -->

<!-- Traits Restrictifs -->

<!-- ***** -->

<TRAIT_REFLEX id="TR111" nieme_cposition="1" nieme_cposant="1" saturesynt="OUI">
```

```

<TRAIT_REFLEX id="TR112" nieme_cposition="1" nieme_cposant="2" saturesynt="OUI">
<TRAIT_REFLEX id="TR113" nieme_cposition="1" nieme_cposant="3" saturesynt="OUI">
<TRAIT_REFLEX id="TR114" nieme_cposition="1" nieme_cposant="2" saturesynt="NON">
<TRAIT_NOMBRE id="TR202" valeur="SINGULIER">
<TRAIT_LEX id="TR320" valeur="il">
<TRAIT_LEX id="TR321" valeur="le">
<TRAIT_AUX id="TR500" valeur="AVOIR">
<TRAIT_SSCATSYNT id="TR634" valeur="RELATIVE">
<!-- ***** -->
<!-- Syntagmes Terminaux -->
<!-- ***** -->
<SYNTAGME_T id="ST1" etiquettesynt="DETERMINANT">
<SYNTAGME_T id="ST3" etiquettesynt="VERBE">
<SYNTAGME_T id="ST8" etiquettesynt="ADJECTIF">
<SYNTAGME_T id="ST9" etiquettesynt="NOM">
<SYNTAGME_T id="ST10" etiquettesynt="PRONOM" trait_l="TR320">
<SYNTAGME_T id="ST11" etiquettesynt="PRONOM" trait_l="TR321">
<SYNTAGME_T id="ST20" etiquettesynt="VERBE" trait_l="TR500">
<SYNTAGME_T id="ST26" etiquettesynt="NOM" trait_l="TR202">
<SYNTAGME_T id="ST56" etiquettesynt="ADJECTIF" trait_l="TR112">
<SYNTAGME_T id="ST59" etiquettesynt="NOM" trait_l="TR111">
<SYNTAGME_T id="ST60" etiquettesynt="VERBE" trait_l="TR111">
<SYNTAGME_T id="ST68" etiquettesynt="NOM" trait_l="TR112 TR202">
<SYNTAGME_T id="ST69" etiquettesynt="ADJECTIF" trait_l="TR113">
<!-- ***** -->

```

```

<!-- IntervConst -->

<!-- ***** -->

<INTERVCONST id="INTC10" syntagme_t_l="ST9">

<INTERVCONST id="INTC24" syntagme_t_l="ST3">

<INTERVCONST id="INTC28" syntagme_t_l="ST20">

<INTERVCONST id="INTC45" syntagme_t_l="ST8">

<!-- ***** -->

<!-- Syntagmes non Terminaux de Construction -->

<!-- ***** -->

<SYNTAGME_NT_C id="SNTC1" etiquettesynt="SN">

<SYNTAGME_NT_C id="SNTC6" etiquettesynt="P" trait_l="TR634">

<SYNTAGME_NT_C id="SNTC18" etiquettesynt="SADJ">

<SYNTAGME_NT_C id="SNTC25" etiquettesynt="SP">

<SYNTAGME_NT_C id="SNTC74" etiquettesynt="SADV">

<!-- ***** -->

<!-- Syntagmes non Terminaux de Structure -->

<!-- ***** -->

<SYNTAGME_NT_S id="SNTS562" etiquettesynt="Nbarre"

trait_l="TR202" position_s_l="POSS524 POSS525" optionnalite="OPT2">

<SYNTAGME_NT_S id="SNTS108" etiquettesynt="SV"

position_s_l="POSS250 POSS611" optionnalite="OPT2">

<SYNTAGME_NT_S id="SNTS889" etiquettesynt="SN"

position_s_l="POSS45 POSS46 POSS47 POSS48" optionnalite="OPT33">

<SYNTAGME_NT_S id="SNTS11" etiquettesynt="SADJ">

<SYNTAGME_NT_S id="SNTS109" etiquettesynt="SV"

position_s_l="POSS250 POSS612" optionnalite="OPT2">

```



```

<SYNTAGME_NT_S id="SNTS890" etiquettesynt="SN"
position_s_l="POSS45 POSS46 POSS47" optionnalite="OPT27">

<!-- ***** -->

<!-- ModifSyntagmes Terminaux et non Terminaux -->

<!-- ***** -->

<MODIFSYNTAGME_T id="MODST12" ajoute_trait_l="TR202">

<MODIFSYNTAGME_NT id="MODSNT49" ajoute_trait_l="TR112">

<MODIFSYNTAGME_NT id="MODSNT57" ajoute_trait_l="TR113">

<MODIFSYNTAGME_NT id="MODSNT79" ajoute_trait_l="TR114 TR202">

<MODIFSYNTAGME_NT id="MODSNT80" ajoute_trait_l="TR114">

<!-- ***** -->

<!-- Optionnalites -->

<!-- ***** -->

<OPTIONNALITE id="OPT1" libelle= "P0"></>

<OPTIONNALITE id="OPT2" libelle= "P0 P1"></>

<OPTIONNALITE id="OPT3" libelle= "P0 (P1)"></>

<OPTIONNALITE id="OPT15" libelle="(P0)"></>

<OPTIONNALITE id="OPT25" libelle="P0 (P1) (P2)"></>

<OPTIONNALITE id="OPT27" libelle= "P0 (P1) P2"></>

<OPTIONNALITE id="OPT33" libelle= "P0 (P1) P2 P3"></>

<!-- ***** -->

<!-- TransfSyntagmes -->

<!-- ***** -->

<TRANSFSYNTAGME id="TRSYNT22"
commentaire="pronominalisation SN sujet en il"

```

```

syntagme_origine="SNTC1" syntagme_resultat="ST10">

<TRANSFSYNTAGME id="TRSYNT23"

commentaire="pronominalisation SN objet en le"

syntagme_origine="SNTC1" syntagme_resultat="ST11">

<TRANSFSYNTAGME id="TRSYNT213"

commentaire="lexicalisation partielle SN par 2eme composant + sing"

syntagme_origine="SNTC1" modifsyntagme="MODSNT79">

<TRANSFSYNTAGME id="TRSYNT214"

commentaire="lexicalisation partielle SN par 2eme composant"

syntagme_origine="SNTC1" modifsyntagme="MODSNT80">

<TRANSFSYNTAGME id="TRSYNT332"

commentaire="restriction d'un NOM au singulier"

syntagme_origine="ST9" modifsyntagme="MODST12">

<TRANSFSYNTAGME id="TRSYNT664"

commentaire="saturation SADJ par 2eme composant"

syntagme_origine="SNTC18" modifsyntagme="MODSNT49">

<TRANSFSYNTAGME id="TRSYNT678"

commentaire="saturation SADJ par 3eme composant"

syntagme_origine="SNTC18" modifsyntagme="MODSNT57">

<!-- ***** -->

<!-- ModifIntervConst -->

<!-- ***** -->

<MODIFINTERVCONST id="MODIC122" transfsyntagme_l="TRSYNT332">

<!-- ***** -->

<!-- Positions de Construction -->

```

```
<!-- ***** -->

<POSITION_C id="POSC52" fonction="FT2"
syntagme_c_l="SNTC1 ST10" transfsyntagme_l="TRSYNT22">

<POSITION_C id="POSC48" fonction="FT3"
syntagme_c_l="SNTC1 ST11" transfsyntagme_l="TRSYNT23">

<POSITION_C id="POSC56" syntagme_c_l="ST1">

<POSITION_C id="POSC87" repetable="OUI" fonction="FT7"
syntagme_c_l="SNTC18">

<POSITION_C id="POSC35" repetable="OUI"
syntagme_c_l="SNTC18 SNTC25 SNTC6">

<POSITION_C id="POSC98" repetable="OUI" fonction="FT9"
syntagme_c_l="SNTC74">

<!-- ***** -->

<!-- Positions de Structure -->

<!-- ***** -->

<POSITION_S id="POSS524" repetable="NON" syntagme_s_l="ST59">

<POSITION_S id="POSS525" repetable="NON" syntagme_s_l="ST56">

<POSITION_S id="POSS250" repetable="NON" fonction="FT1"
syntagme_s_l="ST60">

<POSITION_S id="POSS611" repetable="NON" fonction="FT3"
syntagme_s_l="SNTS889">

<POSITION_S id="POSS45" repetable="NON" syntagme_s_l="ST1">

<POSITION_S id="POSS46" repetable="OUI" syntagme_s_l="SNTS11">

<POSITION_S id="POSS47" repetable="NON" fonction="FT1"
syntagme_s_l="ST68">

<POSITION_S id="POSS48" repetable="NON" syntagme_s_l="ST69">
```

```
<POSITION_S id="POSS612" repetable="NON" fonction="FT3"
syntagme_s_l="SNTS890">
<!-- ***** -->
<!-- ModifPositions -->
<!-- ***** -->
<MODIFPOSITION id="MODPOS151" repetable="NON"
retire_syntagme_l="SNTC6 SNTC25" transfsyntagme_l="TRSYNT664">
<MODIFPOSITION id="MODPOS152" repetable="NON"
retire_syntagme_l="SNTC6 SNTC25" transfsyntagme_l="TRSYNT678">
<MODIFPOSITION id="MODPOS299" retire_syntagme_l="ST11"
retire_transfsyntagme_l="TRSYNT23" transfsyntagme_l="TRSYNT213">
<MODIFPOSITION id="MODPOS300" retire_syntagme_l="ST11"
retire_transfsyntagme_l="TRSYNT23" transfsyntagme_l="TRSYNT214">
<!-- ***** -->
<!-- Self -->
<!-- ***** -->
<SELF id="SELF128" intervconst="INTC28">
<SELF id="SELF250" intervconst="INTC10">
<SELF id="SELF774" intervconst="INTC45">
<SELF id="SELF752" intervconst="INTC10" syntagme_nt_s="SNTS562">
<SELF id="SELF989" intervconst="INTC24" syntagme_nt_s="SNTS108">
<SELF id="SELF990" intervconst="INTC24" syntagme_nt_s="SNTS109">
<!-- ***** -->
<!-- Constructions -->
<!-- ***** -->
```

```

<CONSTRUCTION id="CONST8585" etiquettesynt="P" insereself="1"
position_c_l="POSC52 POSC48" optionnalite="OPT2">

<CONSTRUCTION id="CONST55" etiquettesynt="SN" insereself="2"
position_c_l="POSC56 POSC87 POSC35" optionnalite="OPT25">

<CONSTRUCTION id="CONST122" etiquettesynt="SADJ" insereself="1"
position_c_l="POSC98" optionnalite="OPT15">

<CONSTRUCTION id="CONST5811" etiquettesynt="SN" insereself="2"
position_c_l="POSC56 POSC87" optionnalite="OPT3">

<CONSTRUCTION id="CONST7627" etiquettesynt="P" insereself="1"
position_c_l="POSC52" optionnalite="OPT1">

<!-- ***** -->

<!-- Descriptions -->

<!-- ***** -->

<DESCRIPTION id="DESC3547" self="SELF128" construction="CONST8585"></>

<DESCRIPTION id="DESC1552" self="SELF250" construction="CONST55"></>

<DESCRIPTION id="DESC9240" self="SELF774" construction="CONST122"></>

<DESCRIPTION id="DESC6992" self="SELF752" construction="CONST5811"></>

<DESCRIPTION id="DESC4142" self="SELF989" construction="CONST7627"></>

<DESCRIPTION id="DESC4143" self="SELF990" construction="CONST7627"></>

<!-- ***** -->

<!-- Mode de Composition -->

<!-- ***** -->

<MDC id="MDC258" exemple="carte sur carte_maitresse">

<HERITEPOSITION destination="EXTERIEUR">

<CHEMINPOSITION><POSITIONBUT nieme_position="0"></>

</>

```

<HERITEPOSITION destination="EXTERIEUR">

<CHEMINPOSITION><POSITIONBUT nieme_position="1"></>

</>

<HERITEPOSITION destination="INTERIEUR" optionnel="NON"

modifposition="MODPOS151">

<CHEMINPOSITION><POSITIONBUT nieme_position="2"></>

</>

</>

<MDC id="MDC117" exemple="abattre sur abattre carte maitresse">

<HERITEPOSITION destination="EXTERIEUR">

<CHEMINPOSITION><POSITIONBUT nieme_position="0"></>

</>

<HERITEPOSITION destination="INTERIEUR" modifposition="MODPOS299">

<CHEMINPOSITION><POSITIONBUT nieme_position="1"></>

</>

</>

<MDC id="MDC600" exemple="carte sur abattre carte maitresse">

<FILTRESELF modifintervconst="MODIC122">

<HERITEPOSITION destination="INTERIEUR">

<CHEMINPOSITION><POSITIONBUT nieme_position="0"></>

</>

<HERITEPOSITION destination="INTERIEUR">

<CHEMINPOSITION><POSITIONBUT nieme_position="1"></>

</>

<HERITEPOSITION destination="INTERIEUR" optionnel="NON"

```
modifposition="MODPOS152">

<CHEMINPOSITION><POSITIONBUT nieme_position="2"></>

</>

</>

<MDC id="MDC642" exemple="carte_maitresse sur abattre carte_maitresse">

<FILTRESELF modifintervconst="MODIC122">

<HERITEPOSITION destination="INTERIEUR">

<CHEMINPOSITION><POSITIONBUT nieme_position="0"></>

</>

<HERITEPOSITION destination="INTERIEUR">

<CHEMINPOSITION><POSITIONBUT nieme_position="1"></>

</>

</>

<!-- ***** -->

<!-- ***** -->

<!-- Les Unites Syntaxiques -->

<!-- ***** -->

<!-- ***** -->

<!-- ***** -->

<!-- USyn simple abattre -->

<!-- ***** -->

<USYN id="USYN2556" description="DESC3547"

commentaire="abattre transitif"></>

<!-- ***** -->

<!-- USyn simple carte -->
```

```
<!-- ***** -->

<USYN id="USYN4570" description="DESC1552" commentaire="carte"></>

<!-- ***** -->

<!-- USyn associee a Um_S maitre -->

<!-- ***** -->

<USYN id="USYN6734" description="DESC9240"></>

<!-- ***** -->

<!-- USyn composee carte_maitresse (USyn carte + Um maitre) -->

<!-- ***** -->

<USYN id="USYN6789" description="DESC6992"
commentaire="carte ma&icirc;tresse">
<COMPOSITION>
<R_COMPOSEUSYN type="APPELANT" usyn="USYN4570" mdc="MDC258">
<R_COMPOSEUM type="APPELE" um="UM6734">
</>
</>

<!-- ***** -->

<!-- USyn composee abattre_carte_maitresse -->

<!-- -->

<!-- Solution 1 -->

<!-- -->

<!-- (USyn abattre + Usyn carte + Um maitre) -->

<!-- ***** -->

<USYN id="USYN4851" description="DESC4142"
commentaire="abattre carte maitresse">
<COMPOSITION>
```


<R_COMPOSEUSYN type="APPELANT" usyn="USYN2556" mdc="MDC117">

<R_COMPOSEUSYN type="APPELANT_APPELE" usyn="USYN4570" mdc="MDC600">

<R_COMPOSEUM type="APPELE" um="UM6734">

</>

</>

<!-- ***** -->

<!-- USyn composee abattre_carte_maitresse -->

<!-- -->

<!-- Solution 2 -->

<!-- -->

<!-- (USyn abattre + Usyn carte_maitresse) -->

<!-- ***** -->

<USYN id="USYN4852" description="DESC4143"

commentaire="abattre carte_maitresse">

<COMPOSITION>

<R_COMPOSEUSYN type="APPELANT" usyn="USYN2556" mdc="MDC117">

<R_COMPOSEUSYN type="APPELE" usyn="USYN6789" mdc="MDC642">

</>

</>

</GENELEXSYNTAXE>

</GENELEX>