**PROJECT EUREKA**

# GENELEX

## Report on the

## SYNTACTIC LAYER

## GENELEX Consortium

**Version 4.0**

**December 1, 1993**

# Table of contents

A - Preliminaries 1

**1: Generic model, "theory-welcoming" model 3**

1.1: Lexicographic framework and recording formalism 3

1.2: A generic "theory-welcoming" model 4

1.3: Expressing different descriptions using the formalism of GENELEX 4

1.4: Consistent merging of different descriptions 5

1.4.1: Complementary linguistic facts 5

1.4.2: Competitive but not contradictory linguistic facts 6

1.4.3: Competitive and contradictory linguistic facts 8

**2: Coding homogeneity 9**

**3: Lexicon - Grammar articulation 9**

3.1: Compatibility between the lexicon and the grammar 10

3.1.1: Theoretical approach and formalism 10

3.1.2: Minimum compatibility required by the GENELEX model 12

3.2: Cross-references between the lexicon and the grammar 12

**4: Syntax autonomy 14**

B - Analysis of simple syntactic units 15

**I: Articulation with the Morphological Layer 16**

**II: Syntactic Units (Usyns) 17**

**1: General definition 17**

**2: Atomist viewpoint 18**

**3: Syntactist viewpoint 19**

**4: Criteria for splitting Usyns 20**

**III: Descriptions 22**

**IV: Constructions 23**

**V: Self 24**

**1: General definition 24**

**2: Self intervening in the construction of the entry 24**

**3: Self called by an element external to the construction of the entry 26**

**VI: Positions 27**

**1: General approach and justification 27**

1.1: Example of verb complementation 27

1.2: Emergence of the notion of Position 27

1.3: Case of left determination 29

**2: Application to other categories 30**

2.1: Nouns 30

2.2: Adjectives 31

2.3: Prepositions 32

2.4: Determiners 32

2.5: Adverbs 32

**3: Definition of the Position 33**

3.1: Formal definition 33

3.2: Positions and Distributions 34

3.2.1: Position Occupants 35

3.2.2: Sharing of Positions 35

3.3: Position and Functions 36

3.4: Positions and Thematic Roles 37

**4: Positions and Constructions 38**

4.1: Optionality 38

4.1.1: Simple options 38

4.1.2: Distant or interdependent options 38

4.2: Actualization of Positions 41

4.2.1: Linguistic phenomena 41

4.2.2: Indicator of repeatability on Positions 43

4.3: Linearity 44

4.3.1: Canonical linearity 44

4.3.2: Linearity and transformations 45

4.3.3: Linearity and option 45

4.4: Surface phenomena 45

4.5: Interdependency of Positions 46

4.6: Realizations of inter-conditioned Positions 47

4.7: Conditions between Self and Positions 48

**VII: Phrases 49**

**1: General definition 49**

**2: Syntagmatic Label 49**

**3: Feature Constraints 50**

**4: Structure constraints 50**

4.1: Syntactic sub-categorization feature 50

4.2: Tree-structured rewriting 51

4.2.1: Description 51

4.2.2: Examples of use 51

4.3: Partial rewriting of Phrases 55

**VIII: Features 56**

**1: Types, sub-types and values of features 56**

1.1: Lexical features 56

1.1.1: Introducers (Prep, Conj, ProRel, ProIntrog and Introd) 56

1.1.2: Lex 57

1.1.3: Cooccurrence of lexical features 58

1.1.4: Particular case of Agglutinates 58

1.2: Morphological features 59

1.3: Morpho-syntactic features 59

1.3.1: SsCatMorph 60

1.3.2: Aux 61

1.3.3: Pronominal 61

1.3.4: Accord 62

1.4: Syntactic features 62

1.4.1: SsCatSynt 63

1.4.2: Tournure 63

1.4.3: Neg 64

1.4.4: Passif 64

1.5: Syntactico-semantic features 64

1.5.1: Coref 64

1.6: Semantic features 68

1.6.1: Aspect of the verb 68

1.6.2: Denotative Conditions (Bin features) 68

1.6.3: Semantic class (Libre features) 69

**2: Constraints of integrity on Features 70**

2.1: General rules for using Features 70

2.2: Subtypes of Features authorized depending on the syntagmatic label 71

2.3: Values of Sub-category Features authorized depending on the syntagmatic label 75

2.4: Possible conjunctions of Features 78

## IX: Transformations 79

**1: Transformations between Usyns 80**

1.1: Transformations between Usyns drawn from the same Um 80

1.2: Transformations between Usyns drawn from different Ums 80

1.3: Calculation: Transformation Mode 81

**2: Transformations between Descriptions 82**

2.1: Use and use context 82

2.1.1: Example of passivation 83

2.1.2: Example of noun clause restructuring 84

2.2: Inheritance of positions: observations 85

2.3: Calculation: Transformation Mode 85

2.3.1: Explicitation of the process 86

2.3.2: Exemplification of the process 87

**3: Transformations between Phrases occupying the same position 88**

C - Analysis of compound syntactic units 89

**I. Introduction 90**

**1. Recording of compounds 90**

**2. Electronic dictionaries, editorial dictionaries 90**

**3. Coverage of compounds in GENELEX 91**

**II: Characteristics of syntactic compounds 93**

**1: Well-constructed phrases 93**

1.1: Lexicalization 93

1.2: Inflection and agreement 94

1.3: Realization alternatives 94

1.4: Deletion 95

1.5: Modification 96

1.6: Displacement 96

1.7: Transformations 97

1.8: Coordination 98

**2: Fossilization 98**

2.1: Total or partial fossilization 98

2.2: Interdependent constituents 99

**3: Inside and outside of the compound 99**

3.1: Inclusion in a terminal category 99

3.2. Distinction between the inside and the outside of a compound 101

3.3. Boundary between the outside and the inside of a compound 101

**4. Inheritance of the syntactic properties of components 102**

4.1. Inheritance of constructions 102

4.2. Inheritance of the characteristics specific to the compound 104

**5. Support verbs 104**

**III.Principles of formal representation 105**

1. Representation axes 105

2. Atomist viewpoint vs. syntactist viewpoint 106

**IV. Lists of components 107**

1. Type and choice of components 107

2.2. Alternatives of components 108

3. Reference to components 109

**V. Composition Mode (MdC) 110**

1. Interest 110

2. Inhibition, inheritance and filtering 110

    2.1. Inhibition, inheritance and filtering of Positions 110

    2.2. Self filtering 110

    2.3. Example 111

3. Calls of components for lexicalization 111

    3.1. Notion of call 111

    3.2. Head of the syntactic compound 112

    3.3. Lexicalization 113

    3.4. Compatibility between the categories of the calling phrase and those of the called component 113

    3.5. Unification 113

**VI. Internal structure of compound 115**

1. Choosing the category 115

2. Structure Phrases (Syntagme_NT_S) 115

3. Insertions 117

4. Interdependency 117

5. Restricting features 118

5.1. Lexical features 118

5.2. Morphological features 119

5.3. Accord and Coref features 119

5.4. Semantic features 120

6. Conditions 120

7. Transformations 121

**VII. Construction 122**

D - Annex to compound Syntactic Units 124

E - Bibliography 153

F - User's Manual 156

**1: Usyn 157**

1.1: General 157

1.2: Simple Usyn 157

1.3: Compound Usyn 157

**2: Descriptions 157**

**3: Self 158**

3.1: Simple Self 158

3.1.1: Self intervening in the Construction (IntervConst) 159

3.1.2: Self called by an element external to its Construction (ComportAppele) 160

3.2: Compound Self 160

**4: Constructions 161**

4.1: General definition 161

4.2: Construction Skeletons 163

**5: Positions 163**

**6: Insertions 163**

**7: Conditions 164**

7.1: General properties 164

7.2: Conditions on the option of Positions 165

7.3: Conditions on the realizations of Position 165

**8: Phrases 166**

**9: Features 167**

9.1: Lexical Features 168

9.1.1: Family of Lex features 168

9.1.2: Family of the Reflex Features 169

9.2: Morphological features 171

9.3: Morpho-syntactic features 172

9.4: Syntactic Features 174

9.5.: Syntactic-semantic Features 175

9.6: Semantic Features 175

**10: Transformations 176**

10.1: TransfUsyn and TransfDescription 176

10.2: TransfUsyn 177

10.3: TransfDescription 177

10.4: TransfPosition 178

10.5: TransfSyntagme 178

**11: Modifs 178**

11.1: ModifDescription 179

11.2: ModifConstruction 179

11.3: ModifPosition 179

11.4: ModifSyntagme 180

11.5: ModifIntervConst 180

**12: R_Compose and Composition 181**

**13: Composition Mode 182**

13.1: MdC 182

13.2: HeritePosition 182

13.3: FiltreSelf 183

**14: Structure (Syntagme_NT_S) 183**

**15: Appellation, commentaire, exemple 184**

G - Entity-Relation diagrams 185

H - SGML DTD 196

**I : Introduction - Translation of the conceptual model 197**

**II: DTD GENELEX commented 198**

**1: DTD genelex.dtd 198**

**2: DTD syntaxe.dtd 200**

**3 : Entities morpho.ent 221**

**4 : Entities syntaxe.ent 222**

**5 : Entities custom.ent 223**

**6 : Constraints syntaxe.ctr 224**

**III : Examples of tagged data 230**

# A - Preliminaries

This report presents the work realized by all partners of the consortium GENELEX France for the definition of a model describing the syntactic part of an electronic dictionary.

It follows after the presentation document on morphology written by the same consortium, and called "MORPHOLOGICAL LAYER Version 3.0". The general introduction and preliminaries of that first report are still valid for this document. The notions and conceptual choices explained in the morphological layer report are not specified again here, but they are sometimes referred to.

# 1: Generic model, "theory-welcoming" model

## 1.1: Lexicographic framework and recording formalism

When constructing a lexicon, one task is clearly identified, that of the lexicographer who records the relevant information for each entry of the lexicon.

For each entry, the lexicographer systematically carries out a certain number of tests, applying a heuristics, records the characteristics drawn from the criteria given by a linguistic theory, takes decisions on the acceptability of the patterns found, and translates the relevant information into a descriptive formalism. It goes without saying that this work cannot be carried out without having previously defined:

- on the one hand, the linguistic theory that defines notions, criteria grids determining the description of what can be observed, among them:

  - the definition of the lexical unit,

  - the identification, structuring and boundaries of the phrase,

  - the attachment of the prepositional phrase,

  - the decision taken on the notions of inner complement and modifier, that can be considered as notions to be differentiated or not. If the decision taken is to distinguish them, then defining criteria have to be specified;

- on the other hand, a recording formalism, i.e. a formal language for coding information, that determines the way information will be represented (definition of formal objects to be handled, descriptive vocabulary, rules to handle and operate on formal objects, constraints of integrity and consistency).

For "paper dictionaries", editorial manuals (usually confidential in nature) fulfil both functions and, by means of instructions, give the sequence order and the content of tests to be carried out, as well as the coding modes while respecting the selected theoretical frame (which gives the definition of notions).

Coding formalism is limited to the requirements of the following rule: traditional lexicography only resorts to formalism in order to structure texts (issue printing) and to fulfil a person's needs of reading. The situation is different for machine readable dictionaries: a human reader will easily deal with implicit references that he can easily clear up thanks to his knowledge of the world or his more or less conscious knowledge of language, and structure of lexical fields. Machine readable dictionaries can only contain explicit information recorded in a non-ambiguous formalism. In the case of an application-driven dictionary, the application determines the descriptive formalism, whereas the reference school determines the selected linguistic criteria.

Irrespective of the type of dictionary concerned, it follows that three skills (separate, yet which may be drawn simultaneously) are always required: that of the linguist, that of the "formalizer" and that of the lexicographer. A dictionary, either paper or electronic, will be the product of these three skills.

## 1.2: A generic "theory-welcoming" model

The purpose of GENELEX is to represent the various existing descriptions, taking into account their dependency on a given theoretical model, the degree of detail desired and the discriminating criteria selected by the lexicographer, irrespective of his school. This will have consequences on:

- the morphological level,

- the syntactic behavior of lexical units,

- the semantic level,

- the interaction among the above-mentioned levels.

This various descriptions will appear, either in several dictionaries, or in a single dictionary obtained in merging the others.

Actually, it is important to distinguish between a "theory-welcoming" descriptive formalism and a dictionary that uses formalism with reference to a given theory. The claim to genericity in dictionaries can only apply to descriptive formalism, considered as a bridge among different linguistic theories.

The GENELEX model is such a descriptive formalism; as such, and only as such, can it claim genericity.

A descriptive formalism can only be generic if it permits to record linguistic facts drawn from different theories.

## 1.3: Expressing different descriptions using the formalism of GENELEX

Let us consider different source dictionaries, as different in nature as in function such as DNT

(Dictionnaire de Notre Temps - Hachette) and a Ladl table. We want to translate the data contained in each dictionary into the formalism of GENELEX.

To facilitate the reading of the following examples, we will first give a brief description of the formalism adopted in GENELEX. At the level of the syntactic layer, the entry is the Syntactic Unit (referred to as **Usyn**). A Usyn represents one of the syntactic behaviors of the described word. It contains a "base description" (referred to as **db**). This db gathers the "base construction" (referred to as **cb**) and the specificities of the entry in this particular construction (borne by **Self**). A construction is a list of **Positions** that can be instantiated by **Phrases**. We can consider that the unit itself (described by Self) is or is not part of the construction, which is expressed by the indication or not of the insertion point of Self in the construction. For convenience purposes, the different positions are differentiated in the examples by an order number (P0, P1, P2, etc.), except for the position that realizes itself as Self.

**DNT format (Dictionnaire Hachette de Notre Temps)**

*aimer* V. tr. dir.

(to love)

We deduce from the transitive label that the entry *aimer* has the following syntactic description:

**GENELEX format**

*aimer*

```
db : SELF cb

SELF : IntervConst : V

cb : P0 SELF (P1)

P0 : SN

P1 : SN
```

**Ladl format**

*aimer* T7

We deduce from the entry criteria of table 7 that the entry *aimer* has the following syntactic description:

**GENELEX format**

*aimer*

```
db : SELF cb

SELF : IntervConst : V

cb : P0 SELF (P1)
```

```
P0 : SN

P1 : SN

P[SsCatSynt:COMPLETIVE]

[Mode:SUBJONCTIF]
```

# 1.4: Consistent merging of different descriptions

We must distinguish different cases calling for different solutions.

## 1.4.1: Complementary linguistic facts

The data recorded in the dictionaries concerned are different but complementary. Merging them would suffice to integrate them.

**DNT format (Dictionnaire Hachette de Notre Temps)**

*aimer* V. tr. dir.

**Ladl format**

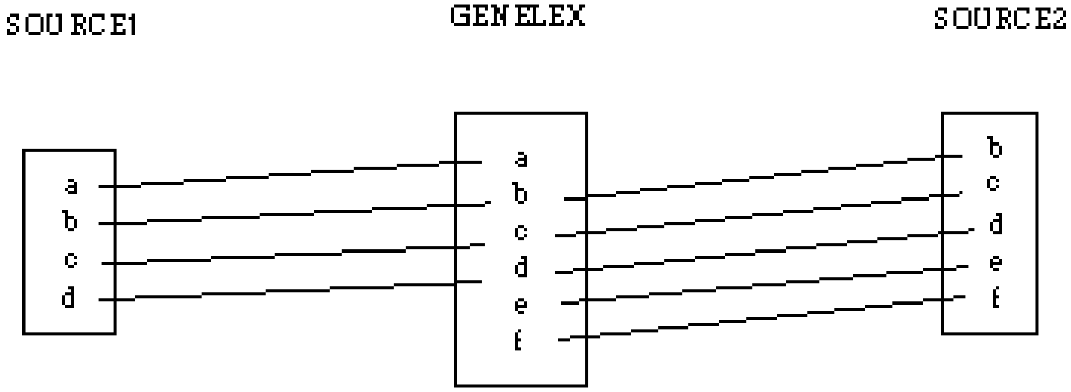
*aimer* T7

**GENELEX format**

*aimer*

```
db : SELF cb

SELF : IntervConst : V

cb : P0 SELF (P1)

P0 : SN

P1 : SN

P[SsCatSynt:COMPLETIVE]

[Mode:SUBJONCTIF]
```

A first dictionary indicates that a verb is transitive or intransitive, and another dictionary specifies which constructions are governed by this verb. In that case, the information given by the first dictionary can be complemented by the information provided by the second dictionary without any conflict. Therefore:

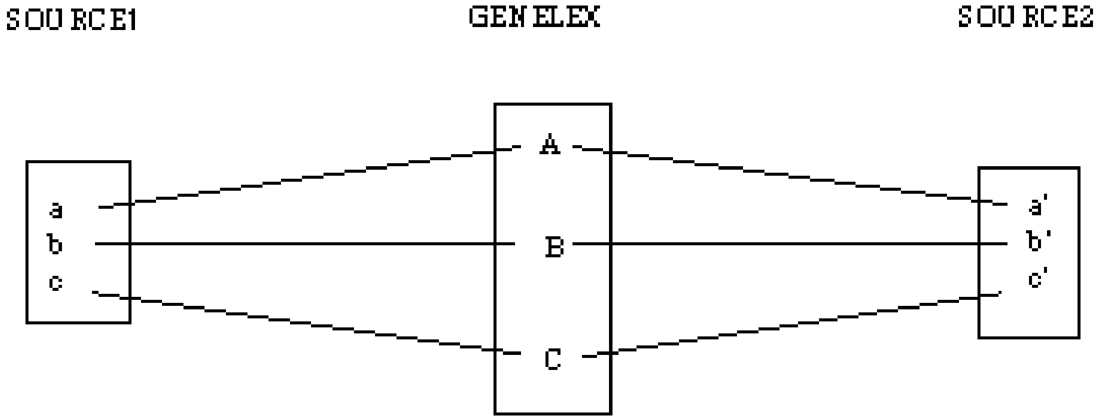


## 1.4.2: Competitive but not contradictory linguistic facts

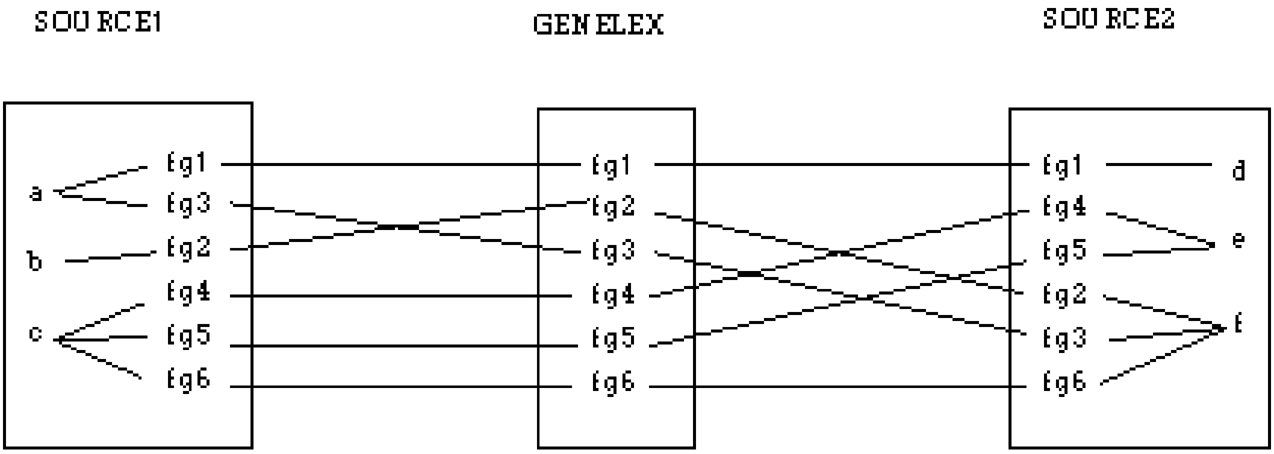
Dictionaries treat similar facts in different terms. The different terminologies express theoretical divisions depending on the way linguistic objects are considered, and the types of definitions allowed, rather than on the identification of the objects themselves.

These differences can actually be resolved if we succeed in:

a) defining a metalanguage to describe all phenomena, i.e. finding a system of equivalency between notions.



b) identifying granular facts constituting notions, irrespective of their original gathering, i.e. finding a system of equivalency between the elements forming notions.



All available data are processed in order to extract the "elementary pieces of information" (constituent granular facts) that are first laid down flat, then expressed into the generic format.

**Ladl format**

*aimer*

Table 7 : aimer ˆ Vinf

Table 32h : qqun aimer qqun

Table 12, etc...

sujet Nhum +

sujet V1 W +

Compl complŽtive pc z +

Compl inf V0 W +

Compl Pronom ppv le +

Compl le fait que P +

Compl Nhum +

Compl N-hum +

**IBM format**

*aimer*

(VERB IF TRAN (INFV TR) (COPLOBJ (A P)) AUXA

(QUEOBJ SUBJ) INFA INFD)

**GENELEX format**

*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:ˆ][Mode:INFINITIF]

P[SsCatSynt:COMPLETIVE]

[Mode:SUBJONCTIF]

PRO[Lex:le]

PRO[Lex:quelqu'un]

PRO[Lex:quelque chose]
```

## 1.4.3: Competitive and contradictory linguistic facts

In some cases, and if there is no global solution, theoretical conflicts cannot be solved, and the linguistic data drawn from a theory are clearly in conflict with others. Except if coding inconsistencies are tolerated (what would then be the reliability of a dictionary?), it is clear that contradictory facts cannot co-exist in a same dictionary.

Thus, for a neutral verb,

*Ex : Jean* ***casse*** *la branche* (Jean broke the branch)

*la branche* ***casse*** (the branch broke)

it will be impossible to declare at the same time that *la branche casse* corresponds both to the base description and to a transformed description of the same Usyn "casser".

*Ex : (1) Jean casse la branche*

*->(2) la branche casse*

(1) is one of the possible actualizations of the base description (db) of "casser".

Its transformed may be an actualization (2) of its transformed description (dt):

If we consider that we have two distinct syntactic behaviors at hand:

*(1) Jean casse la branche*

is one of the possible actualizations of a db of "casser".

*la branche casse*

is one of the possible actualizations of another db of "casser".

Similarly, for a verb of movement, we cannot record in a same dictionary that such-and-such adjunct of position is both an inner complement and a modifier.

*Ex : A Paris, Jean* ***a marchŽ*** *de la place Vend™me jusqu'ˆ la DŽfense*

(In Paris, Jean walked from Place Vend™me to La DŽfense)

Depending on the theoretical option selected, adjuncts of position are treated differently:

- exclusively as inner complements
- exclusively as modifiers
- depending on verbs

-either as inner complements (for instance, verbs of movement)

-or as modifiers (other verbs).

It will be impossible to merge these dictionaries, except if we give priority to a certain viewpoint over the others. In that case, to guarantee genericity, the selected theoretical option will be the one that induces the least loss of information and permits, by calculation, to extract an application-driven dictionary in another theory. Even then, in all cases, one still has to prove that such a theory exists.

## 2: Coding homogeneity

The objectives of consistency, thus (re)usability, can only be met if the choices concerning the descriptions of linguistic phenomena are specified: these choices are not only theoretical, but they also concern the mode for coding into the formalism proposed by GENELEX. As an example, one can quote the necessity to globally -and not on a case-to-case basis - specify the coding levels of the structuring of phrases described by lists of positions, or transformations (between Usyns, descriptions, phrases occupying positions).

Besides, the lexicographic strategy will have to specify the criteria necessary to determine the data that need to be coded in the lexicon.

It is also vital to apply a single strategy, in order to guarantee consistency between the different layers (morphological, syntactic and semantic).

What clearly emerges from the preceding is that one step cannot be ignored before initiating any lexicographic task when implementing the GENELEX model: the definition of a coding strategy.

# 3: Lexicon - Grammar articulation

The lexicon and the grammar are the two guardians of the descriptive knowledge of language. Both entities are expected to be complementary in an exhaustive description (and usable in automatic processing, in our case). This is not without consequences on the global approach required for modelling the lexicon.

The option selected in GENELEX is to code any necessary information in a format as "neutral" as possible, irrespective of the applications planned. In other words, GENELEX, a lexical information storage base, makes it possible to extract dictionaries corresponding to the above-mentioned choices and to pursue the objective to be able to represent as much information as possible.

Let us remind the traditional situation: both a lexicon and a grammar are available.

## 3.1: Compatibility between the lexicon and the grammar

## 3.1.1: Theoretical approach and formalism

Let us consider a theoretical viewpoint determining splitting into linguistic units (constructions, attachment of phrases, delimitation of the boundaries of phrases, etc.). This viewpoint determines the linguistic description done by the grammar as much as the syntactic information contained in the lexicon.

Here are the two cases when a lexicon and a grammar are available:

1. the lexicon and the grammar have been built according to the same theoretical approach

2. the lexicon and the grammar have been built according to different

theories, which means that some linguistic phenomena are described differently, or identified in one and not in the other.

Irrespective of the theoretical choices supposed to guide the coding of information, both the description of the lexicon and that of the grammar presuppose choosing a formalism of description, with its own units, its vocabulary and its way of structuring information, and of course powerful enough to convey the selected theory.

Once again, when both a lexicon and a grammar are available, there are two cases:

1. the grammar and the lexicon are coded according to the same formalism.

2. the grammar and the lexicon are coded according to two different formalisms.

Compatibility between the lexicon and the grammar can also be considered under both angles, as indicated in the following table.

| | *IDENTICAL FORMALISMS* | *DIFFERENT FORMALISMS* |
|---|---|---|
| *IDENTICAL THEORETICAL*<br>*APPROACHES* | **CASE No. 1**<br>**Optimum situation:**<br>"Communication" between the lexicon and the grammar is immediate.<br>There is only one concern: the same degree of description is desirable.<br>Note: a lexicon with a too high degree of description will only be underexploited. | **CASE No. 2**<br>**Favorable situation:**<br>"Communication" between the lexicon and the grammar is possible, provided they are converted into a single formalism.<br>This translation is always possible because sharing the linguistic approach guarantees the presence of the same linguistic units. |

| | | |
|---|---|---|
| *DIFFERENT THEORETICAL APPROACHES* | **CASE No. 3**<br>**Rather favorable situation:**<br>An "application-driven" dictionary compatible with the grammar must be extracted from the lexicon.<br>The strategy is the same as that for merging dictionaries (previously described).<br>**Risk to loose information**<br>Note: the convergence or divergence of different theoretical approaches conditions the quality of the resulting information. | **CASE No. 4**<br>**Hard situation:**<br>The existence of a "bridge" between the lexicon and the grammar is not guaranteed.<br>The lexicon has to be translated into the same formalism as that used by the grammar (without loosing too much information, if possible)<br>This situation is the same as case no. 3.<br>**Risk to loose much information** |

In that case also, the choice made in GENELEX tends to the most welcoming "neutrality " to be ever elaborated. The formalism adopted has to be converted into the application formalism.

> **Note:** considering these problems, certain theories have therefore decided to adopt not only the same theoretical approach but also the same formalism to describe the lexicon and the grammar. The syntactic data associated with the lexicon have then a fine granularity that can even describe the possible combinations of lexical units with one another, as in lexical functional grammars (LFGs).

## 3.1.2: Minimum compatibility required by the GENELEX model

To ensure minimum compatibility between the lexicon and the grammars implemented to use it, GENELEX requires, for the coding of syntactic information in French, to use phrases which, we hope, will be unanimously accepted in the scientific community. These phrases will be identified by labels; a list of labels (SN SP, SADV,...) that can be extended, is proposed by the model of the syntactic layer. Actually this list gathers the labels most commonly used to designate the syntagmatic structures obtained when actualizing complementation patterns of units and often called "major".

This (minimum) list of syntagmatic labels must minimally guarantee a theoretical "bridge", on the one hand between different theories implemented in the GENELEX model, on the other hand with different grammars. One can say that it is a hard core upon which one can base any theory used as a basis for defining the criteria necessary to code a dictionary using the GENELEX model or to construct a grammar that will use the information drawn from such a dictionary.

The theoretical constraint induced by this list is offset by the possibility to refine the description of phrases ad libitum:

- thanks to the combination of restricting features that may be:

  - lexical

  - morphological

  - morpho-syntactic

  - syntactic

  - syntactic-semantic or semantic,

- thanks to the eventual expression of constraints for rewriting phrases,

- thanks to the freedom of linear or hierarchic structuring of phrases between one another.

It is thus possible to express, down to the smallest detail, the characteristics of realizations allowed for the linguistic elements used in the description of the described unit.

This set of tools available allows descriptions with varying depth, depending on:

1. the theoretical choice,

2. the intended degree of description,

3. the progression of works,

hence different modes of customization of the model by users.

Besides, users may enrich the proposed list with additional more detailed syntagmatic labels. The only drawback is that they will lose in genericity.

## 3.2: Cross-references between the lexicon and the grammar

A lexicon that describes the syntagmatic behaviors of lexical entries is a complement to the general syntactic information contained in a grammar.

These objects are of necessity complementary to reach a satisfactory description of language and they function as communicating vessels:

- the richer and the more detailed the grammar, the poorer and more concise the lexicon may be.

- the richer the lexicon, the more reduced the grammar.

The current trend is to grant an increasing role to the lexicon in the syntax description, and besides, the "lexicalist" attitude, to an extreme degree, consists in coding all syntactic data in the lexicon and let the grammar only dictate the rules of allowed combinations. For information purposes, we will indicate the tree adjoining grammars (TAGs) and categorial grammars.

Therefore this complementarity lexicon-grammar implies that some data are recorded either in the lexicon, or in the grammar, as one wishes.

In the model of the syntactic layer of GENELEX, we want to code in the lexicon the syntactic information specific to an entry that could not be sufficiently predicted by its belonging to a morphological category and sub-category. There is no indication on the required level of precision.

The principle is the following: we want the finest pieces of information to be recorded, even if they are likely to be "deleted" when they are extracted to build up a dictionary for an application requiring less precise lexical information.

In particular, the grammar is implicitly in charge of describing a certain number of linguistic phenomena:

- inflection and agreement phenomena
- tense agreement
- the disruption in linearity likely to induce a gap between the canonical order recorded in the dictionary and certain surface realizations

  This disruption can be connected to:

  - the style,
  - the emphatic or extraposition structures,
  - the interrogative or negative structures,
  - the permutations induced by heavy NP phenomena,
  - the insertion of modifiers,
  - the co-ordination phenomena,

  ...

...

Implicit cross-references from the lexicon described in the model of GENELEX to the grammar do not presuppose the way the grammar will convey these phenomena. From that viewpoint, one can say that the grammar is transparent.

## 4: Syntax autonomy

The model of the syntactic layer of GENELEX makes it possible to take into account right from the syntactic phase elements of semantics that have consequences on the syntactic level and allow to highlight different syntactic behaviors. In consequence, these semantic data lead to the splitting of syntactic units that syntax alone could not justify without implicitly referring to semantics: thus the identification of two verbs *voler* is not only based upon the identification of two formal structures.

It is therefore to be noted that the introduction of semantic information right from the syntactic phase allows to identify different constructions associated with different meanings, hence it allows to disambiguate without requiring finer information from the semantic level. Actually the semantics we have chosen to be able to make **explicitly** appear is most often used **implicitly** in the existing syntactic descriptions. It is restricted to thematic roles and denotative conditions.

Those who prefer a syntax completely independent from semantics are obviously free not to use this type of information.

# B - Analysis of simple syntactic units

## I: Articulation with the Morphological Layer

The morphological layer of the model describes lexical units from a morphological viewpoint. For each morphological unit (**Um**), one has to record information concerning - among other description elements - the written form and spelling variants of a same lexical unit, its form of inflection, its grammatical category, etc. Therefore the morphological layer allows to gather on a single morphological unit the pieces of information shared by lexical units on this level, irrespective of differing syntactic or semantic behaviors that could be associated with them.

To be complete, the description of lexical units requires the representation of other types of information concerning the particular syntactic behavior of the entry. The objective is to record as accurately as necessary the specific characteristics that distinguish it from the general behavior associated with its grammatical category recorded in the morphological layer. For instance, we know that verbs have one subject and 0 to n complements. We will have to specify the number of complements and their nature (refer to the rules of sub-categorization in HPSGs). The syntactic layer of GENELEX is dedicated to the recording of this type of information.

If a morphological unit (Um) has one and only one syntactic category at the level of the morphological layer (grammatical category), it may have **one** or **several** "syntactic behaviors".

*Ex : L’homme* ***arrive*** *ˆ Paris.* verb of movement

(The man arrived in Paris)

*L’homme* ***arrive*** *ˆ comprendre.* modal verb

(The man succeeded in understanding)

*L'homme* ***vole*** *une pomme.* transitive verb

(The man stole an apple)

*L'oiseau* ***vole*** *rapidement.* intransitive verb

(The bird flew quickly)

Syntactic units (**Usyn**) are used to describe these syntactic behaviors (attested, insofar as possible), that are also usually called turns of phrase or uses, whatever the category of the entry.

The model presented concerns simple words and compound lexical units. Besides, the formalization of simple words obviously applies to the representation of the external syntax of compound words.

# II: Syntactic Units (Usyns)

# 1: General definition

Syntactic units are access points to the syntactic layer.

They can be simple or compound. In this part, we will only consider simple syntactic units, since compound units will be dealt with in part C of this document.

A syntactic unit (**Usyn**) describes one - and only one - syntactic behavior of a morphological unit (**Um**).

Each morphological unit is therefore in relation with at least one syntactic unit. If a Um has several "syntactic behaviors", then it is in relation with several Usyns. In the Entity-Relation model, this point is expressed by cardinalities. Therefore it is not possible to draw up a finite list of syntactic units. Actually they directly depend on the Ums instantiated in the dictionary.

A simple Usyn corresponds to one and only one Um.

If the entry belongs to a major category, then the Usyn gives a minimum description of its complementation pattern (complementation of the Verb, the Noun, the Adjective, the Adverb). The Usyn of an entry may be considered as an abstraction on the phrase whose entry is the head. What we mean here is that at a "meta" level, the Usyn contains all the pieces of information that permit to describe its behavior in a surface actualization.

The syntactic unit permits to represent the syntactic behavior of lexical units according to two very different viewpoints:

• the "**atomist**" viewpoint, focused on the properties of the lexical unit irrespective of the grammar chosen, is intended to describe the elements that directly depend on the described unit. This information concerns deep syntax, since surface syntax must be dealt with by grammars;

• the "**syntactist**" viewpoint, encountered in "lexicalist" approaches aiming at recording all or part of the grammar in the lexicon (categorial grammars, TAGs, etc.) by the combination of phrases, allows, at the level of the described Usyn, reference to the syntactic structures in which it fits. The lexicographer will have to define his strategy concerning the grammar, to ensure homogeneous and consistent coding of the lexicon.

In a concern for homogeneity and to ensure consistency checks on the lexicographic base, it is recommended to avoid having both interpretations of the model in a same dictionary.

## 2: Atomist viewpoint

The atomist viewpoint gives priority to the saturation of the described unit: it is assumed that the unit emits saturation calls, i.e. it bears a certain number of "holes", **the positions to be saturated**, and that it expresses restrictions on possible occupants without indicating anything on the surface syntagmatic realizations, since the organization of phrases is left to the grammar.

Positions are therefore pieces of information emitted by the unit towards the outside. Positions will be occupied (actualized) by the realizations of the called Usyns. The called behavior is the minor functioning of the units that remain free to behave as callers (being called, they may also emit calls themselves). If the calling unit imposes restrictions on the units it calls, these restrictions are recorded when the realizations of its positions are described: in the formalism, position occupants are phrases that can be restricted by features. In the atomist viewpoint, phrases as described by the formalism are labels allowing to name allowed Usyn actualizations, they are not re-written, unlike in the syntactist viewpoint.

*Ex : un petit **groupe** de personnes* (a small group of persons)

```
cb : P0 (P1) (P2)

P0 : DET

P1 : SADJ

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

[SsCatSynt:DET_VIDE]
```

This example shows how the basic construction of a syntactic unit describes a major functioning, that of the unit as a caller. Major behaviors of entries will be the only behaviors described by a construction in this interpretation of the model. We also note that the entry insertion point, SELF, is not indicated: the cb is not interpreted as a phrase description, but as the sum of information emitted by the unit described.

In the atomist vision of the model, in the construction one does not go beyond the level of the phrase whose head would be the entry. The relations between a syntactic unit and the other syntactic units, in a frame that goes beyond the units it "governs" as a caller may be specified in Self using a set of sub-categorizations, feature descriptions and details on its possible functions. Thus, if we want to specify the possible uses of adjectives as subject or object predicative, right- or left-positioned attributive, we will have to indicate the following functions:

EPITHETE_GAUCHE, EPITHETE_DROIT, ATTRIBUT_SUJET, ATTRIBUT_OBJET.

In the examples given in this document, the line `[cb : P0 SELF (P1)...]` is used to show synthetically the information likely to be borne by the construction and relating to its actualization: option, constraints on the surface organization. The insertion point of Self will therefore systematically be indicated. However, this information will never be relevant for the atomist vision of the model; we have adopted this convention here in a concern for presentation homogeneity.

## 3: Syntactist viewpoint

The syntactist viewpoint gives priority to the inscription of the unit in a context of occurrence; the position occupied by Self (the unit described) appears in the base construction; this construction is the expression of the syntactic structure in which the described unit fits. The syntactic unit is thus presented as fitting in a syntactic tree. This tree represents a more or less extended syntactic context, depending on the specifications and constraints to be expressed.

Usyns do not exclusively describe major behaviors: Self may be in a position whose function is "TETE" (head) (major behavior) or not (minor behavior). For an entry belonging to a minor category, the Usyn describes a context of occurrence.

In the syntactist vision of the model, it is thus possible, to describe a context of occurrence, to go beyond the frame of the phrase whose entry is the head (SN for N, SADJ for ADJ, P or SV for V).

Therefore, if we want to specify the possible uses of adjectives as subject or object predicative or right- or left-positioned attributive, it will be possible to extend the description frame to the SN, even to the sentence, which will allow to express constraints on the subject from a predicative adjective:

*Ex: Il est* ***intŽressant*** *de remarquer cela.* (It is interesting to note that)

The description of a phrase through rewriting is expressed by a list of positions, one of these positions may have the "TETE" function. Positions describing the rewriting of a phrase can themselves be occupied by a rewritten phrase (recursivity) so as to introduce a tree structure.

*Ex : un petit* ***groupe*** *de personnes (*a small group of persons)

*le* ***groupe*** *des personnes qui...* (the group of persons who...)

```
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: Criteria for splitting Usyns

The formal criterion retained for splitting syntactic units is the fact that a syntactic unit has one and only one base description.

Linguistic and lexicographic criteria leading to the elaboration of syntactic descriptions are under the lexicographer's responsibility. However we will give here certain recommendations to make good use of the model.

We have first to adopt a description that will permit to obtain a correct coverage of the syntactic behaviors while trying to avoid redundancy or too powerful gatherings at best. The following examples are a good illustration of this point:

*Ex : (1) Pierre* ***arrive*** *ˆ partir* (Pierre succeeded in leaving)

*(2) a. Pierre* ***arrive*** *ˆ Paris* (Pierre arrived in Paris)

*(2) b. Pierre* ***arrive*** (Pierre arrived)

*(3) a.* ***tarte*** *aux abricots / ˆ la banane* (apricot/banana tart)

*(3) b.* ***tarte*** (tart)

• avoid redundancy at best (which is the sign of a rather unsatisfactory modelling, since what belongs to a same concept has not been correctly gathered) that induces the creation of a multiplicity of useless syntactic units.

Ex : we have chosen to gather:

- (2) a. and (2) b, for *arriver* verb of movement

- (3) a. and (3) b, for *tarte* and *tarte ˆ*...

• avoid the opposite case, since a too powerful model does not allow to manage efficiently the restrictions on the realizations specific to a unit.

Ex : we have chosen to dissociate the two *arriver* ˆ :

- use of a modal verb, e.g. (1)

- use of the verb of movement , e.g. (2).

To shematize, the purpose is thus to determine which type of representation of the syntactic information will have to be retained to avoid both shortcomings of a description such as "this verb may be complemented" and a description that would lead to an inventory as exhaustive as possible of encountered constructions (even simply conceivable), since certain local observations can be formulated by syntactic rules. To the extreme degree, we could actually think of associating a unit with a specific construction (construction having here the meaning of "sentence") found in a corpus, for instance. The description of this construction could be in itself a construction that could not be connected with a described syntactic behavior. This construction would then imply the creation of a distinct Usyn.

# III: Descriptions

A Usyn is defined by a **Base description** and 0 to n **Transformed descriptions**.

A Description is defined by Self and a Construction.

It is also possible to share the constructions independently from the constraints on Self: for instance, some verbs share the same Construction, whereas they differ in the auxiliary and/or the preverbal particle:

*Ex : partir* (to leave, to go away)

*s'en aller* (to leave, to go away)

A description is not intrinsically basic or transformed, but it fulfils this role for a given Usyn. A same description may also play the role of a base description for a Usyn and that of a transformed description for another.

# IV: Constructions

A **Construction** describes the syntactic context called and/or restricted by the described entry. In other words and for verbs, a construction describes what is usually called a "complementation pattern".

A construction is defined by a canonically ordered list of positions, their option, interdependency and surface organization (described in section VI-4). It can also bear a non-terminal syntagmatic label and restricting features.

To take up again our distinction between both interpretations of the model:

- in the **atomist** viewpoint, the construction is only the structure gathering the descriptors of complementation patterns. As such, it is not considered as a phrase and is not associated with any syntagmatic label. Besides, the construction will only describe major behaviors.

- In the **syntactist** viewpoint, a construction implicitely describes a phrase; hence a syntagmatic label can be associated with it. Depending on the case, the described context may be at the level of the sentence, the noun or prepositional phrase, etc. Besides, since the construction describes a context of occurrence, it does not exclusively describe a major behavior: it can also give information on minor behaviors of units described as major (e.g. adjectives) or on behaviors of minor units.

# V: Self

# 1: General definition

The element **Self** allows to describe the properties of the entry for a Usyn

- inside its construction
- outside its construction.

For compound units, Self also allows to describe the internal structure of the compound unit.

# 2: Self intervening in the construction of the entry

Given a syntactic construction, Self allows to express all the characteristics of the entry for this construction, as a caller or as a called unit:

- **category of the entry:** it may be different from its morphological category (grammatical category of the Um), which allows to indicate a difference between a morphological category and a functional category (category indicated by the syntagmatic label of the Usyn). It is thus possible to describe the adjectival behavior of the name "abricot", for instance.

*Ex : abricot* (apricot)

```
Self : IntervConst : ADJ
```

- **conjugation auxiliary** if it is a verb

*Ex : tomber* (to fall)

```
Self : IntervConst : V[Aux:ETRE]
```

- **morphological restrictions:**

*Ex : lustres*

```
Self : IntervConst : N[Nombre:PLURIEL]
```

- **preverbal particle** for uses of "true pronominal verbs"

*Ex : s'en aller (to go away)*

```
Self : IntervConst : V[Pronominal:SE_EN]
```

These pairs (category - restricting features) correspond formally to phrases as defined in the formalism of the model. Consequently, the characteristics of Self as Intervening in the Construction will be expressed by a list of terminal phrases borne by an element called **IntervConst**. It is thus possible to gather in a single description uses of verbs having the same construction for a varying Self. Here are some examples:

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

(We have agreed upon a meeting)

(ETRE or AVOIR auxiliary)

```
Self : IntervConst : V[Aux:ETRE]

V[Aux:AVOIR]
```

*Ex : **(se) moquer*** (to laugh at)

```
Self : IntervConst : V[Pronominal:SE]

V
```

- **function and thematic roles:** they have a meaning in the syntactist vision where constructions fall into the category of phrases in which Self functions:
  - either exclusively as a head when it is a caller (behavior of "major" category)
  - or as a *N*-level rewrite element receiving a function when it is called (behavior of "minor" category), which does not exclude the description of the phrase of which it is the head.

*Ex : Il est **intŽressant** de remarquer cela*

(It is interesting to note that)

```
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 called by an element external to the construction of the entry

Besides, it is possible to indicate the behavior(s) of Self as called by an element which is not described in the construction.

In the **atomist** viewpoint, where only caller behaviors are described (by a construction), it is also possible to specify that such-and-such construction described by an adjective refers to its use as attributive or predicative. These behaviors as called units are described in a list of **ComportAppele**, i.e. elements equivalent to phrases, made of a pair (category - restricting features) to which a function and one or several thematic roles may add, if necessary.

In the **syntactist** viewpoint, on the contrary, all behaviors as called units can be entirely described by the construction. However one reserves the right to describe them differently, namely when the heaviness of a construction to be implemented does not seem justified. It will be possible to describe behaviors of Self as attributive or predicative by means of ComportAppele, if one does not want to give further details on these contexts.

# VI: Positions

## 1: General approach and justification

### 1.1: Example of verb complementation

To present the elements that have governed the elaboration of the notion of **Position**, we will use the category best described in current syntax studies: the verb. It is to be noted that verb complementation patterns are described in very different terms, depending on the theory applied. Let us take as examples:

- subject, direct object, indirect object, predicative, etc.: traditional grammars, functionalism and LFGs

- N0 (noun paradigm), N1, Prep N2: distributionalism and transformational grammar

- A0 (actant), A1, A2: Tesni•re

- Arg0 (argument), Arg1, Arg2: predicative logic, Montague

- thematic role: generative grammar, Jackendoff and Fillmore

### 1.2: Emergence of the notion of Position

We call position a meta-notion that subsumes all of these, an abstraction that can be formalized through the restructuring of the elementary information irrespective of the way it is organized in the original notion.

> For a verb, a position is a paradigm that gathers the various syntactic realizations of subject or verb complement and is part of the maximal definition of verb valence. (GENELEX June 91)

J.-C. Milner [1989] explicitely introduces the notion of position in a more complex system that also includes spots and places.

There is not a complete overlap between our definition of position and the notion introduced by Milner: we maintain the distinction between position and position occupants. Like him, we think that:

> "The categorial belonging of a term X and the categorial label of the position Y occupied by X [are] independent by right" [Milner, 89, p.370].

Therefore we consider it justified to assign different labels to both types of "categories", whereas in Milner's theory, positions bear the label of their typical syntactic realization. Treating them as homonyms would only create confusion between nature and function.

In the GENELEX format, positions are defined (among others) by the set of phrases (syntagmatic labels and/or rewritten phrases) that can instantiate them: it is not necessary to mark out a "leader" (necessary to label a position in Milner's theory) in this set of phrases.

Actually, it is not always easy to extract a "leader" from all syntactic paradigms. It is commonly acknowledged that the Noun Phrase (SN) is the typical realization of a position with a subject function, though various phrases occupying a position with a subject function exist:

```
SN

P[SsCatSynt:COMPLETIVE]

P[Mode:INFINITIF]

...
```

*Ex :* ***Cette dŽcision*** *regarde Marie.*

(This decision concerns Marie)

***Qu'il prenne cette dŽcision*** *regarde Marie.*

(The fact that he takes this decision concerns Marie)

***Prendre cette dŽcision*** *regarde Marie.*

(Taking this decision concerns Marie)

...

On the contrary, it is sometimes impossible (or arbitrary) to be so positive on a position with an object function.

Let us mention speech verbs in which noun clauses, infinitive clauses and noun phrases or verb phrases are in the same paradigm.

*Ex : Jean* ***pense*** *que Marie est partie*

(Jean thinks that Marie has gone)

*Jean* ***pense*** *ˆ Žcrire*

(Jean is thinking of writing)

*Jean* ***pense*** *Žcrire ˆ Marie*

(Jean is thinking of writing to Marie)

*Jean* ***pense*** *au dŽpart de Marie*

(Jean is thinking about Marie's departure)

*Jean* ***dit*** *que Marie est partie*

(Jean says that Marie has gone)

*Jean* ***dit*** *partir avec Marie*

(Jean says he would leave with Marie)

*Jean* ***dit*** *de partir ˆ Marie*

(Jean tells Marie to leave)

*Jean* ***dit*** *des choses ˆ Marie*

(Jean tells things to Marie)

*Jean* ***recommande*** *d'•tre sage ˆ Marie*

(Jean advises Marie to be good)

*Jean* ***recommande*** *la sagesse ˆ Marie*

(Jean advises Marie to be good)

## 1.3: Case of left determination

This left determination refers for instance to the different determiners for nouns and to a position with a "subject" function for verbs. This position with a determiner or subject function can be put in the dependency of the described unit, the noun or the verb. As already seen, the subject may have several realizations in French.

However certain verbs exclude one or several of these realizations.

*Ex : Pierre* ***opine*** *de la t•te* (Pierre is nodding assent)

```
P0 : SN
```

`PRO[Lex:il][SsCatMorph:PERSONNEL_FORT]`

Certain verbs restrict one or several of these realizations.

*Ex : Il* ***pleut*** (It is raining)

(meteorological verb)

*Il* ***semble*** *qu'il neige* (It seems to be snowing)

(verb used as an utterance modalizer)

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

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

(The garden is swarming with ants )

(one of the uses of a symmetrical verb, the other being *le jardin grouille de fourmis*)

`P0 : SN[Nombre:PLURIEL]`

Having the subject depend on the verb allows to express these exclusions or restrictions on the subject. It is the way to give account for lexical selection phenomena.

For the same reasons, we have other determiners depend on the determined.

For instance, certain nouns require a definite determiner.

*Ex : la* ***plupart*** (most (of))

*le* ***palŽolithique*** (the Paleolithic)

# 2: Application to other categories

Considering the complementation pattern or the description of a context in terms of positions allows to extend the principle to any other categories.

Note: the line `[cb: P0 P1 ...]` is used to show synthetically the information likely to be borne by the construction and concerning its actualization: option, constraints on the surface organization. The systematic presence of Self, adopted in a concern for presentation homogeneity, is not relevant for the atomist vision of the model.

## 2.1: Nouns

It is possible to use describe nouns in this way. In litterature, the problem of noun complementation is most often restricted to the study of deverbal nominalizations. The positions observed then are very similar to the position of the verb:

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

(the dreadful destruction of the town by the enemies

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

P0 : DET

P1 : SADJ

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

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

But this type of description also applies to the other nouns. It is, for instance, possible to characterize quantifier nouns:

*Ex : un petit **sac** de clous* (a small bag of nails)

*un **sac** de farine* (a bag of wheat)

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

P0 : DET

P1 : SADJ

P2 : SP[Prep:de]
```

Certain noun determiners constrain a definite determiner.

*Ex : la **plupart** du temps* (most of the time)

```
cb : P0 SELF P1

P0 : DET[SsCatMorph:DEFINI]

P1 : SP[Prep:de]
```

Others constrain a plural complement.

*Ex : une Žnorme **meute** de loups* (A huge pack of wolves)

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

P0 : DET

P1 : SADJ
```

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

Position realizations permit to sub-categorize nouns at various degrees of detail.

## 2.2: Adjectives

The same system can be applied to adjectives. All adjectives that govern a complement will be separated from the others.

*Ex : tr•s* ***capricieux*** (very wayward)

```
cb : P0 SELF

P0 : SADV
```

*tout ˆ fait* ***apte*** *au travail* (quite capable of working)

```
cb : (P0) SELF (P1)

P0 : SADV

P1 : SP[Prep:ˆ]
```

Restrictions on phrases occupying a position (see paragraph on this point below) allow even more detailed subcategorizations.

*Ex :* ***content*** *de venir* (glad to come)

***content*** *de ses rŽsultats* (happy with his/her results)

```
cb : (P0) SELF (P1)

P0 : SADV

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

SP[Prep:de]
```

***difficiles*** *ˆ satisfaire (*difficult to satisfy)

```
cb : (P0) SELF (P1)

P0 : SADV

P1 : P[Prep:ˆ][Mode:INFINITIF]
```

# 2.3: Prepositions

Prepositions can also be defined using this model.

*Ex : ˆ Paris* (in Paris)

ˆ *ma m•re* (to my mother)

ˆ *boutons* (with buttons)

ˆ *faire des b•tises* (to play the fool)

```
cb : SELF P0

P0 : SN

P[Mode:INFINITIF]
```

# 2.4: Determiners

The construction given here represents a context of occurrence of a minor category in the syntactist vision of the model.

*Ex :* ***certaines*** *petites b•tises qui sont pardonnables*

(some little silly things that can be forgiven)

```
cb : SELF P0* P1 P2*

P0 : SADJ

P1 : N

P2 : SADJ

P[SsCatSynt:RELATIVE]

SP
```

In this example, it is to be noted that the entry is not the head of the phrase in which it fits, because it is a minor category. The same applies to conjunctions.

# 2.5: Adverbs

The scope of the adverb can be expressed by positions by different syntactic units.

Adverb emphasizing an adjective:

*Ex :* ***tr•s*** *grand* (very tall)

```
cb : SELF P0

P0 : SADJ
```

Adverb emphasizing an adverb:

*Ex :* ***tr•s*** *poliment* (very politely)

```
cb : SELF P0

P0 : SADV
```

# 3: Definition of the Position

Let us remind that on the base of positions, a same lexical unit may have one or several syntactic behaviors that give rise to one or several Usyns.

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

(The man stole an apple)

```
cb : P0 SELF (P1)
```

2. *L'oiseau* ***vole***. (intransitive)

(The bird flies)

```
cb : P0 SELF
```

*Ex :* 1. *Il* ***arrive*** *ˆ Paris*

(He arrived in Paris)

```
cb : P0 SELF (P1)

P1 : SP[SsCatSynt:LIEU]
```

P1 is defined by a prepositional phrase introduced by a sub-set of prepositions belonging to a group of "prepositions of place". (Besides, it is optional).

2. *Il* ***arrive*** *ˆ partir* (modal)

(He managed to leave)

```
cb : P0 SELF P1

P1 : SP[Prep:ˆ]

P[Prep:ˆ][Mode:INFINITIVE]
```

P1 is defined by a prepositional phrase introduced by the preposition ˆ. (Besides, it is mandatory).

## 3.1: Formal definition

On the operation level, the position is an element entering into the definition of a construction.

A Position is defined by a set of three elements: distribution ; function ; thematic roles.

Positions are noted P, P being the abbreviation of Position. To differentiate them easily, they are noted Pi, i being an integer ranging from 0 to n, that indicates the rank of the position according to the others in the canonical order established for a construction.

This notation is only used in the text of option, interdependency and constructions. The other elements of the model that manipulate positions will identify them only by the rank i.

A **Distribution** on a position is the set of Phrases that can instantiate this position. Distribution is therefore to be understood in the linguistic meaning of substitution paradigm.

A position has 0 or one **Function**, 0, one or several Thematic Roles (referred to as **RoleTh**).

- **Documentation of Distribution**

As distribution plays an essential role in the definition of the position, it will always be documented. If we want to indicate the presence of an "empty position", i.e. a position not realized in surface, we use the value "e" (empty element) of the Phrase. The category "e" allows to record traces for adherents to generative grammar and to consider them as "phantom" phrases (used in particular to represent the subjects of infinitive clauses bearing constraints of coreference) with which one can associate the necessary restrictions.

- **Documentation of Function**

A function may not be documented for several reasons:

- no particular designation in use on the functional level

*Ex : Un mammif•re qui allaite ses petits*

(a mammal that suckles its young)

*Un fumeur qui ne supporte pas la fumŽe*

(a smoker who cannot bear smoke)

if we want to distinguish appositive relative clauses from restrictive relative clauses

- lack of information due to intentional or unintentional incompleteness.

- **Documentation on the Thematic Role**

A thematic role may not be documented for several reasons:

- no particular designation in use

- autonomous syntax

- intentional or unintentional incompleteness

**Note**: as the values provided for by the model may seem insufficient to the user, both as regards function and thematic role, the user may add his own values.

# 3.2: Positions and Distributions

# 3.2.1: Position Occupants

The list of phrases that can occupy a position is the distribution on this position.

In the **atomist** viewpoint, phrases are not rewritten, except when the conjugated set of syntagmatic labels and restricting features would be insufficient to express all the constraints.

In the **syntactist** viewpoint, phrases are either rewritten or not, depending on needs.

Besides, transformation links between phrases occupying a same position (called TransfSyntagme) may exist.

# 3.2.2: Sharing of Positions

Positions may be shared by different entries, whatever their rank in a construction.

It could be possible to propose, in a lexicographic station, a pre-defined list of positions that could be extended ad libitum.

Thus, based on the works relating to Ladl, we could propose lists of the following type for the different positions of a verb:

Realizations of **`P[SUJET]`** :

```
e (position not realized in surface)

SN

PRO

P[Mode:INFINITIF]

P[SsCatSynt:COMPLETIVE][Mode:SUBJONCTIVE]

etc.
```

Realizations of **`P[OBJET]`** :

```
SN

PRO

PRO[SsCatMorph:PERSONNEL_FAIBLE]

SP

SADJ

SADV

P[Mode:INFINITIF]

P[Prep:^][Mode:INFINITIF]

P[Prep:de][Mode:INFINITIF]

P[SsCatSynt:COMPLETIVE][Mode:INDICATIF]

P[SsCatSynt:COMPLETIVE][Mode:SUBJONCTIF]

P[Prep:^][SsCatSynt:COMPLETIVE]

P[Prep:de][SsCatSynt:COMPLETIVE]

P[Tournure:INTERROGATIVE]

etc.
```

# 3.3: Position and Functions

A function may be associated with any position: a function is a defining property of the position.

Constructions can therefore be differentiated by the functions associated with their positions.

*Ex : Jean* ***aime*** *Marie* (Jean loves Marie)

```
P0 SELF P1

P0[Fonction:SUJET]

P1[Fonction:OBJET_DIRECT]
```

Functions associated with positions are defined according to the head that can be:

- the calling unit, i.e. Self having a function of TETE (head), if the construction describes the major behavior of this unit;

- a Position of the same level with a TETE function if the construction describes the minor behavior of a unit.

It is to be noted that the function is defined for one position, and the nature of the phrases occupying this position is not always determining: actually a noun phrase and a prepositional phrase may co-exist in a same position. It is the case of verbs whose main verb complement is a position with a "place" function:

*Ex : Jean* ***fouille*** *ses poches* cf. [Boons Guillet Lecl•re, 1976]

*Jean* ***fouille*** *dans ses poches* (Jean is going through his pockets)

GENELEX proposes a basic list of functions; users can modify or extend it.

## 3.4: Positions and Thematic Roles

The usefulness to use thematic roles to describe positions will be proved by reducing it to the absurd.

*Ex : (1) Jacques* ***tra"ne*** *la valise.* cf. [G.B•s, 1991]

(Jacques is pulling/dragging the suitcase)

*(2) Jacques* ***tra"ne****.* (Jacques is lounging about)

*(3) La conversation* ***tra"ne****.* (The conversation is dragging on)

Case number 1:

We use the notion of paradigm in which different phrases may alternate.

We use option as a technical means to gather surface structures.

Consequently, for the verb tra"ner, we have the following structure: P0 SELF (P1). We do not give account of the fact that the transitive use selects the interpretation "tirer" (to pull) whereas the intransitive use selects the meanings "fl‰ner" (to hang about, to lounge about) and "s'Žterniser" (to take ages).

The dictionary does not remove the ambiguities it could remove right from the syntactic level: a syntactic parser using this dictionary would not be capable to select the meaning "tirer" (to pull) from sentence (1). Constructions (2) and (3) remain ambiguous, and disambiguation takes places at the semantic level (selection of the "correct" semantic unit among those associated with the single syntactic unit).

Case number 2:

For a syntax operating in connection with semantics, one has to have at hand the necessary means to distinguish the transitive form of "tra"ner" from its intransitive form, i.e. give option a linguistic status and not a technical status.

In fact, functions alone are not sufficient to justify the separation in `P0 P1` for (1), `P0` for (2) and (3). Thematic roles implicitely allow this distinction:

```
(1) cb : P0 SELF P1

P0[RoleTh:AGENT]

P1[RoleTh:THEME]

(2)and(3) cb : P0 SELF

P0[RoleTh:THEME]
```

Note: interpretations (2) and (3) will be differentiated in the semantic layer.

GENELEX allows those who want to make this distribution to explicit what is implicit by specifying the thematic roles on positions right from the syntactic phase.

# 4: Positions and Constructions

Let us remind that a Construction is defined by an ordered list ranking from 1 to n positions, their option, and their interdependency.

Besides, one may wish to express the insertion point of SELF in the construction (or the phrase in which SELF occurs). To do so, the attribute *insereself* that takes the value *i* is used. It means that SELF is inserted before the position *Pi*. If SELF comes after all the positions of the construction, the value of *i* will be the value of the last position +1. If one does not want to record the insertion point, the attribute will not be documented.

## 4.1: Optionality

## 4.1.1: Simple options

The actualization of positions can be obligatory or optional; **option** is expressed in brackets.

*Ex : Jean* ***s'adresse*** *ˆ ses invitŽs* (Jean is speaking to his guests)

`cb : P0 SELF P1` => P1 obligatory

*Il* ***mange*** *du pain* (He is eating bread)

*Il* ***mange*** (He is eating)

`cb : P0 SELF (P1)` => P1 optional

When several positions are optional, then we obtain a set of possible surface realizations when combining them.

*Ex : Pierre* ***parle*** *de sa soirŽe ˆ Marie*

(Pierre is telling Marie about his party)

*Pierre* ***parle*** *de sa soirŽe*

(Pierre is talking about his party)

*Pierre* ***parle*** *ˆ Marie*

(Pierre is talking to Marie)

*Pierre* ***parle***

(Pierre is talking)

```
P0 (P1) (P2)

-> P0 P1 P2

-> P0 P1

-> P0 P2

-> P0
```

## 4.1.2: Distant or interdependent options

The fact that a position is effectively realized in surface (or not) may have consequences on the realization of other positions :

- their realization becomes forbidden

- their realization becomes obligatory

These phenomena will be expressed by conditions on options. Optionalities make it possible to express constraints on the mutual option of positions.

*Ex : Jean* ***attire*** *Marie ˆ lui* (Jean caught hold of Marie)

*Jean* ***attire*** *Marie* (Jean appeals to Marie)

*Jean* ***attire*** (Jean is attractive)

**Jean* ***attire*** *ˆ lui*

```
P0 (P1) (P2)

Cond : Si !P1 Alors !P2
```

As conditions are oriented, one has to express as many conditions as necessary.

*Ex : Cela* ***ajoute*** *des souffrances ˆ sa peine*

(This adds suffering to his/her pain)

*Cela* ***ajoute*** *des souffrances*

(This adds suffering)

*Cela* ***ajoute*** *ˆ sa peine*

(This adds to his/her pain)

**Cela* ***ajoute***

**(**This adds)

```
P0 (P1) (P2)

Cond : Si !P1 Alors P2

Cond : Si !P2 Alors P1
```

It will then be possible to give account of phenomena such as :

- **Positions excluding one another:**

Positions cannot be realized at the same time, the occurrence of the one requires the absence of the other and vice versa:

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

(Jean says that Marie is beautiful)

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

(Jean tells Paul that Marie is beautiful)

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

(Jean tells Paul that Marie is beautiful)

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

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

P0 : SN

P1 : SP[Prep:ˆ]

P2 : SP[Prep:de]
```

```
P3 : P[SsCatSynt:COMPLETIVE]

Cond : Si P1 Alors !P2

Cond : Si P2 Alors !P1
```

(It is a case in which the lexicographer has gathered all arguments in a same cb and where the maximum structure is never completely realized. P1 and P2 cannot be present simultaneously.)

- **Positions being both optional**:

Positions can only be realized or deleted together, the absence of the one requires the absence of the other:

```
(P1) (P2)

Cond : Si !P1 Alors !P2

Cond : Si !P2 Alors !P1
```

- **Any other** more or less complex **constraint**:

In particular, the expressive power of such conditions is valid whatever the number of optional Positions (especially when more than two are optional).

*Ex : Il* ***a ŽtŽ rŽpondu*** *oui ˆ Luc par le service du personnel*

(Luc was answered yes by the personnel department)

*Il* ***a ŽtŽ rŽpondu*** *oui ˆ Luc*

(Luc was answered yes)

*Il* ***a ŽtŽ rŽpondu*** *oui par le service du personnel*

(The personnel department answered yes)

*Il* ***a ŽtŽ rŽpondu*** *oui*

(Yes was answered)

*Il* ***a ŽtŽ rŽpondu*** *ˆ Luc*

(Luc was answered)

```
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:ˆ]

P3 : SP[Prep:par]

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

Cond : Si !P2 Alors P1

Or in extension:

P0 SELF P1 P2 P3

P0 SELF P1 P2

P0 SELF P1 P3

P0 SELF P1

P0 SELF P2
```

## 4.2: Actualization of Positions

## 4.2.1: Linguistic phenomena

As previously seen, a position (formally defined by a substitution paradigm, i.e. the set of phrases that can occupy it), may or may not be realized in surface, in other words it may or may not be actualized.

As regards the actualization of positions in surface, different linguistic phenomena that are usually distinguished require comments:

1. adjunction of modifiers

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

(The left door of the landing that I have opened)

2. coordination

*Ex : La porte gauche et qui s'ouvre mal*

(the left door that does not open well)

*La porte que j'ai ouverte et refermŽe*

(the door that I have opened and closed)

3. apposition

*Ex : La porte, ce seuil de la maison*

(the door, this home doorway)

- **Atomist viewpoint**

The 3 linguistic phenomena are distinct surface phenomena that require different analyses and treatments in terms of grammar.

They are not the responsibility of the lexicon that only records the positions called by the lexical entry; the grammar manages their simple or multiple actualization and gives the adequate representation.

At the level of the lexicon, we consider that any position may have no, a simple or a multiple realization, i.e. it can be instantiated by the Phrases entering into its distribution, whatever its function. These Phrases range from 0 to N if the Position is optional, from 1 to N if the Position is obligatory.

-

- **Syntactist viewpoint**

The three phenomena cover very different linguistic facts.

1. adjunction of modifiers

It is considered as a property of the function associated with the position, since modifiers can intrinsically be repeated N times.

On the contrary, the subject function has the property not to be repeatable.

2. coordination

It is considered as a meta-operation (see HPSG) on phrases. This operation potentially applies to all the phrases of a same distribution (position) according to very complex grammar rules.

It is not the responsibility of the lexicon, except for indicating obligatory coordinations

*Ex : A ses risques et pŽrils* (at one's own risks) and for giving the representation corresponding to the theoretical frame

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

-

3. apposition

It is considered as a stylistic element external to positions, their occupants and managed by the grammar - in the best case.

It is to be noted that these linguistic phenomena (in particular, the adjunction of modifiers) are interpreted very differently, depending on the atomist or syntactist viewpoint one has on the lexicon.

Consequently, the indicator of repeatability borne by positions will be interpreted differently, depending on the theoretical model selected.

## 4.2.2: Indicator of repeatability on Positions

The possible number of surface realizations is expressed by an indicator of repeatability borne by the position. This indicator (***repetable***) takes the values OUI/NON (the latter particularly applies to compound words, and positions saturated by a component).

These are local properties of positions (as opposed to distant options).

-

- **Atomist viewpoint**

Unless otherwise specified, any position can be actualized n times, optional positions can be actualized 0 to n times: in all cases, the attribute *repetable* is implicitely set on OUI.

We do not want to prevent ourselves from the possibility to suppress this property enabling multiple actualization, if necessary. In those cases only, we will use the value NON.

- **Syntactist viewpoint**

Unless otherwise specified, positions cannot be repeated and can only be actualized by a single occupant, optional positions are actualized by an occupant or not: in both cases, *repetable* = NON.

To indicate that a position can be repeated, we use the value OUI.

However and fortunately in both cases, these values will permit to give account of the same linguistic phenomena, i.e. they will give account of the surface cooccurrence of an SADJ, SP, P[SsCatSynt: RELATIVE] for instance.

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

(the left door of the landing that I have opened)

```
(P0) P1* SELF P2*

P2 : SADJ

SP[Prep:de]

P[SsCatSynt:RELATIVE]
```

Only interpretation changes: in the atomist viewpoint, the occupant is repeated, in the syntactist viewpoint the position is repeated.

## 4.3: Linearity

A canonical linearity is conventionally established: the "rank" of apparition of a position in a construction (noted with a number in the examples) does not necessarily correspond to a specific place in the surface linear order. This rank is (partially) arbitrarily given in the canonical linear order, and it designates the recording order in a construction. The term "position" cannot be understood in the strict meaning of place (distinction established by Milner [1989]), since the constraint on surface linearity has been dismissed.

## 4.3.1: Canonical linearity

We want to refer to positions as defined in the maximal structure [Fradin & Marandin, 1979], even if this maximal structure is never to be found in surface. Thus, certain verbs described as having three complements will never have these three complements simultaneously realized in surface.

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

(Jean says that Marie is beautiful)

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

(Jean tells Paul that Marie is beautiful)

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

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

(See inter-conditioned positions)

The disruption of the canonical linearity is part of a system of grammar rules, even stylistic rules, that may depend on the type of positions and that do not generally apply when describing the constructions of a syntactic unit.

## 4.3.2: Linearity and transformations

Certain pronominalization phenomena may change the realization order of these positions in surface.

*Ex: Je* ***pense*** *ˆ Pierre* (I am thinking of Pierre)

*Je* ***pense*** *ˆ lui* (I am thinking of him)

*J'y* ***pense*** (I am thinking of him)

French offers a rather large freedom as regards the place occupied by the different constituents of the utterance:

*Ex : Je* ***promets*** *de venir ˆ Pierre* (I have promised Pierre to come)

*Je* ***promets*** *ˆ Pierre de venir* (I have promised Pierre to come)

## 4.3.3: Linearity and option

We have seen that position realizations can be optional. The realizations P0 P1 P2, P0 P1, P0 P2 and P0 can correspond to a structure P0 (P1) (P2).

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

(Someone is talking about something to someone)

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

(Someone is talking about something)

*Quelqu'un* ***parle*** *ˆ quelqu'un*

(Someone is talking to someone)

*Quelqu'un* ***parle***

(Someone is talking)

Omitting P1 must not lead to the redefinition of P2 in P1.

## 4.4: Surface phenomena

Since the positions of a construction are canonically ordered, it is allowed to indicate the place of Self among these positions, using the attribute *insereself* (of course, in the syntactist viewpoint in which the construction is considered as a phrase).

```
Ex : P0 SELF P1 (P2)
```

```
corresponds to insereself = 1
```

Besides, if the syntactic unit is compound, it will be possible to note eventual phenomena such as elements of the external construction re-entering in the internal structure, using the attribute *insereinsertion_l*. It functions like the attribute *insereself*, except that several insertions may follow one another between two positions.

```
Ex : P0 I0 P1
```

```
corresponds to insereinsertion_l = 1
```

These indications do not change the canonical numbering, nor they question the canonicity of the numbering of positions; in other words, rules for changing the surface linearity can still apply.

The surface order may however be constrained by such or such position:

*Ex : Il* ***craint*** *de Marie qu'elle ne vienne*

(He is afraid/fears Marie will come)

```
cb : P0 SELF P1 P2

P0 : SN

P1 : SP[Prep:de]

P2 : P[SsCatSynt:COMPLETIVE]
```

It is not allowed to indicate, in the current vision of the model, how to inhibit the rules for changing the linearity.

## 4.5: Interdependency of Positions

Certain positions are firmly attached to one another, i.e. they do not allow any type of insertion: they form an undividable sequence. This convention is useful for composition, to prevent from inserting interpolated clauses, appositions, etc. However the order of apparition in surface is not necessarily constrained (cf. *temps plein, plein temps* - full time)

```
Ex : P1-P2
```

A **dash** between two positions indicates that these two positions are firmly attached to one another, i.e. no element can be inserted between them and if they move, they move together.

## 4.6: Realizations of inter-conditioned Positions

We want to be able to express that position realizations can constrain the realization of other positions. We will have to use **Conditions** to express these constraints.

*Ex : Pierre* ***rŽpond*** *ˆ la question*

(Pierre answered the question)

*Pierre **rŽpond** que c'est exact.*

(Pierre answered that it was true)

*Pierre **rŽpond** ˆ sa m•re que c’est exact.*

(Pierre answered his mother that it was true)

*Qu'il ait une telle attitude **rŽpond** ˆ la question.*

(His acting this way answers the question)

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

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

P0 : SN

P[SsCatSynt:COMPLETIVE]

[Mode:SUBJONCTIF]

P1 : SP[Prep:ˆ]

P2 : P[SsCatSynt:COMPLETIVE]

[Mode:INDICATIF]

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

[Mode:SUBJONCTIF]

Alors !P2
```

*Ex : Il a **volŽ** un livre ˆ Marie* (he stole a book from Marie)

*Il a **volŽ** un livre* (he stole a book)

*Il a **volŽ** Marie* (he stole Marie)

> If the lexicographer decides that *Marie* and ˆ *Marie* are different realizations of the same position (RoleTh PATIENT), the realization of this position has to be conditioned by the actualization of the preceding:

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

P0 : SN
```

```
P1 : SN

P2 : SN

SP[Prep:ˆ]

Cond : Si !P1

Alors P2 = SN

Cond : Si P2 = SP[Prep:ˆ]

Alors !P1
```

### 4.7: Conditions between Self and Positions

Constraints of realization may also concern the actualization of the Self as Intervener in the Construction.

*Ex : Jean **a descendu** la colline*

(Jean went down the hill)

*Jean **est descendu** de la colline*

(Jean has come down from the hill)

If we choose to have a single Usyn describing the purely "location" use of *descendre* (as opposed to *descendre quelqu'un* -to bump off someone- or *sa valise* - one's suitcase), then we can associate it with the following conditions:

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

Alors P1 != SN

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

Alors P1 != SP[SsCatSynt:LIEU]
```

# VII: Phrases

## 1: General definition

A position may be actualized in various ways. A **Phrase** is a given realization of position.

A phrase occupying a position is formally described by a terminal or non-terminal syntagmatic label (***etiquettesynt*** attribute) with which a set of constraints is associated, if any (rewriting constraints - only for non-terminal phrases - and restricting features).

## 2: Syntagmatic Label

We admit a priori a nomenclature of syntactic symbols of maximum level {P, SV, SN, SP, SADV, SADJ}, to which the empty element {e}, {Nbarre} (noted N') and the list of terminal categories as defined in morphology add.

These symbols are assumed to be known and described in a grammar external to the dictionary. GENELEX proposes a list that can be extended by the user.

We will give some details on the meaning of *e* and Nbarre.

The category *e* is generally used to describe

> - the optional character of an immediate constituent in distributional and transformational grammars. This is due to the poorness of the formalism and does not apply to GENELEX. Let us remind that the rewriting correspondence of DET_VIDE is the absence or option of the 'Specifieur' Position.
>
> - traces in generative grammar. This is the reason why it has been introduced in GENELEX and it cannot have another meaning. We want to avoid any misuse that would induce a position occupied by *e* to become an "Empty position" for formal and not linguistic reasons.

We have introduced the label Nbarre essentially for *syntagme_NT_S*, phrases allowing to describe the internal structures of compounds. Actually, if the compound can globally play the role of N as an **InterConst,** it is because it is not intrinsically a noun group: it lacks at least its pre-determiners. We have decided to use this label because it seemed to us the least ambiguous to indicate a level of realization between N and SN.

## 3: Feature Constraints

Phrases that occupy positions may be restricted by a set of **Features**. Combining these features allows to specify a position occupant. These restrictions may also be expressed by the rewriting of the phrase (see below). The lexicographer may save himself work and only use rewriting when the features he has at hand are insufficient to specify all the necessary information.

Using features (developed by unification grammars LFG, HPSG, etc.) enables the lexicographer to code syntactic descriptions at different levels of detail. It also enables a Generic-Specific hierarchy on phrases and to directly reach the required level: SP, SP[Prep:^], SP[Prep:^] [SsCatMorph:DEFINI], without having to explicitely declare this hierarchy. The link can be deduced from the sharing of the syntagmatic label and the inclusion of a list of constraints - as well as the inclusion of distributions and other relations between phrases when they are re-written.

Different families of features accessible from the syntactic layer are distinguished; all the features can add to one another.

```
Ex: V[Aux:ETRE]

[Pronominal:SE]
```

```
[Temps:COMPOSE]

[Aspect:PROCESSIF]
```

The use of restricting features is governed by a set of rules detailed in the chapter concerning them (Chapter VIII).

# 4: Structure constraints

In most cases, the label referring the phrase is sufficient to describe it as a position occupant, and no constraint on its structure needs be expressed for the described entry. In other words, it is an absolute prototype.

However one may need to express from a lexical entry constraints on the structure in which a lexical entry inserts.

To do so, it is possible to use:

- either syntactic sub-category features

- or lists of embedded positions allowing to describe *n*-depth syntactic trees; in that case, this is a tree-structured rewriting of phrases.

## 4.1: Syntactic sub-categorization feature

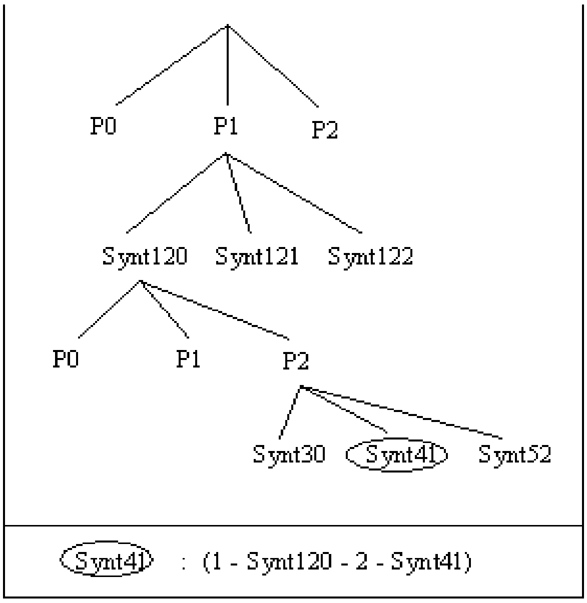
Certain values of syntactic sub-categories implicitly specify a certain type of structuring (e.g. DET_VIDE). When sub-categorization features are used for that purpose, one must keep in mind that the value of SsCatSynt is only a character string on which the formalism does not apply any consistency check: to be interpreted by a grammar, a feature must be defined. Cohesion must be checked by the team of lexicographers.

## 4.2: Tree-structured rewriting

### 4.2.1: Description

Rewriting phrases, (only applicable to non-terminal phrases) is done via the formalism of positions. Each phrase can be rewritten in the form of a list of positions, for which option conditions, surface order and interdependency may be specified.

A **syntactic tree** in GENELEX will have the position and the phrase alternate at each level:

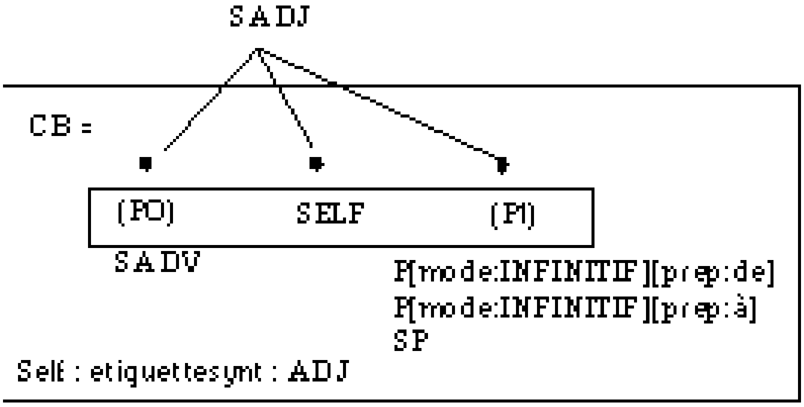


Progress in a syntactic tree is expressed by an alternance of integers and identifiers. Each integer indexes a position in its list, each identifier refers a rewriting phrase of that position.

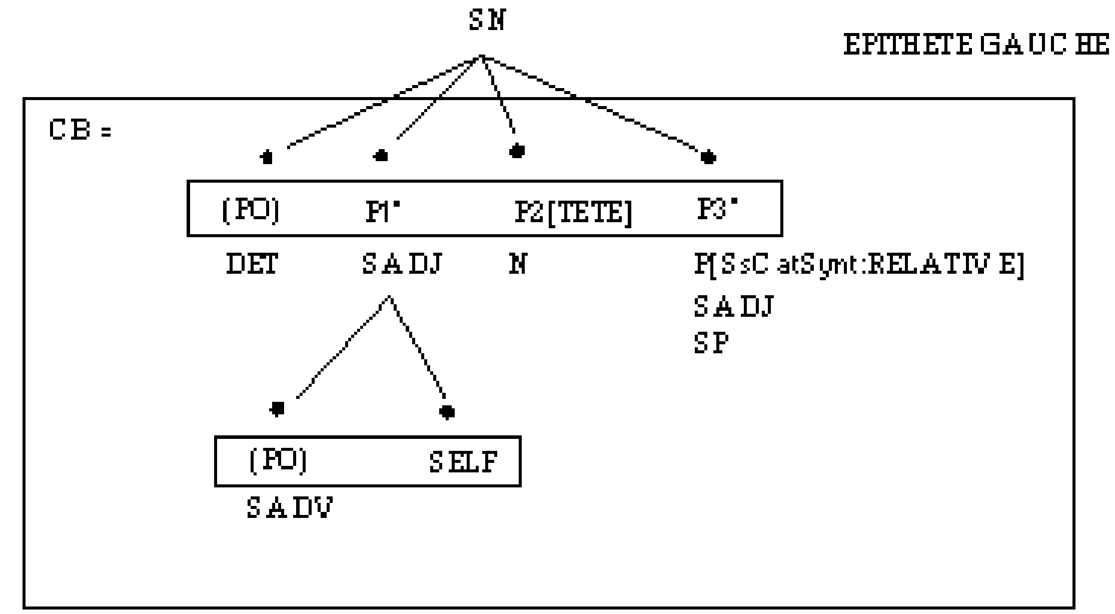
## 4.2.2: Examples of use

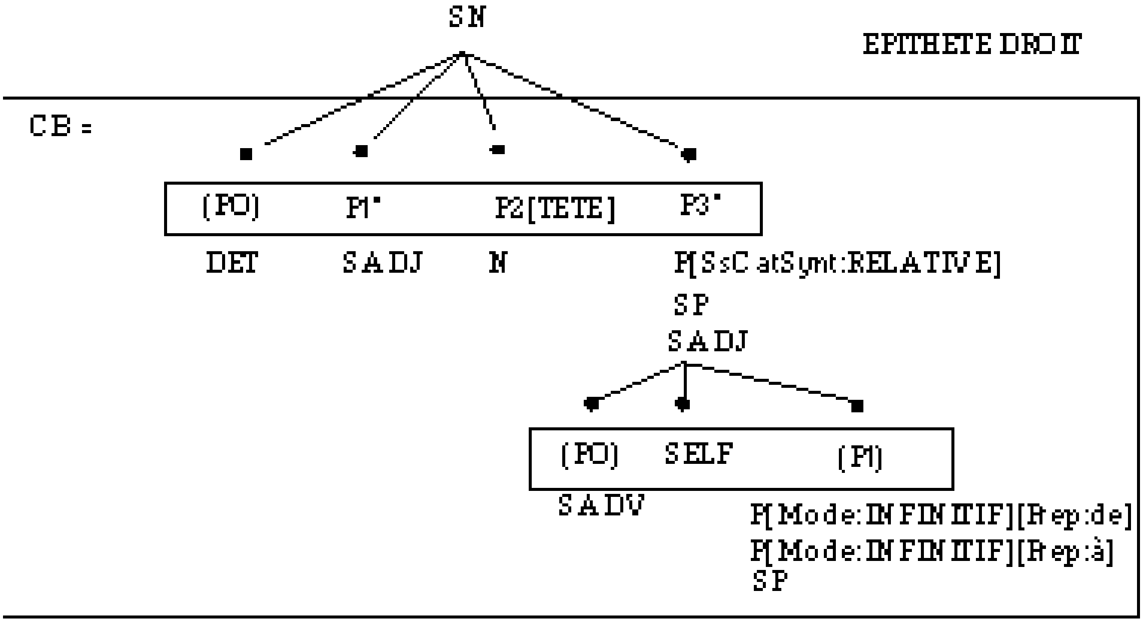
- **description of adjectives**

SADJ will be rewritten by GENELEX entities as follows:

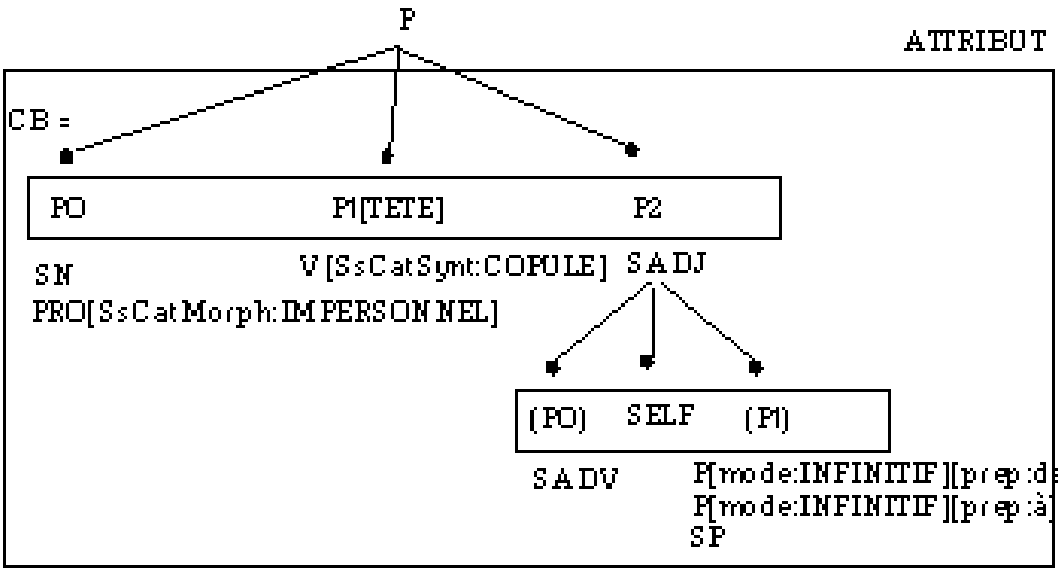

SADJ
CB =
(P0)
SELF
(P1)
SADV
P[mode:INFINITIF][prep:de]
P[mode:INFINITIF][prep:à]
SP
Self : etiquettesynt : ADJ

- **description of left- and right-positioned attributive adjectives**





- **description of subject predicative adjectives**



*Ex: Il est tr•s* ***intŽressant*** *de remarquer cela*

(It is very interesting to note that)

*Il est tr•s* ***intŽressant*** *que tu remarques cela.*

(It is very interesting you note that)

db: Self cb

Self: IntervConst: ADJ

cb: P0 P1 P2 P3

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

P1: V[SsCatSynt:COPULE]

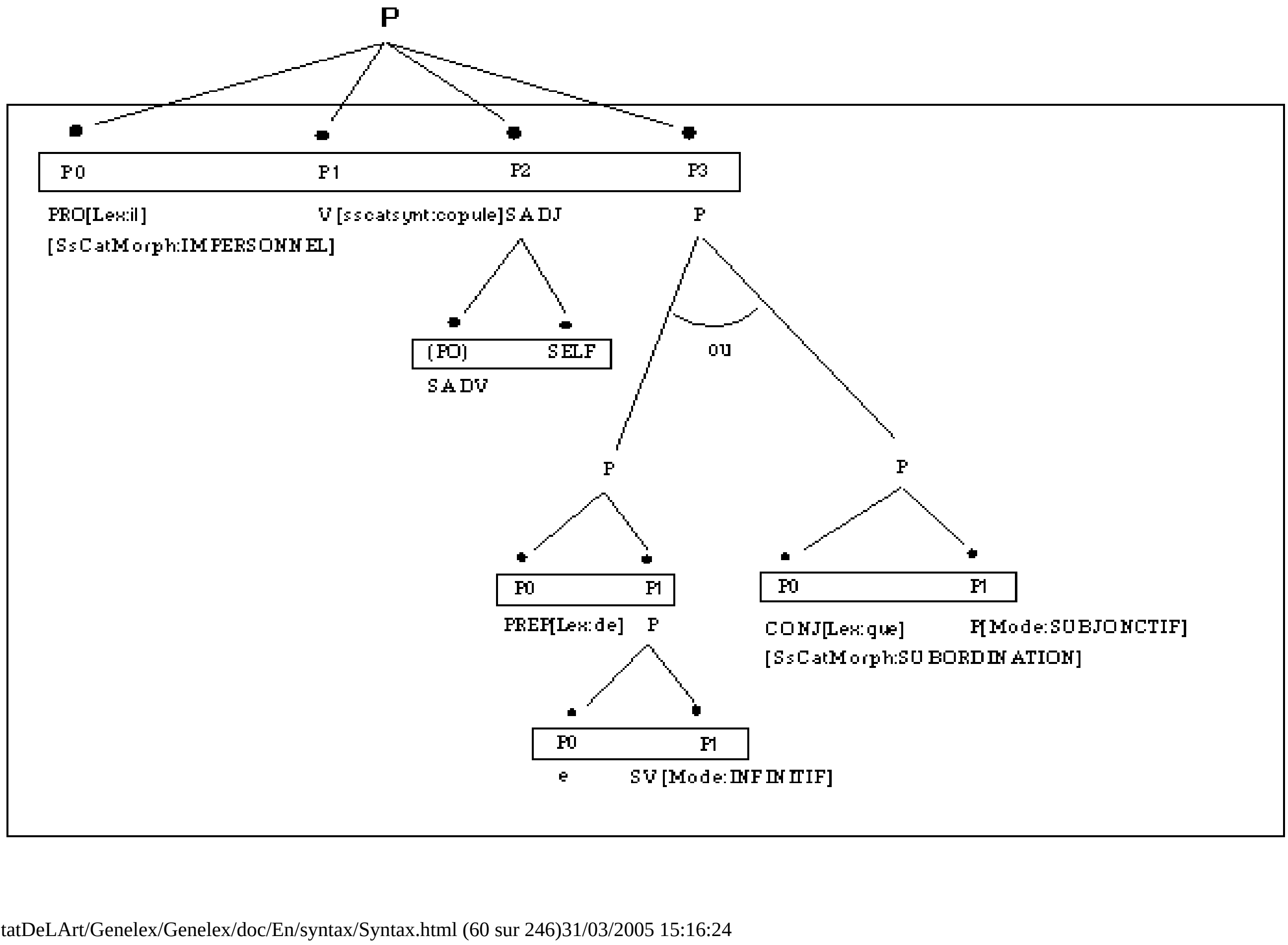
P2: **SADJ**

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

P[Conj:que][Mode:SUBJONCTIF]

SADJ: (P0) SELF

P0: SADV



### 4.3: Partial rewriting of Phrases

For certain entries we want to express partial restrictions on a phrase without having to rewrite it completely, because it is not always possible; for instance, in the case of a verbal phrase, we do not know how many positions there are if we do not know the head.

In that case, the structure of a "prototypic" phrase (list of positions) is maintained, but certain position occupants are restricted both at the level of their list (removal of occupants) and at the level of constraints (addition of restricting features) on each.

The solution is then to give only the list of positions that is restricted or the type of occupants, and to use the attribute *listepositions* whose values (OUVERTE, FERMEE) allow to specify whether the list of rewriting positions specifies entirely (FERMEE) or partially (OUVERTE) the rewriting of the phrase.

# VIII: Features

**Features** are restrictions adding to the syntagmatic label in the specification of a Phrase. (See Chapter VII).

First we will specify the different usable types and sub-types of features, their attributes and the values they can take.

Then we will see the different constraints governing their use, for each type of feature but also for the different possible combinations of these features.

## 1: Types, sub-types and values of features

### 1.1: Lexical features

Lexical features allow to constrain all or part of the lexicalization of a phrase.

To do so, it is possible:

- either to specify the canonical form of the "lexicalizing" Um,

- or directly point at this Um (thus removing any possible ambiguity on the written form).

There are two types of lexical features: introducers and the Lex feature itself.

### 1.1.1: Introducers (Prep, Conj, ProRel, ProIntrog and Introd)

**General**

These features allow to specify the lexicalization of phrase introducers without having to rewrite them. They are not ambiguous and only apply to non-terminal phrases.

Actually it is a sub-type that includes 5 features:

## Types of Features

### 1.- **Prep**

for the preposition introducing a non-terminal phrase, typically a Prepositional Phrase

The **Prep** feature has three uses, depending on the type of phrase that bears it:

```
P[Prep:ˆ][Mode:INFINITIF]
```

phrase introducer (full meaning)

```
SP[Prep:ˆ]
```

phrase rewriting element

To these two "standard" uses we add the possibility, for a preposition, to be an introducer in the full meaning of noun phrase. We may actually want to distinguish cases in which the preposition is governed by an element external to the prepositional phrase, which is a traditional case with verbs governing prepositions, for instance, from cases in which it seems to be governed by the head noun of the prepositional phrase as in "en nourrice" (in the care of a nurse) for instance. The non-terminal noun phrase will then bear the feature **Prep.**

```
SN[Prep:en][Lex:nourrice]
```

phrase introducer (full meaning)

### 2.- Conj

for the conjunction introducing a Sentence

### 3.- ProRel

for the relative pronoun introducing a Sentence

### 4.- ProIntrog

for interrogative pronouns that can be specified by verbs requiring interrogative clauses (mainly verbs of speech).

### 5.- Introd

for any non-terminal phrase introducer (particle, "recategorizer", etc.) not belonging to any of the four other types.

**Rewriting**

In the rewriting alternative, we will use the **Lex** feature (see next paragraph) on terminal phrases:

`Ex:` **`SP[Prep:^]`**

can be rewritten:

**`SP`**`: P0 P1`

`P0:` **`PREP[Lex:^]`**

`P1: SN[...]`

if we want to specify more SN than SP.

## 1.1.2: Lex

The **Lex** feature allows to specify the lexicalization of

- a syntagmatic leaf, if it applies to a terminal phrase

- the head of a Phrase, if it applies to a non-terminal Phrase. Since we have Introducers, we can admit that syntagmatic heads are:

**N** for SN

**N** for SP

**V** for SV or P

**Adj** for SADJ

**Adv** for SADV

It is to be noted that the structuring of that phrase is implicitely determined by the lexicalized head, more precisely by the syntactic behavior of its head Usyn (that exists on its own). If we want to apply further constraints on the structuring of the phrase, we can use sub-category features, or rewrite it.

A particular case of restriction is when the phrase is saturated by its head, i.e. the noun phrase is limited to the noun. To express this property, we use the attribute *saturesynt*. When it has the value OUI, the head of the phrase is the only leaf with the Introducer, if any. This applies to phrases entering in the definition of simple and compound Usyns.

The field *saturesynt* must always have the value OUI when the feature applies to a terminal category: by definition, the leaf is saturated by the lexicalizing element.

## 1.1.3: Cooccurrence of lexical features

A same phrase can only bear one and only one Lex feature that will lexicalize the terminal phrase in the one case, and the head, and only the head of the non-terminal phrase, in the other case.

Since each non-terminal phrase has only one introducer, then a same phrase can only bear one feature specifying its introducer.

On the contrary, for non-terminal phrases, introducers may be combined to the Lex feature.

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

These remarks are also valid by reflection for the features RefLex, RefPrep, RefConj, RefProRel, RefProIntrog and RefIntrod; these features are described in the chapter on compound syntactic units.

## 1.1.4: Particular case of Agglutinates

Agglutination is a morphological phenomenon that has no influence on the syntactic representation.

Therefore, the Lex feature will not usually be used to reference an agglutinate. It will be preferable to find this agglutinate by referring to the agglutinating elements.

Consequently, a same Phrase

```
Ex: SP[Prep:ˆ][SsCatMorph:DEFINI]
```

may or may not be realized in surface by an agglutinated form:

*Ex:* ***au*** *pharmacien* (to/at the chemist)

*ˆ* ***la*** *pharmacienne* (to/at the chemist)

## 1.2: Morphological features

They allow to express a restriction on the value of a morphological feature (**Mode**, **Temps**, **Personne**, **Genre**, **Nombre** and **NombrePosseur)** of the phrase or of one of its components.

```
Ex: SN[Nombre:PLURIEL]
```

Mode, Temps, Personne, Genre, Nombre and NombrePosseur are features that can combine with one other to form specifically a combination of morphological features.

```
Ex: [Mode:SUBJONCTIF]
```

```
[Temps:PRESENT]
```

```
[Personne:3]

[Nombre:SINGULIER]
```

## 1.3: Morpho-syntactic features

This family of features includes **SsCatMorph**, **Aux**, **Pronominal** and **Accord**. All the constraints expressed by these features may add to one other.

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

## 1.3.1: SsCatMorph

This feature takes as possible values the morphological subcategories (refer to the GENELEX report on the Morphological Layer) that are criteria for splitting UMs.

They may apply to terminal and non-terminal phrases.

```
Ex: SN[SsCatMorph:DEFINI]

DET[SsCatMorph:DEFINI]
```

The list of values is finite and corresponds to the list of grammatical subcategories defined in morphology:

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.

It is to be noted that certain values of morphological subcategory features entirely cover a paradigm of inflected forms.

It is the case of definite and partitive determiners:

- Definite determiner {*le, l', la, les*}
- Partitive determiner {*du, de l', de la, des*}

In those cases, and only in those cases, it will always be possible to add or substitute a lexical feature to the subcategory feature:

```
Ex: DET[SsCatMorph:DEFINI]
```

DET[Lex:le]

DET[SsCatMorph:DEFINI][Lex:le]

3 phrases inducing a same lexicalization.

However and even though they refer to the same unit, the sub-category feature remains the most explicit and powerful, since it can apply to a non-terminal phrase.

Ex: SN[SsCatMorph:PARTITIF]

## 1.3.2: Aux

**Aux** is a complex feature whose **Valeur** (ETRE or AVOIR) may be completed by a restriction of **Temps** and **Mode** on the auxiliary.

Most of the time, only the Value is documented, and specifies the auxiliary associated to a given verb in such-and-such syntactic turn:

*Ex: **descendre** ˆ la cave (**•tre** descendu)*

(to go down to the cellar) (to be gone down)

V[Lex:descendre][Aux:ETRE]

***descendre** les escaliers (**avoir** descendu)*

(to go down the steps) (to have gone down the steps)

V[Lex:descendre][Aux:AVOIR]

It may be necessary to specify, for a given turn, that the verb is in a compound tense:

*Ex: **•tre arrivŽ** socialement*

(to have succeeded (in life))

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

But it may also be necessary to constrain the tense and mood of the conjugation auxiliary:

*Ex: **Žtant donnŽ***

(since, given)

V[Lex:donner][Temps:COMPOSE]

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

## 1.3.3: Pronominal

This feature allows to specify for a given verb the non-referential preverbal particle that is associated with that verb in such-and-such syntactic turn. This is the way to indicate the behaviors of "true pronominal" verbs.

*Ex:* ***se plaindre***

(to complain)

```
V[Lex:plaindre][Pronominal:SE]
```

The values of this feature in French are the following:

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

*Ex:* ***les avoir*** *ˆ zŽro;* ***en baver;***

(to be scared out off one's wits) (to have a rough time of it)

***la ramener; se le tenir*** *pour dit;*

(to kick up a row) (to be warned once and for all)

***se la couler*** *douce;* ***s'y coller;***

(to have it easy) (to get stuck into something)

***s'en moquer.***

(not to care about anything/anybody)

It is to be noted that the preverbal particle can be specified on the Self of a simple or compound unit.

*Ex: se rendre compte* (to become aware)

## 1.3.4: Accord

When applying to several terminal or non-terminal phrases, this feature allows to express agreement constraints determined by a lexical unit, namely certain determining nouns.

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

(half people came)

```
Self: IntervConst: N[Nombre:SINGULIER]
```

```
[Accord:I]

cb: P0 P1

P0: SN

P1: SV[Accord:I]

SV[Accord:J]

where P0 : SN is rewritten:

SN: P0 (P1) SELF (P2)

P0: DET

P1: SADJ

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

Therefore, this feature allows to explicitely record difficult agreement rules. It goes without saying that, in the absence of this feature, general agreement rules apply.

## 1.4: Syntactic features

This family of features includes **SsCatSynt, Tournure, Neg** and **Passif**. All the constraints expressed by these features may add to one another.

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

## 1.4.1: SsCatSynt

This feature takes as possible values the syntactic subcategories that label subsets of syntactic behaviors for a given category.

They can apply to terminal or non-terminal phrases.

```
Ex: P[SsCatSynt:COMPLETIVE]

V[SsCatSynt:COPULE]

SP[SsCatSynt:LIEU]

SP[SsCatSynt:DET_VIDE][Prep:en]
```

`[Lex:consŽquence]` *(en consŽquence)*

consequence in consequence

The list of values can be extended, but Genelex proposes the following default values:

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

## 1.4.2: Tournure

The presence of a **Tournure** feature allows to indicate whether a Sentence is obligatorily EXCLAMATIVE or INTERROGATIVE

We must here make a digression on interrogative clauses. A minimum typology allows to distinguish:

- subordinate interrogative clauses (relative or not) that, depending on the verb, come under the reported speech, direct or indirect :

Ex *: SN demander* ***si/pourquoi qqchose***

(to ask whether/if/why something)

*SN demander* ***quel/combien***

(to ask which/how much/how many)

Ex *: SN demander* ***"pourquoi qqchose"***

(to ask why something)

*SN demander* ***"quel/combien "***

(to ask which/how much/how many)

These are sub-categories (**SsCatSynt** feature) of sentences entering in the complementation of a verb.

- "free" main clauses that are not reported in the speech. These are called interrogative forms (**Tournure** feature). These sentences have consequences on the inversion of the subject.

We have therefore two values of interrogative, connected with two different features:

**SsCatSynt**: INTERROGATIVE_DRI, INTERROGATIVE_DRD

**Tournure**: INTERROGATIVE

## 1.4.3: Neg

The presence of a **Neg** feature indicates that the phrase described is in the negative form. In that case,

- either the lexical value of the negation is free (value: LIBRE)

*Ex:* ***n'arr•ter pas/plus/jamais/...***

- or it is constrained, then chosen among:

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 b•tises*

(to do nothing but silly things)

## 1.4.4: Passif

The presence of a **Passif** feature indicates that the Sentence described is in the passive form. This feature is Boolean (values: PLUS or MOINS).

**Note**: the absence of the Passif feature involves an active form, which prevents from recording the feature [Passif:MOINS] for each entry.

## 1.5: Syntactico-semantic features

## 1.5.1: Coref

The coreference feature (**Coref**) is the only representative of this family of features.

It allows to express the coreference indices used in distributional grammars to give account of the pronominal reference, the control of infinitive clauses, etc.

This feature can take the following values:

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

*Ex: (se)* ***laver*** => **rŽflŽchi**

**(**to wash (oneself) => reflexive)

```
cb: P0 SELF P1
```

```
P0: SN[Coref:I]

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

We express here the coreference between the subject and the reflexive pronoun, since the verb is not an essential reflexive verb.

Conventionally, when the feature Coref applies to a Sentence, it concerns the subject of P.

Coreferences must be resolvable: if a feature with a value I is present on a Phrase of a position in a given construction, there is at least one feature I or NON_I that corresponds to it on a phrase of another position in the same construction. Coref features do not impose the co-realization of the phrases that bear them. If we want to impose this co-realization, we will do it as usual by means of conditions applying to position realizations.

**Control of the infinitive**

This control may take several forms, among them:

Control of the **infinitive subject**

- by the subject of the main clause

*Ex : Marie* ***aime*** *danser* (Marie likes dancing)

```
cb : P0 SELF P1

P0 : SN[Coref:I]

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

- by the direct object of the main clause

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

(Marie can see the children dancing)

```
cb : P0 SELF P1

P0 : SN

P1 : SN[Coref:I]

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

In this case, the object complement has been separated from the infinitive; we can also consider that the infinitive is an integral part of the object, in which case the infinitive has an expressed subject that is not concerned by coreference.

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

(Marie can see the children dancing)

```
cb : P0 SELF P1

P0 : SN

P1 : P[Mode:INFINITIF]
```

- by the subject and the direct object of the main clause

The cooccurrence of two coreference features (using two different variables) is to be understood as the "sum" of the values:

*Ex : Marie* ***emm•ne*** *son fils faire un voyage*

(Marie has taken her son for a trip)

```
P0 : SN[Coref:I]

P1 : SN[Coref:J]

P2 : P[Mode:INFINITIF]

[Coref:I][Coref:J]
```

- by the indirect object (or any other second object) of the main clause

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

(Jean has asked Marie to come)

```
P0: SN[Coref:I]

P1: SP[Prep:ˆ][Coref:J]

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

P[Mode:INFINITIF][Coref J]

Cond: Si P1 Alors P2[Coref:J]
```

Control of the **infinitive object**

- by the object of the main clause (presented here as separated from the infinitive)

*Ex : Jean* ***a donnŽ*** *ce travail ˆ faire*

(Jean has given this work to be done)

P0 : SN

P1 : SN[Coref:I]

P2 : P[Mode:INFINITIF][prep:ˆ]

where

P[Mode:INFINITIF]

[*listepositions* : OUVERTE] :

P0[SUJET] : e

P1[OBJET_DIRECT] : e[Coref:I]

- by the subject of the main clause

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

(This boat is hard to drive)

P0 : SN[Coref:I]

P1 : SADJ

where

SADJ

[*listepositions* : OUVERTE] :

P1 : P[Mode:INFINITIF]

where

P[Mode:INFINITIF]

[*listepositions* : OUVERTE] :

P0[SUJET] : e

```
P1[OBJET_DIRECT] : e[Coref:I]
```

## 1.6: Semantic features

For those who have chosen to explicitely use semantic properties right from the syntax level, it is possible to record these properties in the form of features.

These features cover **Aspect** of the verb, denotative conditions (**Bin** feature) and semantic class (**Libre** feature).

### 1.6.1: Aspect of the verb

Certain verbs appearing in the context of the entry are only used with such or such aspect.

Certain verb constructs constrain a PROCESSIF, STATIF or RESULTATIF **Aspect**.

```
Ex: V[Aspect:PROCESSIF]
```

### 1.6.2: Denotative Conditions (Bin features)

CDs (a notion coined by G.B•s) note the minimum differences between two constructions. Let us take the case of a verb for which there are a P0 and a P1[Fonction:OBJET_DIRECT]. Different objects, therefore P1, can be distinguished thanks to their denotative conditions.

*Ex: (1) il* ***mange*** (he eats)

*(2) il* ***mange*** *une pomme* (he eats an apple)

```
P0 (P1)
```

*(3) il* ***mange*** *les virgules* (he swallows commas)

```
P0 P1
```

The purpose here is to highlight two structures P0 P1 and P0 (P1), i.e. what distinguishes the optional P1 from the other. These syntactic properties are related to the denotative conditions. What we record is the minimum semantic information necessary to characterize a syntactic property (option or any other syntactic property).

We must specify here that the denotative conditions we are dealing with are not denotative conditions on the interpretation universe of the utterance. The semantics does not rely on truth values, nor on the likelihood of an utterance.

In effect, if *"un ŽlŽphant donne une pomme ˆ Marie" (an elephant is giving an apple to Marie)* is not a plausible utterance, it may however be perfectly interpreted.

The denotative conditions that we record are the conditions necessary to block certain interpretation processes: in *"Jean mange la virgule"*, "manger" is understood as "omettre" (to omit), which is not the case for the following utterances: *"Jean mange la pomme", "Jean mange".*

Using semantic indications explicitely, one gives account of a property that has implications on the semantic level (here P1 is obligatory). The denotative conditions to be noted are consequently [comestible:MOINS] for P1 in (2) and [comestible:PLUS] for P1 in (1). Denotative conditions are semantic features of binary values.

Note: figurative uses such as *"la machine mange la carte"* (the cash dispenser has swallowed the card) and *"cette voiture mange beaucoup d'essence"* (this car gets through much petrol) concern the structure P0 (P1). In other words and in a "figurative" way, petrol is edible by the car.

### 1.6.3: Semantic class (Libre features)

Very close from denotative conditions, the semantic class formally differs from them in that it has no binary values +/- but a set of free values; besides, the name of the class is rather a superordinate designating a set of possible lexical realizations.

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

(color: yellow, blue, red...)

***discours**: speech, confŽrence, plaidoyer, ...*

(speech: small talk, conference, plea...)

***v•tement**: veste, pantalon, chemise, ...*

(clothes: jacket, trousers, shirt...)

***profession**: mŽdecin, commer•ant, artisan, ...*

(occupation: physician, shopkeeper, craftsman...)

The features Bin and Libre can appear together on a same category, namely the category NOM (name).

```
Ex: NOM[animŽ:PLUS][classe:profession]
```

## 2: Constraints of integrity on Features

### 2.1: General rules for using Features

- Features apply to phrases and constructions.
- A feature is at least defined by its name and its value (either taken from a list of possible values, or free).
- It is possible to combine as many features as wanted to express a complex restriction.

- Any list of features on a phrase must be understood as a list "and", i.e. all the features of that list add restrictions. It is not possible to express an alternative of features on a same phrase. It is compulsory to go through the position realization alternative and therefore, through several phrases.

```
Ex: P[Prep:ˆ][Mode:INFINITIF]
```

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

- When the set of phrases occupying a position share one or several features, these features must be reported on each phrase. The position does not factor the features of its occupants.

However, depending on their type and value, all features cannot apply to all phrases, and all combinations of features are not licit. Therefore we will define for each family of features, its related conditions of use and constraints of integrity.

## 2.2: Subtypes of Features authorized depending on the syntagmatic label

All phrases can potentially bear a feature whose values are free (since these values are not known a priori). The features concerned are: **Lex**, **Libre**; the features **Bin** have also no restriction of use on the category of the bearing phrase: actually, if their values are constrained (+ or -), their names, on the contrary, are free.

- **Inheritance of Features between categories**

Non-terminal categories "inherit" the features of certain categories entering in their rewriting. The following table specifies the inheritance rules for each listed non-terminal category:

**EtiquetteSynt: Inheritance:**

------------------- -------------------------------

P Morphological Features of VERBE

SV All Features of VERBE

SN All Features of NOM

All Features of ADJECTIF

All Features of DETERMINANT

SP All Features of SN

All Features of PREPOSITION

SADJ All Features of ADJECTIF

SADV All Features of ADVERBE

In these cases, the value of the feature, when it "signs a category" (i.e. when it is only licit for one category), allows to determine the phrase to which this feature applies (i.e. the phrase that bears the feature or one of its rewriting elements).

```
Ex: SN[SsCatMorph:DEFINI]
```

applies to the DET of the SN

```
SN[Nombre:PLURIEL]
```

applies to the SN

If the value of the feature can apply to several categories, the value of the feature remains implicit. It is the lexicographer's task to estimate the shortcomings - or advantages - of this implicit and to rewrite it or not.

```
Ex: SN[SsCatMorph:CARDINAL]
```

***deux*** *enfants* (two children)

*les* ***trois*** *enfants* (the three children)

In the following tables of this chapter, for non-terminal categories we will only specify the Features or values of Features that cannot be deduced from these inheritance rules.

- **Table of sub-types of Features authorized per category**

Given a terminal or non-terminal phrase identified by a syntagmatic label, only a subset of the other features can be associated with it.

**EtiquetteSynt Sub-Types of Features authorized**

------------------- ------------------------------------------

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

CONJONCTION SsCatMorph

INTERJECTION *No feature*

PARTICULE *No feature*

P Conj

Prep

ProRel

Introd

SsCatMorph

SsCatSynt

Neg

Tournure

Passif

Coref

SN Introd

Prep

SADV Introd

e *All features except* SsCatSynt

## 2.3: Values of Sub-category Features authorized depending on the syntagmatic label

The values of sub-category, either morphological or syntactic, depend on the category of the phrase. In effect, a value of category restricts the set of values of possible sub-categories. This will have to be managed by constraints of integrity.

The default values of **SsCatMorph** have been defined so that they can mutually exclude one another when they are associated with the same category.

We have taken the same precaution for the default values of **SsCatSynt.** The user who adds values to SsCatSynt has to make sure he/she respects this constraint.

**Note**: features of morphological sub-category are distinguished from features of syntactic sub-category in that the features of morphological sub-category are splitting criteria for Ums.

**EtiquetteSynt Authorized values of SsCatMorph:**

**---------------- -----------------------------------------------**

verbe *No feature of SsCatMorph*

nom propre

commun

adjectif indŽfini

possessif

interrogatif

cardinal

ordinal

exclamatif

qualificatif

comparatif_ŽgalitŽ

comparatif_supŽrioritŽ

comparatif_infŽrioritŽ

superlatif_supŽrioritŽ

superlatif_infŽrioritŽ

superlatif_absolu

adverbe comparatif_ŽgalitŽ

comparatif_supŽrioritŽ

comparatif_infŽrioritŽ

superlatif_supŽrioritŽ

superlatif_infŽrioritŽ

superlatif_absolu

dŽterminant possessif

dŽmonstratif

partitif

indŽfini

dŽfini

exclamatif

interrogatif

cardinal

relatif

pronom personnel_fort

personnel_faible

impersonnel

indŽfini

relatif

possessif

exclamatif

prŽposition *No feature of SsCatMorph*

conjonction coordination

subordination

interjection *No feature ofSsCatMorph*

particule *No feature of SsCatMorph*

e *Any values authorized*

The values of syntactic sub-category features also depend on the category of the phrase. GENELEX proposes a set of sub-category features that may be extended by the user. It is possible to express constraints of integrity on the features proposed.

**EtiquetteSynt Authorized values of SsCatSynt:**

**---------------- -----------------------------------------------**

verbe copule

nom dŽterminatif

adjectif comparatif

superlatif

adverbe temps

lieu

mani•re

quantitŽ

degrŽ

comparatif

superlatif

dŽterminant *No feature of SsCatSynt*

pronom *No feature of SsCatSynt*

prŽposition temps

lieu

mani•re

conjonction *No feature of SsCatSynt*

interjection *No feature of SsCatSynt*

particule *No feature of SsCatSynt*

P relative

subordonnŽe

complŽtive

exclamative

coordonnŽ

SV coordonnŽ

SN DET_VIDE

coordonnŽ

SADJ coordonnŽ

SADV coordonnŽ

e *No feature of SsCatSynt*

## 2.4: Possible conjunctions of Features

In all cases, a list of features borne by a terminal or non-terminal phrase must be interpreted as a "liste_et", i.e. each feature in the list adds a new constraint.

Therefore, the lexical, morphological, morpho-syntactic, syntactico-semantic and semantic features add to one another.

```
Ex: N[Lex:plupart]

[Nombre:SINGULIER]

[SsCatSynt:DETERMINATIF]
```

However all features cannot be together in the same list.

A list cannot contain two features of the same sub-type if the possible values for this sub-type rigorously exclude one another. This comment applies to all features, excepted the following:

- With the features **SsCatMorph** and **SsCatSynt**, one wants to be able to express restrictions such as:

```
SN[SsCatMorph:DEFINI][SsCatMorph:CARDINAL]

-> DEFINI applying to the DETERMINANT

CARDINAL applying to the ADJECTIF
```

- Since the features **Libre** and **Coref** have non exclusive values, a feature of the same sub-type can be repeated. The constraints introduced will add to one another. In the case of Coref, one has to be careful not to use values that are mutually exclusive, i.e. common to one of the sub-sets {I, NON_I}, {J, NON_J}, {K, NON_K}, or {L, NON_L}.

`Ex: SN[Coref:J][Coref:NON_I]` is licit

`SN[Coref:J][Coref:NON_J]` is not licit

- The feature **Bin** has values that exclude one another (PLUS and MOINS), but its name is free. It will therefore be possible to combine **Bin** features, provided they have different names.

```
Ex: N[animŽ:MOINS][concret:PLUS]

[massif:PLUS]
```

To express an alternative on the value of a feature, lets us remind that one has to go through the alternative of position occupants.

```
Ex: Pi = DET[SsCatMorph:DEFINI]

DET[SsCatMorph:INDEFINI]
```

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

P[Mode:INFINITIF][Prep:ˆ]
```

# IX: Transformations

For those who want to record transformations, this model offers different possibilities for coding these transformations: transformation between Usyns, between Descriptions and between Phrases. It is the lexicographer's task to estimate the level desired to record such a transformation and to apply the same rules of coding on the whole dictionary, so as to ensure consistency.

We can however give certain recommendations.

- Limit Transformations between Usyns

    - to semantically distinct uses for different syntactic structures: typically the transitive or intransitive form of neuter verbs.

E*x*: *Pierre* ***plie*** *la branche* (Pierre bent the branch)

*La branche* ***plie*** (The branch bent (over))

- to Usyns drawn from different Ums in relation of derivation

*Ex: Les ennemis* ***dŽtruisent*** *la ville*

(the enemies destroyed the town)

*la* ***destruction*** *de la ville par les ennemis*

(the destruction of the town by the enemies)

- Limit Transformations between Descriptions to different syntactic turns that have no fundamental difference of meaning (except the displacement of the theme from an argument to the other).

    Typically passivation:

*Ex: Jean* ***casse*** *la branche*

(Jean broke the branch)

*La branche* ***est cassŽe*** *par Jean*

(The branch was broken by Jean)

- Limit Transformations between Phrases to syntactic structures that have the same canonical order, where a phrase occupying a position is the transformed phrase of another phrase in a same position.

Typically pronominalizations:

*Ex: Il aime **Marie*** (He loves Marie)

*Il l'aime* (He loves her)

These examples are only clues and a lexicographer is free to code neutrality between Usyns, whereas another may code it between Descriptions.

# 1: Transformations between Usyns

## 1.1: Transformations between Usyns drawn from the same Um

Two Usyns drawn from the same Um can be connected by a named transformation link that makes it possible to connect two distinct turns right from the syntactic level

E*x*: *Pierre **plie** la branche* neuter verb

(Pierre bent the branch)

*La branche **plie***

(The branch bent (over))

## 1.2: Transformations between Usyns drawn from different Ums

Actually this type of transformation corresponds to the coding of a syntactic derivation, typically nominalizations. The transformation wording can be "nominalization" for instance.

Ex: coller *(Verbe)*

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

(the militant is sticking posters up)

collage *(Nom)*

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

*nominalization*

Usyn01 --------------------> Usyn02

Besides, Ums can have several syntactic behaviors. In that case, the transformation link will connect the Usyns concerned, and will permit to highlight the different Usyns.

Let two morphological entries be connected by a derivation link (morphological), either having two syntactic behaviors:

Ex: `briller` (*Verbe)*

*Le jardin* ***brille*** *de lumi•re* (Usyn01)

(The garden is shining)

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

(This student is brilliant/outstandingly intelligent)

`brillant` (*Adjectif*)

*Le jardin est* ***brillant*** *de lumi•re* (Usyn03)

(The garden is bright with light)

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

(This student is brilliant)

Connecting their Usyns by a link of derivation transformation, it is therefore possible to draw a parallel between:

Ex: `Usyn01 -> Usyn03`

- *Le jardin* ***brille*** *de lumi•re*

- *Le jardin est* ***brillant*** *de lumi•re*

`Usyn02 -> Usyn04`

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

- *Cet etudiant est* ***brillant***

## 1.3: Calculation: Transformation Mode

Parallel to the static description of the original Usyn and that of the Usyn when transformed, it is possible to describe the calculation mode that has governed this transformation: for that purpose, one has to explicit the switching operations from the base description (db) of the original Usyn to the base description of the result Usyn, using the Transformation Mode.

This Transformation Mode is explained in section 2.3.

# 2: Transformations between Descriptions

## 2.1: Use and use context

A transformed description (**dt**) results from the application of a transformation operation on the base description (db). This operation can be formalized by a calculation method. Transformations between descriptions are borne by the Usyn and operate between the Descriptions of this Usyn. The base description can be the origin of one of these transformations, but it can also be a transformed description of this Usyn, which allows to express serial transformations.

Thus we can consider that the transformed description can be deduced from the base description to which the transformation is applied.

Transformations give account of three phenomena:

- transformation on Self realizations associated with an external construction (IntervConst).
- transformation on the internal structural phrase describing a compound.
- transformation on the construction.

## 2.1.1: Example of passivation

The passive voice is a typical transformation that we want to represent in this way. In that case, the transformation applies to the Self **and** to the construction.

*Ex: Jean **casse** la branche* (Jean broke the branch)

*La branche **est cassŽe** par Jean* (The branch was broken by 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: Example of noun clause restructuring

Transformations at this level can be used to code more complex phenomena of phrase restructuring, such as the reduction of a noun clause into an infinitive clause. This transformation affects the Construction.

*Ex: Ca* ***emp•che*** *que le lait ne dŽborde*

(It prevents milk from overflowing)

*Ca* ***emp•che*** *le lait de dŽborder*

(It prevents milk from overflowing)

*Ca l'****emp•che*** *de dŽborder*

(It prevents it from overflowing)

*Ca l'en* ***emp•che***

(It prevents it from it)

**Ca en* ***emp•che*** *le lait*

(It prevents milk from it)

```
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: Inheritance of positions: observations

Knowing that a position is defined by three fields: function, RoleTh (thematic role), and distribution, between a db and a dt, a sub-set of these fields is inherited for each position of the dt:

1. The function is not inherited, the distribution is not systematically inherited (we only mention here one of the phrases that is an element of the distribution).

Passivation is an example:

*Ex: le vent* ***casse*** *la branche* (The wind has broken the branch)

```
PO[Fonction:SUJET]: SN
```

```
P1[Fonction:OBJET_DIRECT]: SN
```

*a branche* ***est cassŽe*** *par le vent*

(the branch was broken by the wind)

```
PO[Fonction:SUJET]: SN
```

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

2. On the contrary, thematic roles must be inherited, apart from the theme. Nevertheless, the representation chosen authorizes thematic roles to be modified.

*Ex: le vent* ***casse*** *la branch* (the wind has broken the branch)

```
PO[RoleTh:AGENT]: SN
```

```
P1[RoleTh:THEME]: SN
```

*a branche* ***est cassŽe*** *par le vent*

(the branch was broken by the wind)

```
PO[RoleTh:THEME]: SN
```

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

## 2.3: Calculation: Transformation Mode

Whatever the type of transformation (transformation of Self, transformation on the structural phrase describing a compound or transformation on the construction), it can be described by an algorithmic transformation mode.

If the transformation concerns a Construction, it will be possible, if one so wishes, not to record the result Construction and have all the information relating to the transformation borne on the transformation mode.

## 2.3.1: Explicitation of the process

The transformation mode consists in:

- identifying a skeleton: it is a construction whose position occupants can be partially specified.

Ex: `P1: SP[Prep:par]` for a passive skeleton in "par" (by)

- covering this skeleton with elements or attributes coming from the original construction or specified in the transformation.

The pair skeleton + covering allows to entirely deduce the result construction that is explicitely given in the transformed description.

It is possible to express the way each position is affected by the transformation, selecting a position of the "skeleton" of the transformation applied. The process is then called position Transformation. A transformation mode is thus a set of position transformations. Beside the distribution modifications, one has also to indicate the characteristics of the construction modified by the transformation: option, interdependency, syntagmatic label

The chosen position is then "enriched" with attributes and elements inherited from the positions of the original construction or redefined by the transformation. The correspondence between the position of the transformed construction (result) and the position of the base construction (origin) is thus expressed. A position is selected by reference to its rank and is submitted to the modifications desired on its distribution. The phrases that are inherited, inhibited, modified or created for that purpose are specified.

If the transformation concerns the internal structural phrase of a compound, the transformed phrase has to be fully documented. The transformation mode describes the shift from the original structural phrase to the transformed phrase.

The transformation mode of Self as intervening in the construction indicates the modifications done on IntervConst during the transformation.

## 2.3.2: Exemplification of the process

We will re-use the preceding example on passivation to illustrate this process:

*Ex: Jean **casse** la branche*

(Jean broke the branch)

*La branche* ***est cassŽe*** *par Jean*

(The branch was broken by 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]
```

**For Self:**

- **adjunction of the feature** `[Passif:PLUS]`

**For the Construction:**

- **selection of the skeleton**:

```
P0 SELF P1

P1: SP

SP: P0' P1'

P0': PREP[Lex:par]
```

- **covering:**

```
P1(origine) ==> P0(resultat)

Fonction: OBJET_DIRECT ==> SUJET

RoleTh: PATIENT ==> (hŽritŽ)
```

```
Occupants: ==> (hŽritŽ)

P0(origine) ==> P1(resultat)

Fonction: SUJET ==> OBJET_INDIRECT

RoleTh: AGENT ==> (hŽritŽ)

Occupants: SN ==> SP

SP: P0' P1'

P0':PREP[Lex:par]

P1': SN(hŽritŽ)
```

# 3: Transformations between Phrases occupying the same position

If necessary, we want to be able to express transformation relations between occupants of the same position.

For instance, we want to be able to indicate whether a noun clause is in a transformation relation with the infinitive clause, or that a pronoun can be the pronominalization of only a sub-set of distribution on the position. The two phrases concerned are then connected with one another.

*Ex: il aime* ***Marie*** (he loves Marie)

-> *il* ***l'****aime* (he loves her)

*il aime* ***le chocolat*** (he is fond of chocolate)

-> *il* ***l'****aime* (he is fond of it)

-> *il aime* ***•a*** (he is fond of that)

*il aime* ***dormir*** (he likes sleeping)

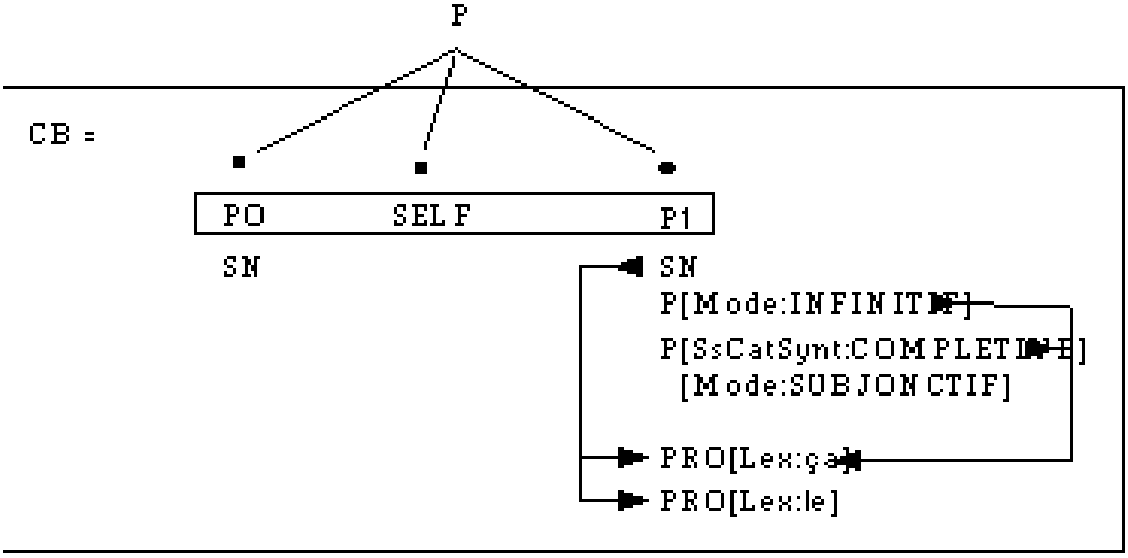
-> *il aime* ***•a*** (he likes that)

*il aime* ***que je vienne*** (he likes me to come)

-> *il aime* ***•a*** (he likes it)

```
P1: SN

PRO[Lex:le]

PRO[Lex:•a]
```

```
P[Mode:INFINITIF]

P[SsCatSynt:COMPLETIVE]

[Mode:SUBJONCTIF]
```



# C - Analysis of compound syntactic units

## I. Introduction

## 1. Recording of compounds

Compound expressions, whatever their type, are those a **native speaker** "knows" without having the feeling that he/she has learnt them once, and that he/she "spontaneously" **uses** (more or less well, besides...). They are fundamentally those a **foreign speaker** must **learn** to master a language perfectly. This comment applies to French and to any other language.

To be known until they may be automated, these expressions must a fortiori be given to the machine, i.e. the language processing program (especially in a context of generation). It is therefore important an electronic dictionary can record them.

However, recording fossilized expressions in a dictionary does not fossilize **language**, but on the contrary, it allows to give a reference point to everything that is considered as defossilization, a game on language.

## 2. Electronic dictionaries, editorial dictionaries

De facto, electronic dictionaries do not depend on the "paper format" specific to editorial dictionaries or on any constraint relating to this format. This is a considerable advantage for compounds. Actually, compounds may take an autonomy of existence which is not without consequences on:

• the recording of properties, since it is possible to associate an unlimited number of the finest and most complex properties with compounds.

- the mode of consultation. Actually, it is possible to access a compound:

  - either by any chosen full compound. Thus the compound no longer depends on one or several entries of simple words, more or less at random for novices. In DNT Hachette and Petit Robert, *mettre en oeuvre* (to implement) may be found under the entry *mettre*, whereas *mettre en marche* (to put into work) may be found under the entries *mettre* and *marche*.

  - or by the canonical form, even by one of the inflected forms of the full compound.

## 3. Coverage of compounds in GENELEX

We want to cover fossilized forms that are very different in nature: adverbial phrases (*ˆ l'avenant*), prepositional phrases (*ˆ l'instar de*), conjunctive phrases (*au moment o•*), compound nouns or verbs (*bec de gaz*, *mettre en garde*), idioms (*tomber des cordes*), set phrases (*glisser comme un pet sur une toile cirŽe*), proverbs (*pierre qui roule n'amasse pas mousse*), and maxims (*un geste vaut parfois tout un discours*, *dis moi qui sont tes amis, je te dirai qui tu es*).

In GENELEX, we have tried to clearly distinguish phenomena concerning lexical selection, collocation and composition.

**Lexical selection** (governed preposition, constrained impersonal subject pronoun, etc.) is dealt with in syntax by lexical features that constrain the realization of phrases occupying a position.

*Ex : il est facile de critiquer il* impersonal subject,

(it is easy to criticize) adjective complement:

infinitive clause introduced by *de*

**Collocations** are dealt with in the semantic layer that allows to connect different lexical entries on the semantic level.

*Ex : Žluder une question* relation "a_pour_objet" between

(to evade a question) *Žluder* and *question*

*meute de loups* relation "a_pour_collectif" between

(pack of wolves) *loup* and *meute*

For **composition**, we distribute the compound forms between "morphological compounds" on the one hand, and "syntactic compounds" on the other, which is the originality of our model.

*Ex : sage-femme* morphological compound

midwife

*bec de gaz* syntactic compound

(lamppost, gaslamp)

In GENELEX, **"morphological compounds"** are multi-lexical realizations of terminal categories that satisfy one of the following characteristics:

- lexical particularity: one of the components appears only in the complex expression (not necessary but **sufficient**).

- particularity in the written form:

    - separators (hyphen...) (not necessary but **sufficient**)

- inflection particularity

    - no respect of general agreement rules between components (not necessary but **sufficient**).

- syntactic particularity:

    - syntactic structuring that does not satisfy general rules in the sequence of components (not necessary but **sufficient**).

Morphological compounds have thus a category independent from the category of their constituents and of their eventual syntagmatic gathering.

**"Syntactic compounds"** are also complex elements, yet contrary to morphological compounds, they have the following characteristics:

- no graphic separator, except apostrophes and spaces,

- respect of general agreement rules between components,

- correct syntactic formation and, if applicable, properties of modification, displacement, deletion, insertion, substitution and transformation.

Syntactic compounds will be our single topic in this part.

# II: Characteristics of syntactic compounds

# 1: Well-constructed phrases

The "syntactic compounds" of GENELEX are well-constructed phrases (refer to synapses by [Benveniste 74]) partially or fully constrained (refer to TAGs by [AbeillŽ 84]): lexical, morphological (number, tense, etc.), syntactic and semantic constraints. As phrases, they are subjected to different regular syntactic operations such as agreement, deletion, displacement, modification, transformation and co-ordination.

While their inner structure is described by a non-terminal category phrase ("internal category"), on the contrary, their outer behavior may be compared to that of a terminal phrase ("external category").

We will have to describe both the "inside" and "outside" of the compound, although it is not always easy to draw a clear boundary between them.

## 1.1: Lexicalization

**• Total or partial lexicalization**

Syntactic compounds are fully or partially lexicalized phrases. All the constituents of the syntactic compound are not necessarily lexicalized, and one or several constituents may have a free lexical realization on one, or even several constituents. These free constituents are therefore more or less numerous and they can indifferently concern a terminal or a non-terminal phrase.

*Ex : carte bleue* (total lexicalization) *(12)*

(Visa card)

*Žprouver* DET SADJ *admiration* (partial lexicalization) *(15)*

(to feel DET SADJ admiration)

**Note**: all the examples in this part followed by a reference number in brackets are explained under the same number in part D (Annex to the Compound Syntactic Units).

-

**• Continuous or discontinuous lexicalization**

The lexicalization of a phrase can be continuous or discontinuous, so that lexicalized constituents and free constituents may alternate in the canonical linear order.

• continuous lexicalization

*Ex : carte bleue (12)*

• discontinuous lexicalization

*Ex : Žprouver* DET SADJ *admiration (15)*

N *and* N*, •a fait deux*

ˆ N *Žgal*, N *Žgal*

The compound does not have any "lexicalization core" around which free elements would revolve. What we have to describe is a phrase, some constituents of which, any of them, in any number and in any order, are lexicalized.

## 1.2: Inflection and agreement

As well-constructed phrases, syntactic compounds are submitted to inflection and agreement. Thus constituents are submitted to the morphological variations specific to their category: variation in person, gender and number for determiners; variation in mood, tense, person, gender and number for verbs.

*Ex : Il* ***est ˆ l'/son aise*** *(2)*

(he is /feels at ease)

*Ils* ***sont ˆ l'/leur aise***

(they are /feel at ease)

*Il* ***a mis*** *les invitŽs* ***ˆ l'/leur aise***

(he put the guests at (their) ease)

*Ils* ***ont mis*** *l'invitŽ* ***ˆ l'/son aise***

(they put the guest at (his) ease)

## 1.3: Realization alternatives

Compounds have not always a single realization, and sometimes there may have lexical or syntagmatic realization alternatives on a same compound. Variations in preposition and support verb (if we use syntactic compounds to record the pairs Vsupport-N) fall into this category. Variations on lexical classes and sub-classes defined by their extensive list, as well as any more complex syntagmatic alternatives, though much rarer, are also concerned. Whether they are terminal or non-terminal, phrases that are considered to represent realization alternatives fit into a substitution paradigm.

- variation in the preposition

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

(projection equipment)

- variation in the support verb

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

(to feel/have/be filled with/be lost in admiration)

- extensive class

*Ex : Žprouver une* ***grande/ immense/ Žnorme...*** *admiration (15)*

(to feel a great/huge/tremendous admiration)

*Žprouver une admiration* ***inconsidŽrŽe***

(to feel boundless admiration)

- extensive sub-class

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

(with full knowledge of the facts)

- syntagmatic variation prepositional phrase /adjective phrase

*Ex : moules* ***ˆ la (mode) marini•re*** *(5)*

*moules* ***marini•re(s)***

## 1.4: Deletion

"Elided" forms of compounds are well known. They are connected with the option of certain constituents that are not necessarily realized in surface. Although noun elision is the most typical and regular, this behavior can be observed on any other category. Deletion can therefore apply to a specifier, a modifier or the head itself.

- deletion of the head

*Ex : fromage de ch•vre* (goat's milk cheese)

*ch•vre* (goat)

- deletion of the noun as the head or embedded element

*Ex : fil de fer barbelŽ* (barbed wire) *(3)*

*fil barbelŽ*

*barbelŽ*

- deletion of the specifier

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

(money is the root of all evil)

*L'argent est cause de tous les maux*

(money is the root of all evil)

- deletion of the modifier

*Ex : crouler sous les applaudissements de* SN

(to collapse under applause from)

*crouler sous les applaudissements*

## 1.5: Modification

Since they are structured like ordinary phrases, certain syntactic compounds accept all types of modification that apply to the phrase that describes their internal structure. Modification may be optional, obligatory, and have free or constrained realizations.

*Ex : Žprouver de l'admiration*

(to feel admiration)

*Žprouver une* ***grande/immense/Žnorme/...*** *admiration (15)*

(to feel a great/huge/tremendous admiration)

*Žprouver une admiration* ***inconsidŽrŽe/sans borne***

(to feel a boundless admiration)

*Ex : en connaissance de cause*

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

(with full knowledge of the facts)

## 1.6: Displacement

Contrary to one could think in the first place, the fossilization induced by lexicalization does not necessarily imply fossilization in the surface linear order. Actually, it is possible to observe displacements or permutations of constituents that induce variations in the surface order. This also applies to the inside and the outside of the compound. In that case, this will be a phenomenon of re-entering (refer to the paragraph on the boundary between the inside and the outside of a compound).

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

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

(to warn)

Consequently, it is once more necessary here to distinguish the canonical order from the surface order; to do so, we use the notion of position. Positions will indicate the canonical order. The disruption of surface linearity (pronominalization, deletion, freedom to position SNs-SPs) will be part of a system of grammar, even stylistic rules.

## 1.7: Transformations

Sometimes (but not always), syntactic compounds accept transformations (restructuring such as reduction of the noun clause into an infinitive clause are a more complex particular type of such transformation) that will apply to the phrase describing them and to their construction. Such transformations are of different types: possessivation, demonstrativation, pronominalization, relativation, passivation and even syntactic derivation one of the components.

- possessivation

*Ex : au grand dam* ***de*** **SN** (to the great displeasure of )

ˆ ***son*** *grand dam* (at his/her great displeasure)

- demonstrativation

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

(at the time of)

ˆ ***ce*** *moment*

(at that time)

**ˆ son moment*

(at his/her time)

- relativation

*Ex :* SN *livrer* ***bataille*** ˆ SN

(to do/join battle with)

***la bataille que*** SN *livrer* ˆ SN

(the battle that SN is doing/joining with SN)

SN *donner libre cours* ˆ *son imagination*

(to give free rein to one's imagination)

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

(the free rein SN gives to one's imagination)

- passivation

*Ex :* **SN*1*** *mettre en garde* **SN*2*** *(10)*

(SN**1** warn SN**2)**

**SN*2*** *•tre mis en garde par* **SN*1***

**SN**2 be warned by **SN**1

**SN** *prendre son pied*

**son pied •tre pris par* SN

- derivation on a head component

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

(to settle)

***mise*** *au point*

(settling)

*prendre en grippe*

(to take a sudden dislike to)

**la prise en grippe*

- derivation on another component

*Ex : festival* ***de musique*** (music festival)

*festival* ***musical*** (music festival)

*machine ˆ laver* (washing machine)

**machine ˆ lavage*

## 1.8: Coordination

Like phrases, syntactic compounds may also be coordinated or contain coordinated phrases. It must therefore be possible to specify the coordinating element as well as the coordinated elements.

*Ex : ˆ ses risques et pŽrils* (at one's own risks) *(7)*

*mŽnager la ch•vre et le chou* (to sit on the fence)

# 2: Fossilization

## 2.1: Total or partial fossilization

Although a syntactic compound is a regular phrase, it differs from it by the number of constraints it bears, especially the importance of constraints on the lexical realizations of its leaves.

All these constraints are fossilization constraints, i.e. the more the constraints, the more fossilized the phrase. This is the reason why we can talk of **fossilization continuum** from the totally free to the totally fossilized.

Motivations that govern the recording of compound forms differ according to whether the objective is to build a general language dictionary or a specialized dictionary like a terminological base. In the first case, purely linguistic fossilization constraints prevail. On the contrary, in the second case, practical criteria like the frequency or the hierarchical level of a concept will govern recording.

## 2.2: Interdependent constituents

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

Certain modifiers, appositions, interpolated clauses, floating quantifiers, etc. are elements that share the characteristic to be inserted in numerous places in a sequence of constituents anf for which the occurrence in the compound surface form has not to be provided for in the lexicon.

However, in the case of syntactic compounds, such interpolated clauses are not completely free. Certain constituents of syntactic compounds are interdependent, i.e. no insertion between them is allowed: they are an undividable sequence.

*Ex : bo"te aux lettres, disait-il* (mailbox, he said) *(16)*

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

# 3: Inside and outside of the compound

## 3.1: Inclusion in a terminal category

Although they are described as non-terminal category phrases, syntactic compounds, as units, can be classed as terminal categories, hence the designation "multi-lexical heads".

Such inclusion is done on several bases:

• **syntax**

Although a form is complex, it can play the role of a head, call complements, restrict its occurrence context. For instance, a compound structured as a verb phrase behaves as a verb in its construction and is therefore given this (external) category as an intervener in the construction.

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

V

(SN warn SN against SN)

In broader terms, the compound invariably occupies the same functional position (noun specifier, sentence modifier, etc.) that has a terminal category as archetype (determiner, adverb, etc.) in which one tends to class it ("complex determiners", "adverbial prepositional phrases", etc.).

- SV classed as V *(mettre son poing sur la gueule // frapper* )

(to get one's fist in somebody's face/to hit)

- SP classed as CONJ *(en consŽquence // donc)*

(in consequence//consequently//therefore)

- SP classed as ADJ *(sans vergogne // indŽlicat)*

(shameless/tactless)

- SP classed as ADV *(ˆ ses risques et pŽrils // imprudemment)*

(at one's own risks//carelessly)

- SN classed as DET *(la plupart des // les)*

(most of the //the)

- Nbarre classed as N *(fil de fer // c‰ble)*

(wire)

Finally, and in the same way as a simple unit, a compound may have a set of restricting properties on its own.

*Ex : ˆ ses risques et pŽrils [nombre:plur] (7)*

(at one’s own risks) *RoleTh:mani•re*

**• semantics**

The semantic non-compositionality of certain syntactic compounds (not required to consider them as compound syntactic units) gives them a status of undividable whole.

Although it is not always possible to decide on the semantic compositionality or non-compositionality of an expression (fossilized expressions with an etymological compositionality, metaphoric expressions, meaning of support verbs), some expressions, however, cannot clearly be composed in synchrony.

*Ex :* SN ***se rendre compte*** *de* SN

V

(to become aware of)

## 3.2. Distinction between the inside and the outside of a compound

The phrase label that describes the internal structure of a compound is as it were its **internal category**. The terminal category preferably designating the functional position occupied by the compound is its **external category**. The notions of inside and outside of the compound are based on that distance between the two labels. The compound outside is the complementation or the occurrence context of the compound. The compound inside is the phrase describing its structure. When we talk about multi-lexical head, we refer to the external behavior of a unit whose particularity is to have a composite inside and not an atomic one.

## 3.3. Boundary between the outside and the inside of a compound

It is not always easy to draw the boundary between the inside and the outside of a compound for two reasons:

- since the compound is a partially lexicalized phrase, some of its constituents are "free" (not lexicalized), in the same way as most of its external complements. The fact that a phrase (in the dependency of a phrase) is "free" does not indicate its belonging to the inside or outside of a compound, even if it depends on the compound head.

*Ex :* SN ***Žprouver*** **DET SADJ** ***admiration*** *for* SN *(15)*

- we observe re-entering (or insertion) phenomena between the inside and the outside of the compound: in other words,. in surface an **external** complement may appear **inside** the compound. Consequently, the surface order does not indicate the internal or external connection of a constituent.

*Ex :* SN ***tirer*** SN ***au clair***

(to clear something up/clarify)

SN ***prendre*** SN ***en compte***

(to take into account)

SN ***mettre*** SN ***en Ïuvre*** *(6)*

(to implement)

It is a real lexicographic difficulty. Therefore it is the lexicographer's responsibility to decide which coding strategy to implement, to evaluate the relevance and linguistic scope in order to ensure consistency. This problem is particularly acute for all traditional "phrases": prepositional or conjunctive phrases that can be affected by possessivation or demonstrativation.

*Ex : <u>au moment o•</u> P (11)*

*ˆ ce moment*

*<u>au dŽtriment de</u> SN (9)*

*ˆ son dŽtriment*

# 4. Inheritance of the syntactic properties of components

We have already mentioned that the compounds under study (that are not fossilized enough to be morphological compounds) are regular phrases presenting a certain degree of fossilization; otherwise, they would be considered as asyntactic. These well-constructed compounds are **realizations authorized by the syntactic behaviors of their different components**.

Thus, like well-constructed phrases, syntactic compounds obey the general rules of syntax, so that the sequence of free and lexicalized components is a licit sequence.

Consequently, components fit into the syntactic compound with their first property, i.e. their category.

Besides, the compound inherits the syntactic behavior of its components, with or without addition of restrictions during the composition process; besides, this process determines the way its components will be organized. In that case, the compound determines the "composition mode" of its components.

As for inheritance, the number of restrictions to be added during composition is sometimes such that it is preferable not to inherit the syntactic behavior of a component.

Each composition mode describes the inheritance by the compound of the syntactic behaviors of a component. Inheritance concerns the construction of the component, the category and restricting feature that characterize it as an intervener in the construction. The construction is inherited inside, and sometimes outside the compound. The characteristics of the component are inherited inside the compound.

## 4.1. Inheritance of constructions

These inheritance properties apply to each Syntactic Unit (Usyn) entering in the composition of the compound.

*<u>• Total inheritance versus partial inheritance</u>*

The less the component keeps properties when it is used in a syntagmatic composition, the more fossilized the result compound. The term "total inheritance" is used when all the properties of the component are inherited without changes. The term "partial inheritance" is used when at least one property of the component is inhibited or filtered. Some positions of the component may be totally inhibited whereas others will be maintained, yet filtered by the selection of a sub-set of phrases that can instantiate it: addition and loosening of constraints on these phrases (a determiner that becomes a definite, a free number restricted to plural, etc.) are some illustrations.

*Ex : Il a abattu sa toute derni•re carte ma"tresse, qui est excellente. (13)*

(He laid his last master card on the table, that is excellent)

DET SADJ ***carte*** *SADJ*/SP/P => inhibition of the left- positioned modifier,

selection of a sub-set of phrases on the right-positioned modifier.

*Ex : tarte ˆ la cr•me (14)*

DET SADJ ***tarte*** *SADJ/SP/P* => inhibition of the left- positioned modifier,

selection of the SP on the right- positioned modifier and restriction to the preposition ˆ.

DET SADJ ***cr•me*** *SADJ/SP/P* => inhibition of the left- and right- positioned modifiers, restriction to the definite determiner.

In the case of important fossilization, some components, even all of them, may be deprived from all their characterisitics. Such is the case of *bleu* (blue) in *carte bleue* (Visa card) that is deprived of all its modification possibilities: *une carte tr•s bleue* (a very blue card) is not the same as *une carte bleue*. In that case, it does not seem justified to refer to a component Usyn that would be completely inhibited. It will be possible to refer to the Um (Morphological Unit).

Adjectives may be used as another example. It is interesting to note that it is the adjective that holds knowledge on its positioning on the left or on the right of the noun in such or such use and that this knowledge is inherited in the compound.

*Ex : temps plein* (full time)

*plein temps*

*temps partiel* (part time)

**partiel temps*

*• Inheritance outside the compound*

The compound taken as a terminal category inherits in its external construction of all or part of the base construction of the component that is exclusively the compound head. Therefore the external construction of *carte bleue* is the same as that of *carte* (in a syntactic viewpoint, not in a semantic one) and not of *bleu*.

*Ex : La tr•s belle* **carte bleue** *dorŽe du CrŽdit Lyonnais. (12)*

(The very nice gilded **Visa Card** of CrŽdit Lyonnais)

*• Inheritance inside a compound and saturation*

The compound taken as a non-terminal phrase inherits in its inside complementation patterns that are specific to each of its component Usyns. That is a misnomer because more exactly each component (and not the compound) inherits, when it is used in a composition (in the internal structure of the compound) of the complementation pattern that is associated with it as a free unit.

Some positions of a component are said "saturated" during lexicalization by another component: these positions inherited inside the compound are not found outside. It is the case, for instance, of *abattre carte maitresse. Carte maitresse* saturates the object position of *abattre* inside the compound, which is a position not inherited in its external construction.

*Ex :* SN ***abattre carte maitresse** (13)*

## 4.2. Inheritance of the characteristics specific to the compound

In the compound, components keep all or part of their specific characteristics.

Thus, one of the syntactic units of *dŽrive* is characterized, among others, by a restriction to the singular. In the syntactic compound, *dŽrive gŽnŽtique,* the component *dŽrive* keeps this property. In the same way as for verbs, the auxiliary that is defined on the verbal compound is inherited in the compound. Consequently *abattre carte ma"tresse* inherits the auxiliary *avoir* from *abattre*.

# 5. Support verbs

Compound Syntactic Units can also be used to record - if one so wishes - the pairs "support-verb/noun" so as to specify their complementation pattern, and have them correspond more easily - if neccessary- with a verb (derived or not from the noun) that is their true synonym (apart from the aspect).

*Ex : SN* ***donner/poser sa dŽmission*** *ˆ SN*

(to hand in one's resignation to SN)

*SN* ***dŽmissionner*** *de SN*

(to resign from SN)

*SN* ***mettre en marche*** *SN*

(to start up)

*SN* ***dŽmarrer*** *SN*

(to start up)

*SN* ***•tre/se mettre en col•re***

(to be/get angry)

# III.Principles of formal representation

## 1. Representation axes

The simple or compound nature of an entry does not affect the recording of a **Construction** specifying its complementation or its context. Compound syntactic units are therefore described as simple syntactic units as regards their external behavior.

What is specific to them is the fact that the described entry (**Self**) is not atomic but composite (multi-lexical).

Composition in syntax is represented by a list of lexicalized components and information relating to their articulation. This information is taken into account by several formal elements (**Composition** + **R_Compose, MdC, Syntagme_NT_S**) that are going to be explained.

Lists of components (**Composition** and **R_Compose**) manage the multi-lexical aspect. The Composition Mode (**MdC**) and the structure phrase manage respectively the syntagmatic regularity as regards the components (by inheritance of properties) and as regards the grammar (rewritten non-terminal phrases). Both may co-exist to complete one another since the approach angles are different but do not exclude one another.

What formally distinguishes compound syntactic units from simple syntactic units is only:

- the presence of lists of components on the Usyn,
- the presence of Composition Modes of the compound Usyn from each of the component Usyn,
- the presence of an internal structure on the **Self** of the Usyn**.**

On the other hand, they are described by the same formalism as simple syntactic units (**Description**).

The articulation between the external syntactic behavior of the compound and the description of its internal behavior is drawn using the entity **Self** that is present both on simple and compound units.

## 2. Atomist viewpoint vs. syntactist viewpoint

The GENELEX model meets the concern to be open to the two different representations of the lexicon, the atomist vision and the syntactist vision (refer to the section on simple syntactic units for the definition of these visions).

However, rewriting a structure phrase is only meaningful in the syntactist vision. In the atomist vision, the specification of a phrase describing the inside of the compound will be reduced to its simplest expression, i.e. its syntagmatic label.

Besides, Composition Modes (**Mdc**) are used to describe and record the inheritance of properties from the components. So defined, they accept both interpretations, although their use is optional in the syntactist interpretation whereas it is obligatory in the atomist viewpoint.

# IV. Lists of components

## 1. Type and choice of components

Components that lexicalize the internal structure of Compound Syntactic Units are explicitely listed on the Usyn. Components are either of the Usyn or Um type. A component is of the Usyn type when we want it to inherit, totally or partially, from its syntactic behavior. Otherwise the component is of the Um type.

The list of components allows to reach the compound, since the compound is indexed on the elements of this list. This list must therefore minimally include all the components that "sign" the compound, i.e. the Usyns we want to inherit from and so-called "full" Ums.

For other lexicalized constituents, one can:

- include them in components as "empty" Ums;

- not include them in the list of components and constrain them as usual, using the features Lex in the MdC and/or the internal structure. This is very useful for:

- the partitive deteminer {*du/de l', de la, des*} to which the notion of canonical form cannot be applied, because of the ambiguity of the forms *du* and *de l'*. One may prefer:

*((Žprouver admiration))* + `DET[SsCatMorph:PARTITIF]`

to

*((Žprouver du admiration pour))*

- non-governed prepositions that we want to record in the internal structure and not in the list of components. One may prefer:

*((bec gaz))* + `PREP[lex:de]`

to

*((bec de gaz))*

In any case, it is the lexicographer's responsibility to establish a coding strategy and to hold on to this decision.

With the list of components that can be refered by their rank from the description of the structure and in the composition mode, it is possible to share the pieces of information that only differ from one another by different lexicalizations, since they are only connected with them indirectly by their rank. It is thus possible in a first time to make an inventory of the structures (that will have the *appellation* NA, AN, NdeN, NˆN, PN, VN, NN, etc for noun compounds). Then we will list all the compounds that match them, i.e. all compounds that lexicalize these structures.

*Example of internal structure Noun Adjective.*

**INVENTORY OF INTERNAL STRUCTURES**

**Internal structure: designation: "N_ADJ"**

**P0 P1**

**NOM[RefLex:1.1] ADJ[RefLex:1.2]**

**INVENTORY OF COMPOUNDS**

((voie lactŽe))

((carte bleue))

((b•te noire))

etc

## 2.2. Alternatives of components

To record variants of lexicalization, we use two levels of lists.

*Ex: ( (avoir admiration) (15)*

*(Žprouver admiration)*

*(ressentir admiration)*

*(vouer admiration) )*

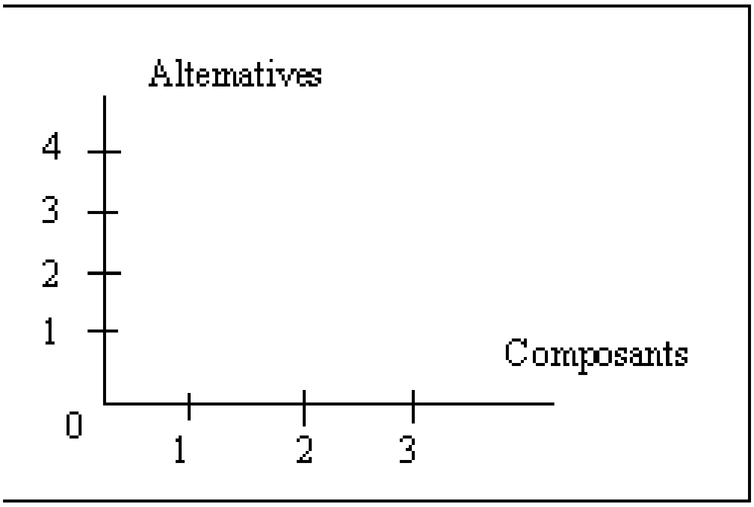
*((ˆ le cas o•) (11)*

*(en ce cas))*

The first level of the list is that of lexicalization alternatives (**Composition**).

The second level of the list is that of lexicalized components for each alternative (**R_Compose**).

These lists can be seen as two orthogonal axes.



## 3. Reference to components

The elements of the lists of components are accessed by double indexing: [RefLex: Nth alternative . Nth component].

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

Index 0 on the alternatives axis means that all of them are selected.

```
Ex: [RefLex:0.1]--->avoir,Žprouver,

ressentir,vouer
```

> **Note**: In the appended examples, we have decided to explicitely indicate lexicalization to ensure a certain level of legibility: [RefLex:mot], rather than using the notation: [RefLex : Ni•me alternative . Ni•me composant]. This comment applies to the whole family of RefLex and includes RefPrep, RefConj, etc.

# V. Composition Mode (MdC)

## 1. Interest

Composition Modes allow to specify for each Usyn component the properties of its Construction and of its Self (IntervConst and Syntagme_NT_S) that are inhibited, filtered or inherited during composition. They also allow to indicate what other components (Um or Usyn) eventually lexicalize their Positions. Therefore, for a given compound Usyn, there are as many MdCs as Usyn components.

> Specifying the composition mode is therefore used to indicate:
>
> • the **relations** between component Usyns and/or Ums,
>
> • the **inheritance** of Positions,
>
> • **addition** or **loosening** of constraints on Position occupants and Self.

## 2. Inhibition, inheritance and filtering

## 2.1. Inhibition, inheritance and filtering of Positions

Unless otherwise specified, all the Positions of a component that are not explicitely inherited are inhibited. For each inherited Position, we have to specify how and where it is inherited, i.e. with or without filtering, INSIDE or OUTSIDE the compound. A same Position, which is then not "saturated" in the compound, can be inherited outside **and** inside the compound. In that case, it is associated with two inheritances of Position.

Filtering a Position during inheritance consists in modifying at least its distribution, its function, its thematic roles or its repeatability value. Distribution is modified in selecting a sub-set of listed phrases or in removing/adding constraints on these phrases (restricting features).

## 2.2. Self filtering

Unless otherwise specified, Selfs of component Usyns are inherited inside the compound, knowing that, in that case, they lose the designation "Self" to become a phrase such as specified in IntervConst. It is possible to filter Self in the composition process and, using the same mechanisms, to modify its IntervConst and internal structure if the component is itself a compound.

## 2.3. Example

*Ex : la vraie **tarte ˆ la cr•me** du cinŽma fran•ais*

(the real slapstick of the French cinema)

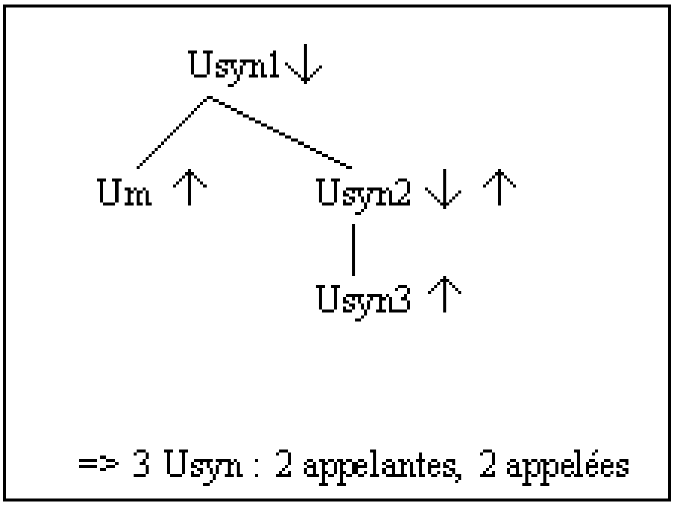
```
Usyn_tarte :

cb : P0 P1* SELF P2*

DET SADJ SADJ

SP

P[SsCatSynt:RELATIVE]

MdC_tarte in tarte ˆ la cr•me:

HeritePosition : P0 EXTERIEUR

HeritePosition : P1 EXTERIEUR

HeritePosition : P2 EXTERIEUR

HeritePosition : P2 INTERIEUR

ModifPosition :

SADJ, P[SsCat_Synt:RELATIVE]

SP + [Prep:ˆ]

[RefLex:cr•me]

[Nombre:SINGULIER]

[SscatMorph:DEFINI]
```

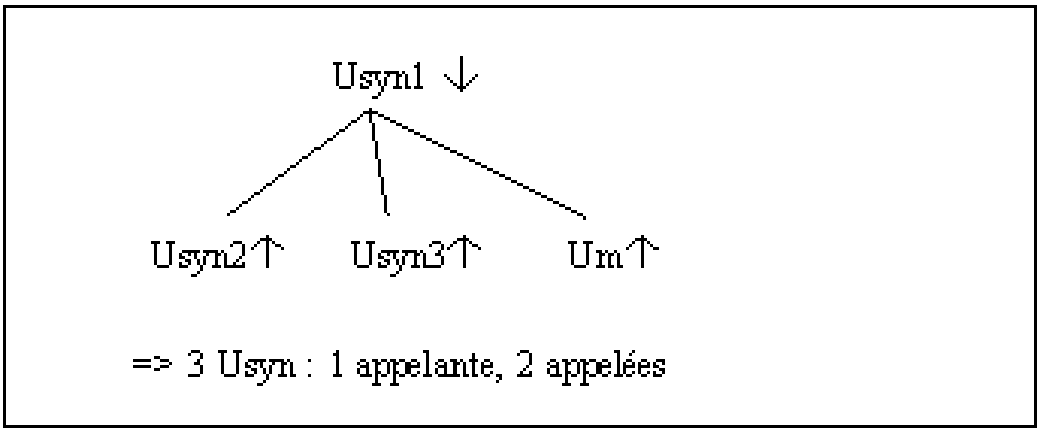
```
FiltreSelf

IntervConst : N + [nbre:SINGULIER]
```

# 3. Calls of components for lexicalization

## 3.1. Notion of call

The organization of components, i.e. the fact that they are organized according to different hierarchic levels and that some of them depend on others (and vice versa), is analyzed as a **call** phenomenon. A component is said to be called if it instantiates a Position of another component. A component is said to be calling if at least one of its Positions is instantiated by another component. A same component can be calling, called and calling, or only called. This information is borne by the elements **R_Compose** that give the list of components.





Let us illustrate these abstract call trees with an example.

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

(to show one's master card)

```
2 Usyn appelantes : Usyn_abattre, Usyn_carte

(calling Usyn)
```

```
1 Usyn appelŽe : Usyn_carte
```

```
(called Usyn)
```

```
1 Um appelŽe: Um_ma"tre
```

```
(called Um)
```

## 3.2. Head of the syntactic compound

The head of a compound Usyn is the most calling component.

In the preceding example, the "head" Usyn of the compound is the Usyn_abattre

For co-ordinated compounds, we can consider that the head of the compound is the co-ordinating element and that it is this element that calls all the other constituents.

*Ex : ˆ cor et ˆ cri*

## 3.3. Lexicalization

A called element realizes a lexical constraint. One expresses the call of a Um or Usyn component by adding a feature **RefLex** on a Phrase of a Position of the Base Construction of the **calling Usyn**.

```
Ex : SN[RefLex:Usyn_admiration]
```

The component gives its value to the lexical feature that calls it (RefLex, RefPrep, RefProRel, RefProIntrog, RefConj, RefIntrod) and the general rules specific to these features apply (see paragraph on restricting features). The attribute *saturesynt* of lexical features allows to indicate if the head is (with an introducer, if applicable) the only leaf node of the Phrase.

## 3.4. Compatibility between the categories of the calling phrase and those of the called component

We must ensure compatibility between the label of the calling phrase (the phrase bearing the feature RefLex) and the morphological or functional category of the called component (of the **Um** or **Usyn** type). In other words, it must be possible for the component to be the head of the calling phrase. The head values authorized are taken from simple units according to the phrase that bears the lexical feature:

- NOM for SN

- NOM for SP

- ADJECTIF for SADJ

- ADVERBE for SADV

- VERBE for SV

- VERBE for P

The morphological category of the component is given by the field *catgram* of the **Um**, whereas its functional category is given by the labels of phrases of the **Usyn IntervConst**. If the component is a **Usyn**, and since each Usyn is associated with a single **Um**, it is always possible to find the **Um** from the **Usyn**.

## 3.5. Unification

The Phrase that bears the lexicalization feature may also bear other restricting features. This applies whatever the Um or Usyn nature of the lexicalizing element. All the restrictions expressed on a called Usyn that would be reported on the calling phrase by a unification mechanism have to be brought up by hand" by the lexicographer, since our formalism does not manage this mechanism.

*Ex : Žprouver de l'admiration*

`Usyn_Žprouver :`

`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_Žprouver in` *`Žprouver de l'admiration`*`:`

`HeritePosition : P0 EXTERIEUR`

`HeritePosition : P2 EXTERIEUR`

`HeritePosition : P1 INTERIEUR`

`ModifPosition :`

`SN + [RefLex:admiration]`

**`[Nombre:SINGULIER]`**

`MdC_admiration in` *`Žprouver de l'admiration`*`:`

`HeritePosition : P0 INTERIEUR`

```
HeritePosition : P1 INTERIEUR

HeritePosition : P2 INTERIEUR

FiltreSelf

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

Finally, the syntactist vision (with rewrite trees) considers that for a component Usyn, the level that enters in composition is a priori the level of Self. If we want to retrieve constraints on the syntactic context of Self (greater than its strict complementation pattern), we have to explicitely indicate it by the inheritance of positions giving account of this context.

# VI. Internal structure of compound

## 1. Choosing the category

The internal category of the compound is borne by **Syntagme_NT_S** that describes the internal structure of **Self**. The external category of the compound is borne by the **Phrase**(s) of its **IntervConst**.

*Ex : bo"te aux lettres* (internal category Nbarre, external category N)

Internal and external categories must be compatible. For information, here are the pairs that we have identified during our work in GENELEX.

| **External category** | **Internal category** |
|---|---|
| NOM | NBARRE, SN |
| ADJECTIF | SADJ, 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. Structure Phrases (Syntagme_NT_S)

In the GENELEX model, the **internal** base structure of compounds is described in **Self** by a "non-terminal structure phrase" (**Syntagme_NT_S**). This phrase is characterized by its label, the features that are associated with it, and it can be rewritten by a list of positions, in the same way as all non-terminal phrases.

*Ex : la plupart* internal structure SN

(most)

*tarte ˆ la cr•me* internal structure Nbarre

(slapstick)

*pierre qui roule n'amasse pas mousse* internal structure P

(a rolling stone gathers no moss)

*mettre en Ïuvre* internal structure SV

(to implement)

*en connaissance de cause* internal structure SP

(with full knowledge of the facts)

*sain et sauf* internal structure SADJ

(safe and sound)

*plus ou moins* internal structure SADV

(more or less)

These phrases are perfectly identical to non-terminal Phrases that appear in Positions describing simple units, except the following:

- they do not contain any Self element, since what we want to describe through them is this element.

- they accept features of the RefLex family on Phrases occurring in their Positions,

- a possible list of Insertions adds to the list of Positions.

For the remaining part (tree rewriting, labelling of nodes, coordination, etc.), refer to the paragraph on Phrases in the section on simple syntactic units.

It is to be noted that syntactic compounds have an internal structure particularly subject to structural constraints. Structural constraints are rewrite constraints on a phrase. For some syntactic compounds, an element normally optional in this context will obligatorily realized or omitted, or even, a complex structure, such as coordination or the embedding of SNs, will obligatorily called.

*Ex : ˆ ses risques et pŽrils (7)* coordination

(at one's own risks)

*sans conteste* no determiner

(unquestionably)

*au dŽtriment de SN (9)* embedding

(to the detriment of)

*en lieu et place* coordination and no (in lieu and place of ) determiner

## 3. Insertions

Re-entering phenomena are represented by Insertions of Positions of the base Construction into the internal structure of the compound.

*Ex :* ***mettre*** SN ***en marche***

(to start up)

***prendre*** SN ***en compte***

*(*to take SN into account)

***mettre*** SN ***en Ïuvre*** *(6)*

*(*to settle)

As soon as an Insertion is recorded, one has to interpret the inner Positions as locations, which is in contradiction with the recording of a canonical order. Therefore Insertions break the distinction location/ position (surface linear order and canonical linear order. See Simple Units). Assuming that the canonical order is only one of the possible surface orders, with Insertion, it becomes the only possible.

Insertions of **Positions** of the external Construction are recorded in the right place in the field *inseresinsertion_l* among the **Positions** of **Syntagme_NT_S** figuring the internal structure. An **Insertion** is therefore both a pointer at a **Position** of the external **Construction** and an insertion point mark. There may have several **Insertions**. Just as **Positions**, **Insertions** are numbered from 0 on.

Certain **Insertions** are obligatory, whereas they are only possible in most cases.

- possible insertion

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

SN ***mettre en garde*** SN

(to warn someone)

- obligatory insertion

*Ex :* SN ***conna"tre*** SN ***comme si*** **PRO** ***l'avoir fait*** *SN ***conna"tre comme si*** **PRO** ***l'avoir fait*** SN

# 4. Interdependency

Elements that may be interdependent, i.e. they form an undividable sequence prohibiting any interpolation, apposition or floating quantifier, are **Position, Self, Insertion.**

This undividable sequence is represented in a field *solidaritŽ* within the compound by a graphical string with a dash between interdependent elements. It is therefore possible to have a *solidaritŽ* field with the value:

```
P0 I0-I2-P1
```

# 5. Restricting features

All restricting features defined by simple Units are available for compound Units. Some of them, however, need to be better explained.

# 5.1. Lexical features

The lexicalization of terminal or non-terminal Phrases entering in the description of the syntactic compound is expressed by lexical features borne by those Phrases. It is possible to re-use all the features defined for simple units, i.e. Lex, Prep, Conj, ProRel, ProIntrog, Introd. Moreover, the features RefLex, RefPrep, RefConj, RefProRel, RefProIntrog and RefIntrod are available in reflection. These "Ref" features have exactly the same semantics as their correspondents but they point at Morphological Units (Ums) or Syntactic Units (Usyns) not directly, but indirectly by rank numbers in lists of components (refer to the paragraph on the lists of components). This indirection has been introduced so as to be able to share the internal structures (N ˆ N, N de N, etc.) independently from their actual lexicalization. RefLex features are only borne by Phrases of the internal structure and are excluded from the external construction.

*Ex : voie lactŽe*

*carte bleue (12)*

*cerveau malade*

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

Note: in the annexes to this chapter, the values of the RefLex features are resolved to facilitate reading. The information will therefore be under the form N[RefLex:voie] ADJ[RefLex:lactŽ].

It goes without saying that one can point at a component by a feature of the RefLex family only if it has been recorded as a true "component" in the list of components.

On the other hand, all lexicalizing elements need not be dealt with in the list of components. It is possible to specify the prepositions and determiners (including partitive determiners) in the internal structure of the compound rather than in its list of components. In that case, the specifications are expressed like for simple units using the features of the "Lex" family (e.g. SP[prep:ˆ]) or using subcategory features (e.g. DET[SsCatMorph:partitif]).

## 5.2. Morphological features

All morphological features defined for simple Syntactic Units are available for the description of compounds. They allow, for instance, to express within the internal structure any morphological restriction on components, the most common restriction being that on number.

*Ex : prendre son pied* => compound form *(1)*

*prendre ses pieds* => free form

## 5.3. Accord and Coref features

Features of agreement in the broad meaning are dealt with exactly as for simple units. Regular agreements are left to the grammar, agreements based on coreferences are managed by the introduction of the Coref features in the appropriate places; irregular agreements are constrained by the introduction of an Accord feature.

We want to be able to express coreferences and agreements between Phrases inside and Phrases outside the compound. Consequently, the field of resolution of the Accord and Coref features is the whole Description of the compound Usyn (@External Construction + IntervConst and structure Phrase of Self).

*Ex : Il* ***est ˆ son aise*** *(2)*

(he is at ease)

*Ils* ***sont ˆ leur aise***

(they are at ease)

Form variations rather than form constraints of personal pronouns and determiners are usually a problem for lexicographers when recording compounds. In GENELEX, the gathering of the forms *{je, tu, il, elle, nous, vous, ils, elles}, {ma, ta, sa, mon, ton, son, mes, tes, ses, notre, votre, nos, vos, leur, leurs}*, considered as inflected forms, allows to rid oneself from that problem.

In cases of composition, variations of person, gender and number will be considered as inflection variations on a single lemma entering in the composition of the compound and they will be dealt with as a simple agreement phenomenon.

On the contrary, it is obviously possible to express, if necessary, morphological constraints on these lemmas, using morphological features on the phrases they lexicalize.

## 5.4. Semantic features

For compounds as for complements of simple words, all the semantic restrictions (for those who want so only) right from the syntactic level are the same:

- aspect constraint on the verb

*Ex : •tre*`[Aspect:STATIF]` *ˆ l'/son aise (2)*

*mettre*[`Aspect:PROCESSIF]` *ˆ l'/son aise*

- semantic class constraint

*Ex : pour les beaux yeux de* SN`[humain:+]`

# 6. Conditions

Just as for simple syntactic units, it is possible to condition the optional character and the occupation of a **Position**. Yet the **Positions** mentioned in **Si** or **Alors** of **Conditions** may belong not only to the external **Construction** but also to the internal structure of the compound (if however its **Syntagme_NT_S** has been rewritten). It is thus possible to condition the surface realization of complements (or context elements) of the compound but also constituents of the compound itself.

There are several cases for the **Conditions** of syntactic compounds:

- Positions belonging only to the external Construction

See the examples for simple syntactic units

- Positions belonging only to the Internal structure

*Ex :* ***au dŽtriment de*** **SN** (9)

(to the detriment of)

***ˆ son dŽtriment***

(to its detriment)

where the realization of the genitive (field *destination* INTERNE) depends upon the sub-category (definite or possessive) of the

determiner (*destination* INTERNE).

- Positions belonging to both

*Ex :* SN ***vouer/Žprouver/... admiration ˆ/pour*** SN (15)

where the governed preposition (EXTERNE) is different depending on the component verb (INTERNE).

Conditions are borne by the Description that gathers Self and Construction. To reach the internal structure of Self in Conditions, they are given the 'portŽe INTERNE' (internal reach); to reach the external Construction of the compound Self, they are given the 'portŽe EXTERNE' (external reach). Then, CheminPosition and CheminSyntagme allow to navigate through and to point at any Position or Phrase of the internal structure or the Description.

# 7. Transformations

Like units, syntactic compounds have a base **Construction** that may or may not be submitted to **Transformations**. For further details on that topic, refer to the transformations on simple syntactic units.

But compound syntactic units have the special feature to transform on themselves, dependently or independently from their Construction, i.e. the internal structure of the compound may be altered during a **Transformation** process.

- **Construction** and **Syntagme_NT_S** of **Self** modified during passivation

*Ex :* SN*1 mettre en garde* SN*2* (10)

(to warn someone)

SN*2 •tre mis en garde par* SN*1*

(to be warned by)

- **Syntagme_NT_S** modified during possessivation

*Ex : au dŽtriment de* SN (9)

(to the detriment of)

*ˆ son dŽtriment*

(to its detriment)

Since an internal structure is modelled by a **Syntagme_NT_S**, all the Transformation mechanisms defined for the Phrases of simple syntactic units apply to the transformations of the compound itself.

Thus, transformations may, in that case, be described as regular computation operations. Actually going through a tree structure provides a certain level of abstraction that allows to code in the same way (since one works on non terminal nodes) a same transformation that would apply to syntactic compounds slightly different form one another in surface.

# VII. Construction

As a general rule, compounds (whether they are morphological or syntactic compounds) are units with a complex lexical realization. As units, one has to describe their way of functioning by specifying their complementation pattern or their occurrence context. From that point of view, they behave exactly just as simple units and all the mechanisms that have been previously defined for these simple units also apply to compounds.

In the Construction, Self is used to refer to the compound. In other words, the process is the same as for simple units, except that in the composition, Self also gives the Inner Structure of the unit considered as a whole in the base Construction.

- **SV compounds in a functional position of VERB**

Compound verbs have to be described as simple verbs while specifying the set of complements and their subject position.

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

(to warn someone)

- **Noun compounds with a functional position of NOUN**

Defined as being on an N position, they have therefore a complementation pattern as a simple N: Determiner, Left-modifier, Right-modifier.

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

(a red mailbox)

*une* ***armoire ˆ glace*** *cultivŽe*

(a cultured strong man)

*une* ***femme ˆ poigne*** *sans poigne*

(a firm-handed woman without strength)

*une Žnorme* ***turbine ˆ hŽlice***

(a huge propeller-type turbine)

*une* ***rinceuse ˆ bouteilles*** *qui ne servira jamais*

(a bottle rinsing machine that will never be used)

- **Noun compounds in a functional position of DETERMINER**

Those who want to process complex determiners as such right from the syntactic level can do so by associating them with a position DŽterminant and specifying their occurrence context (i.e. analysing it).

*Ex : la **plupart** des gens (17)*

(most people)

**Note**: our purpose here is not to discuss the analysis of this quantitative *de* as a particle or as a preposition, knowing that GENELEX allows to record either indifferently.

Nevertheless one has to ensure minimum consistency with the analysis that is made on simple words (*beaucoup* in the following example).

*Ex : **beaucoup** de gens*

(many people)

- **SP compounds in a functional position of CONJUNCTION**

Some prepositional phrases are used as conjunctions and introduce whole phrases. It must be possible to specify the type of sentences introduced and their mode.

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

(In consequence, he did not come)

- **SP compounds in a functional position of ADVERB**

Attempting to describe that such fossilized (or partially fossilized) prepositional phrase occupies an adverbial function allows to define the scope of the prepositional phrase and to remove ambiguities as regards the attachment of the prepositional group. Everybody knows how much this problem is thorny in syntactic analysis.

*Ex : Il empruntera cette voie **ˆ ses risques et pŽrils**. (7)*

(he will take this way at his own risks)

In this example, the SP is a sentence adverb. Its "complement" is therefore a sentence.

- **SP compounds in a functional position of ADJECTIVE**

Used as adjectives, it is interesting to know for these compounds - as well as for simple adjectives - whether they may be left-positioned attributes, right-positioned attributes, predicatives, and whether they may take complements.

*Ex : Il est **sans vergogne***

(He is shameless)

*Un gar•on* ***sans vergogne*** *a dŽboulŽ dans la pi•ce*

(A shameless boy ran into the room)

- **Compound Usyns with no external complementation**

The dichotomy inside/outside of the compound is based on the comparison of the compound to a terminal category.

If a complex unit does not function as a terminal category, then it has no external behavior. Its whole description is done in its internal structure. It is the case of set phrases.

*Ex:* ***pierre qui roule n'amasse pas mousse***

*a rolling stone gathers no moss*

# D - Annex to compound Syntactic Units

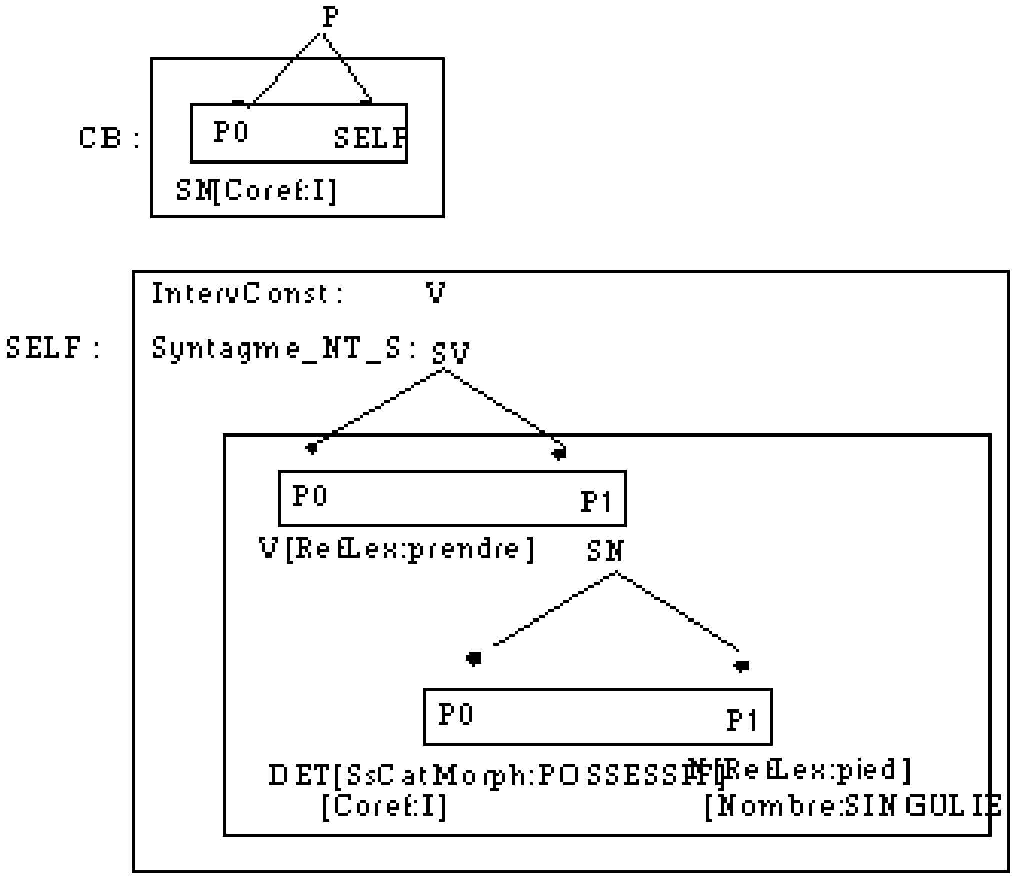
## NOTE

The coding examples that are presented in this document correspond to **special lexicographic** choices, other choices may also be possible within the GENELEX model. The structuring selected for syntagmatic trees, the splitting in syntactic units, the description of those units, the use of associated conditions or even the gathering of lexicalization alternatives do not aim at imposing the **way the example is coded**, but only at illustrating how the GENELEX model allows to give account of it, once certain lexicographic choices have been made.

It is therefore very important to read these examples in this perspective and not to forget that the implementation of the GENELEX model goes first through the elaboration of a coding lexicographic strategy, that specifies the criteria for splitting syntactic units, the structures of phrases used and the gatherings to make.

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

(to get one's kicks)

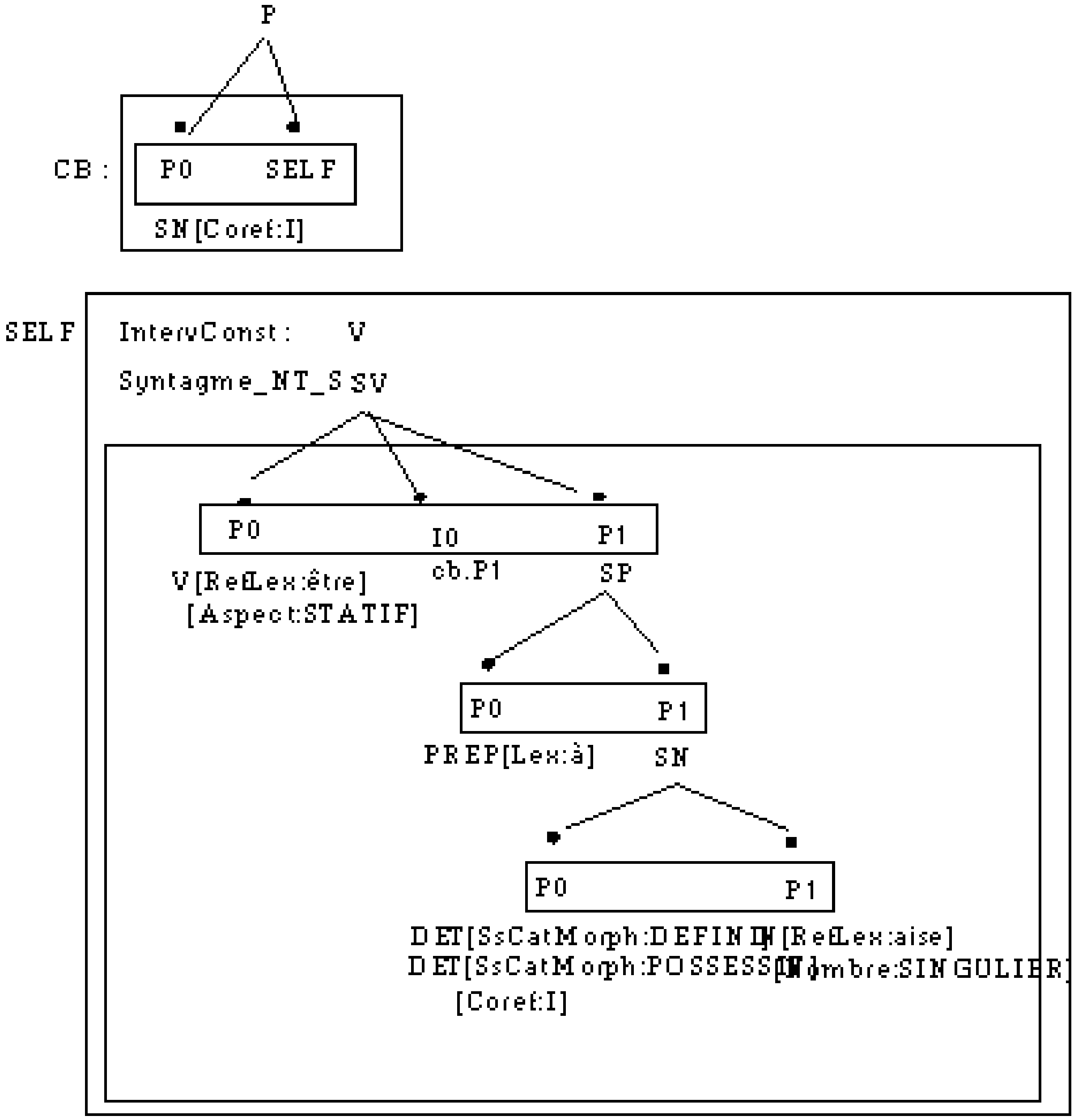


*Example (2) : Il* ***est ˆ l'/son aise***

(he is at ease)

*Ils* ***sont ˆ l'/leur aise***

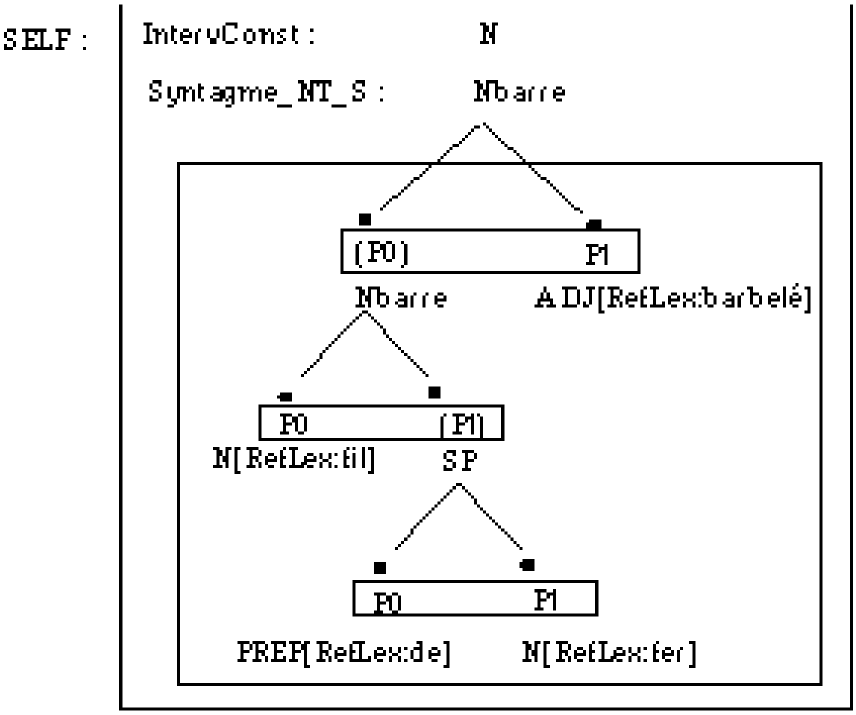
(They are at ease)



*Example (3) :* ***fil de fer barbelŽ*** (barbed wire)

***fil barbelŽ***

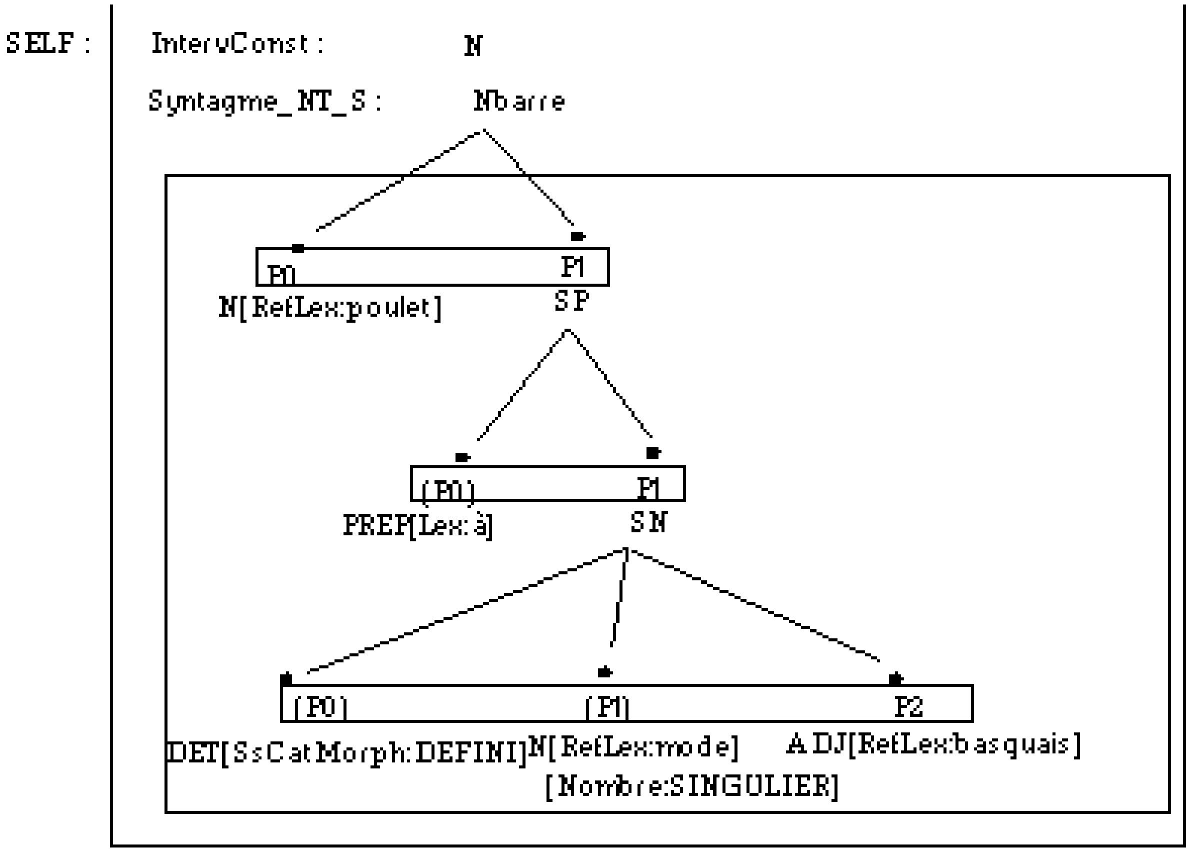
***barbelŽ***



*Example 4* : ***poulet ˆ la mode basquaise*** (Basquaise chicken)

***poulet ˆ la basquaise***

***poulet basquaise***



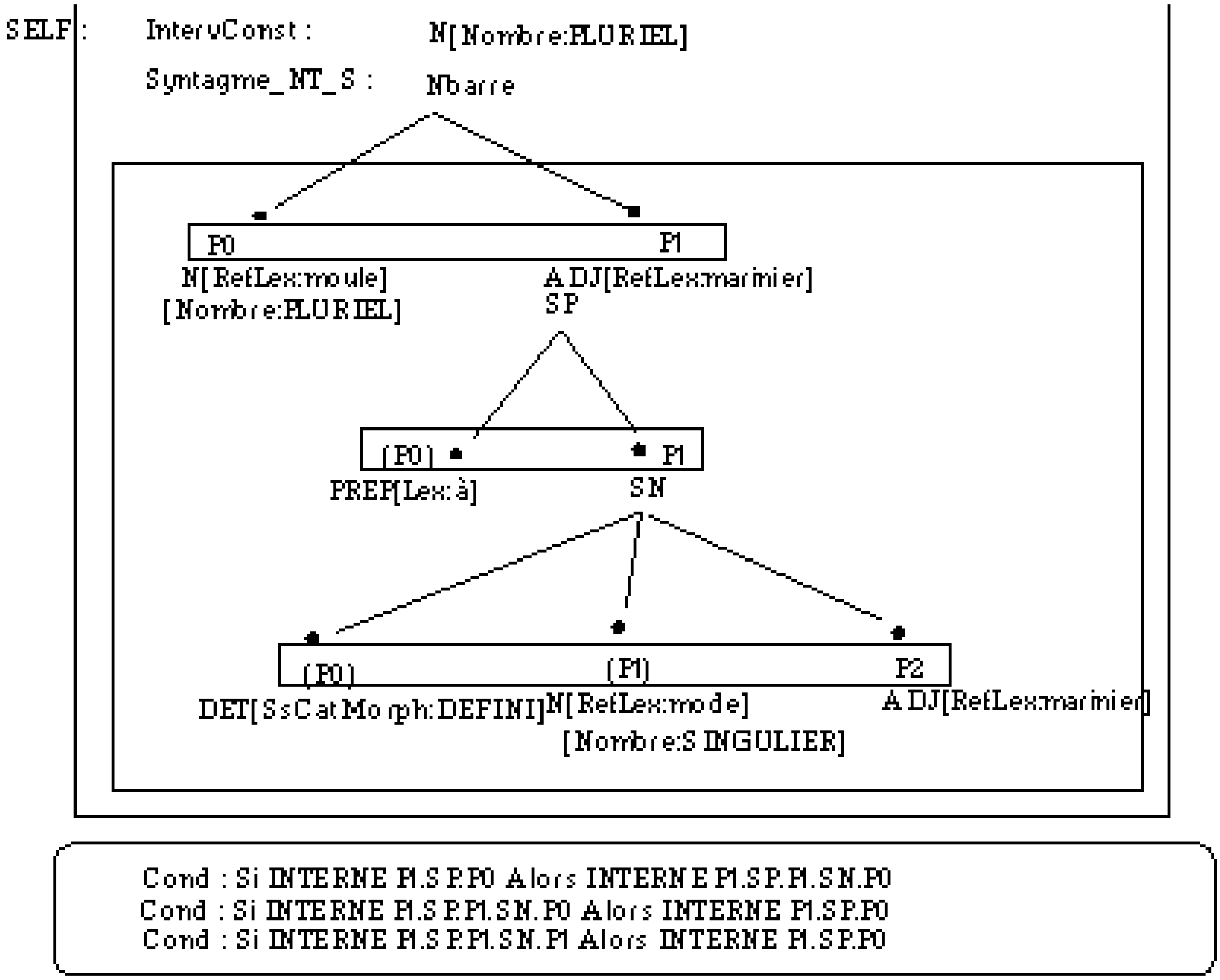
Cond : Si INTERNE ! P1.SP.P0 Alors INTERNE ! P1.SP.P1.SN.P0 & ! P1.SP.P1.SN.P1
Cond : Si INTERNE ! P1.SP.P1.SN.P0 Alors INTERNE ! P1.SP.P0 & ! P1.SP.P1.SN.P1

*Example (5) :* ***moules ˆ la mode marini•re*** (moules marini•res)

***moules ˆ la marini•re***

***moules marini•re***

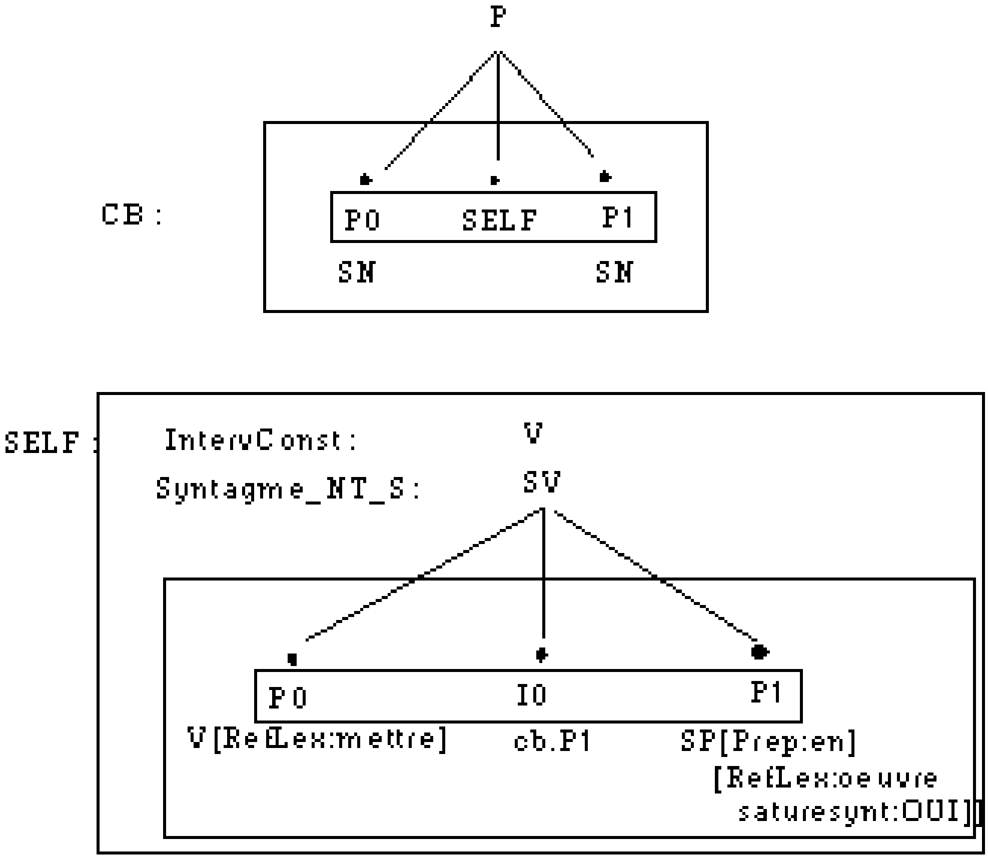
***moules marini•res***



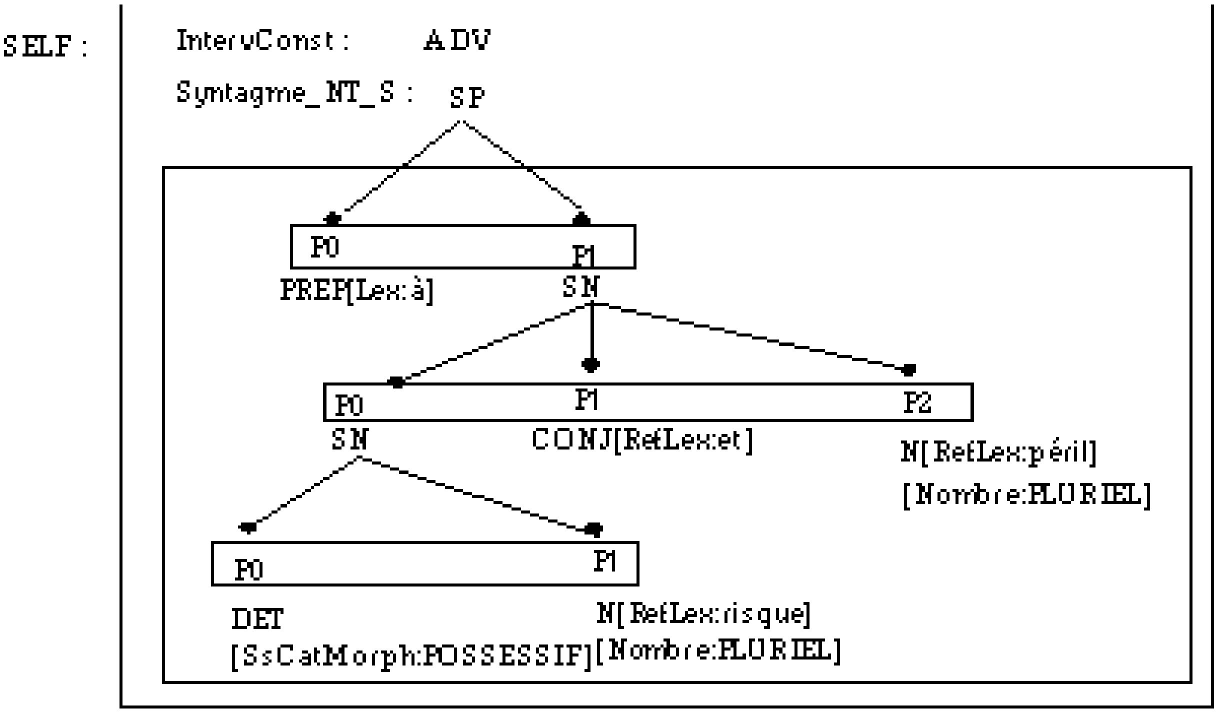
Cond : Si INTERNE P1.SP.P0 Alors INTERNE P1.SP.P1.SN.P0
Cond : Si INTERNE P1.SP.P1.SN.P0 Alors INTERNE P1.SP.P0
Cond : Si INTERNE P1.SP.P1.SN.P1 Alors INTERNE P1.SP.P0

*Example (6) : SN* ***mettre*** *SN* ***en Ïuvre***

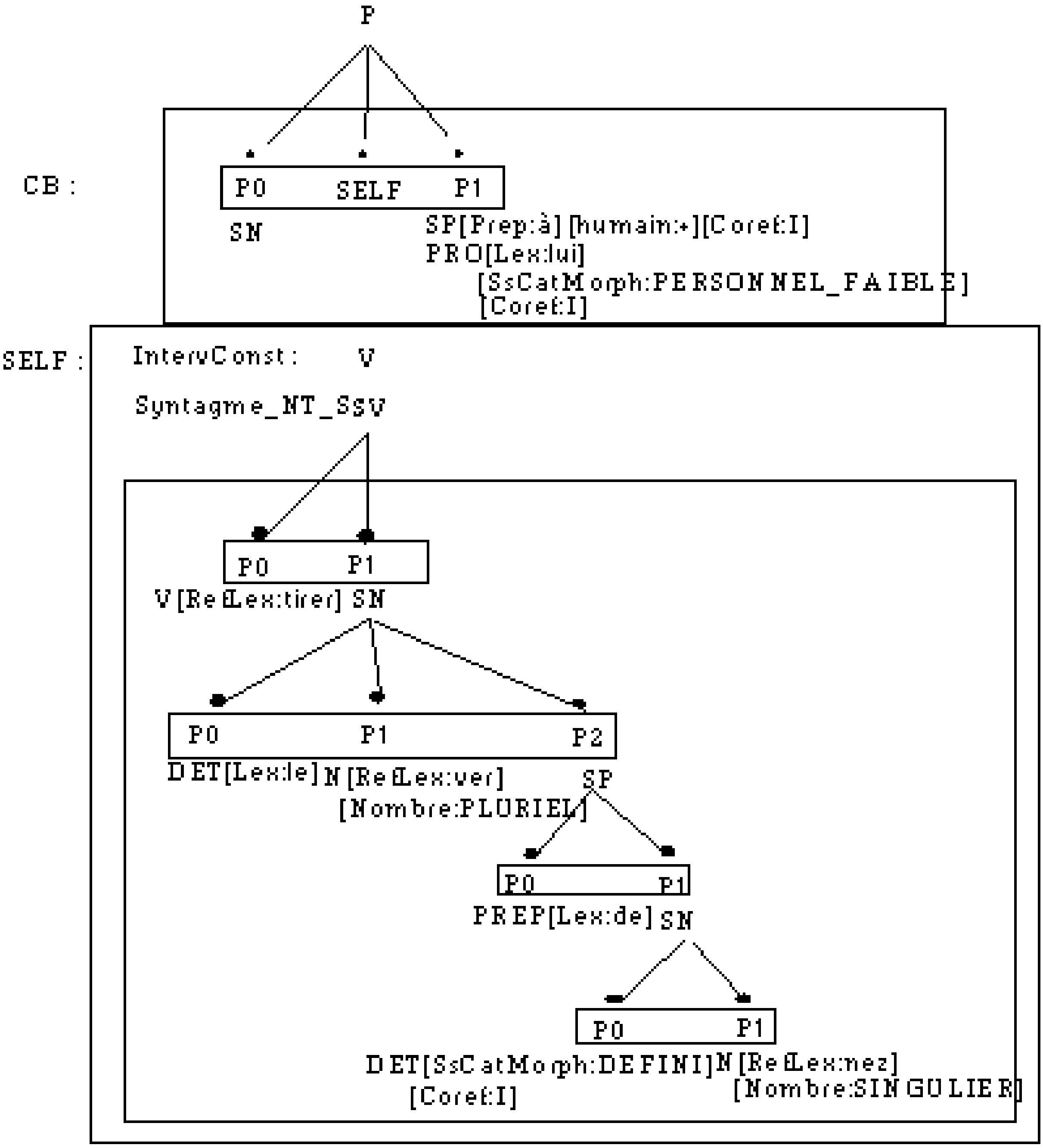
(to implement)



*Example (7) :* ***ˆ ses risques et pŽrils*** (at one's own risks)



*Example (8) :* *t****irer les vers du nez*** (to worm information out of sb)



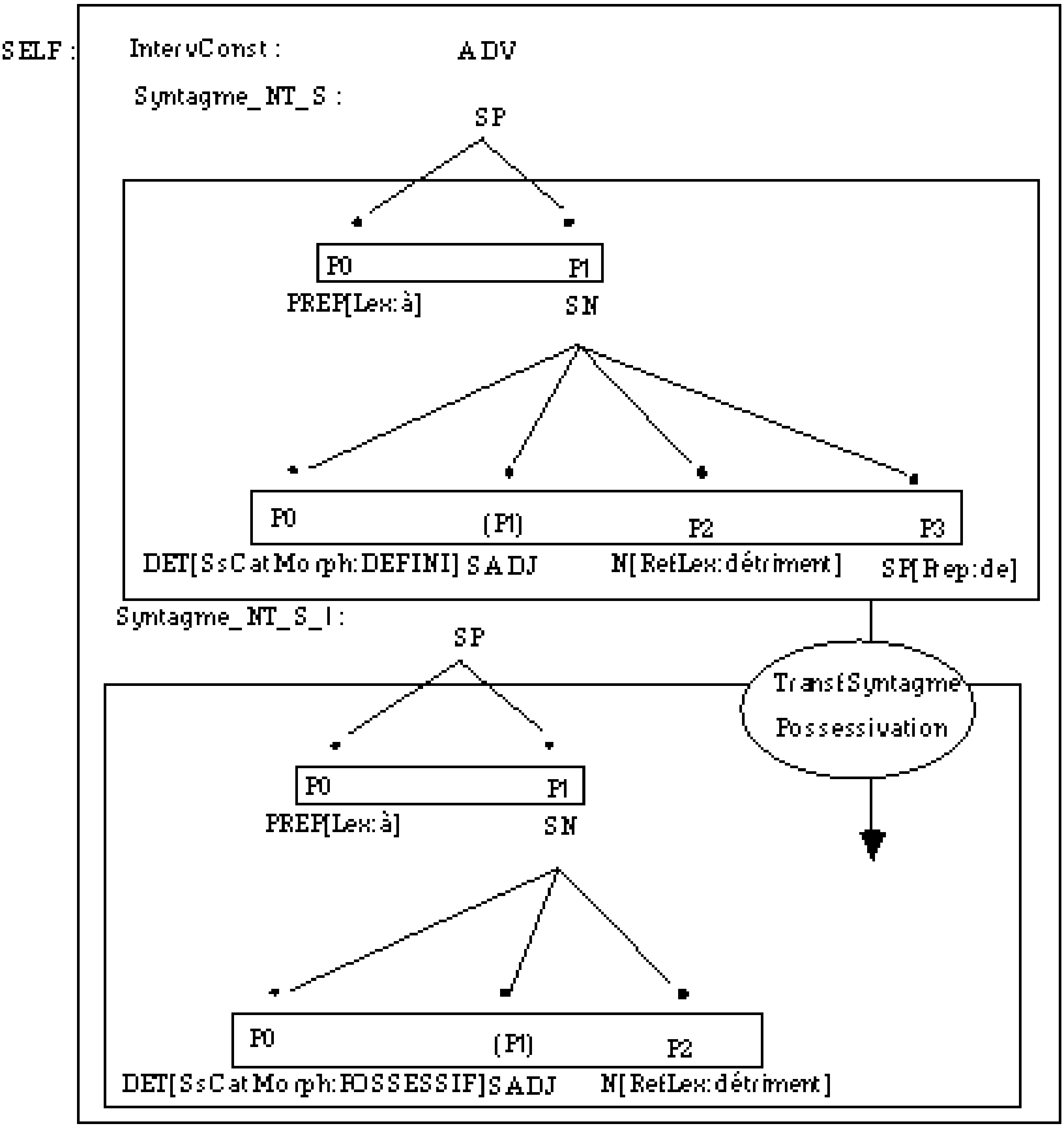
*Example (9)* ***au*** *(rŽel)* ***dŽtriment de SN***

(to the (real) detriment of)

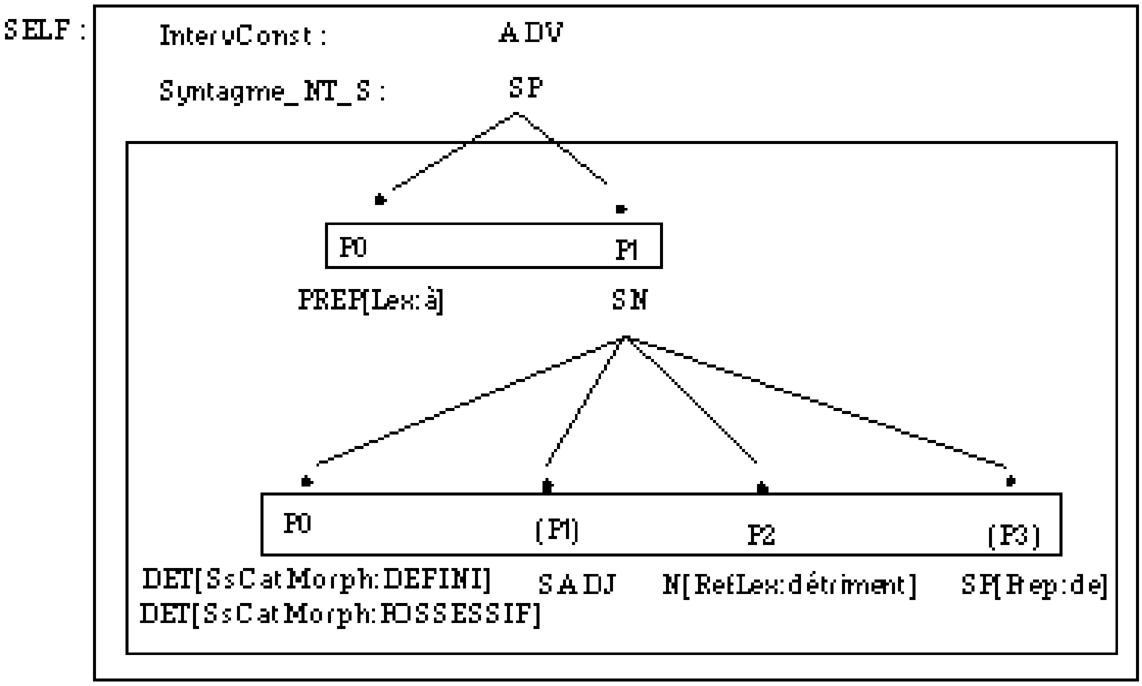
***ˆ son*** *(rŽel)* ***dŽtriment***

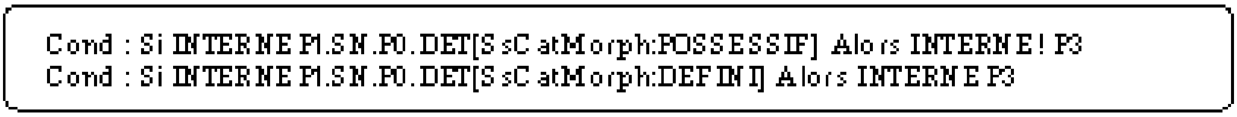
(to his/her (real) detriment)

***Solution 1:*** *Relation of transformation between two phrases:*



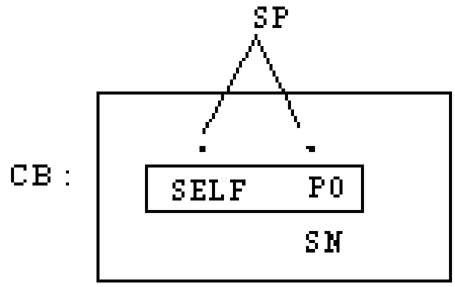
**Solution 2:** inter-conditioned positions

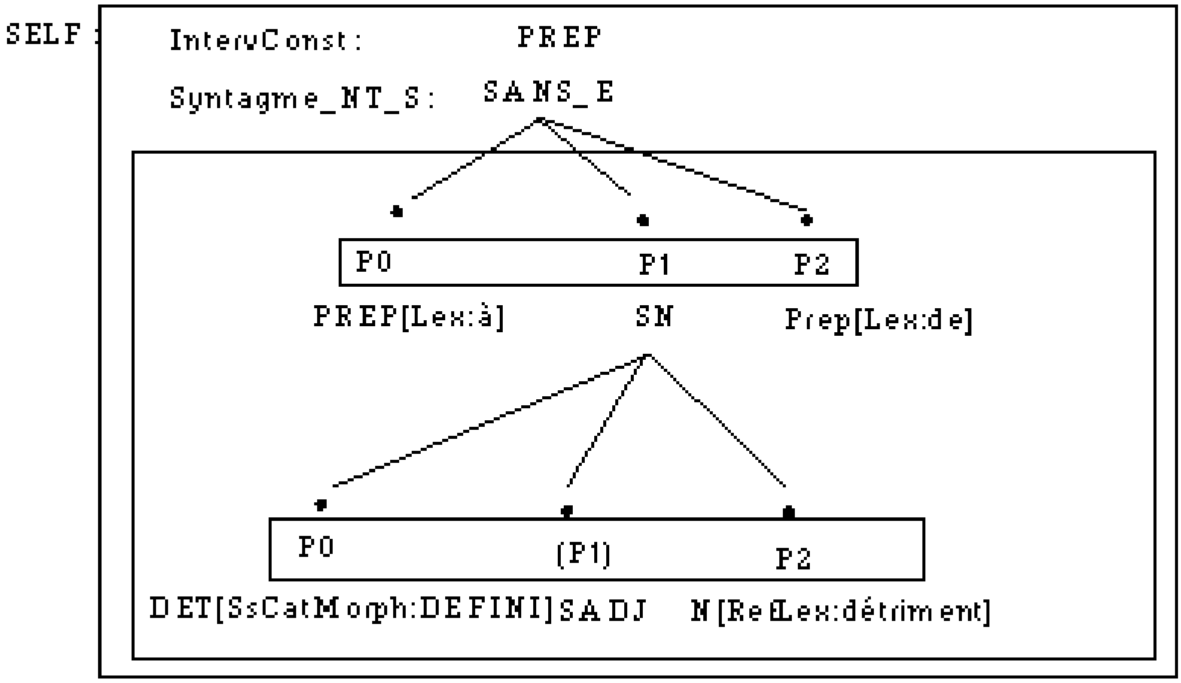



Cond : Si INTERNE P1.SN.P0.DET[SsCatMorph:POSSESSIF] Alors INTERNE ! P3
Cond : Si INTERNE P1.SN.P0.DET[SsCatMorph:DEFINI] Alors INTERNE P3

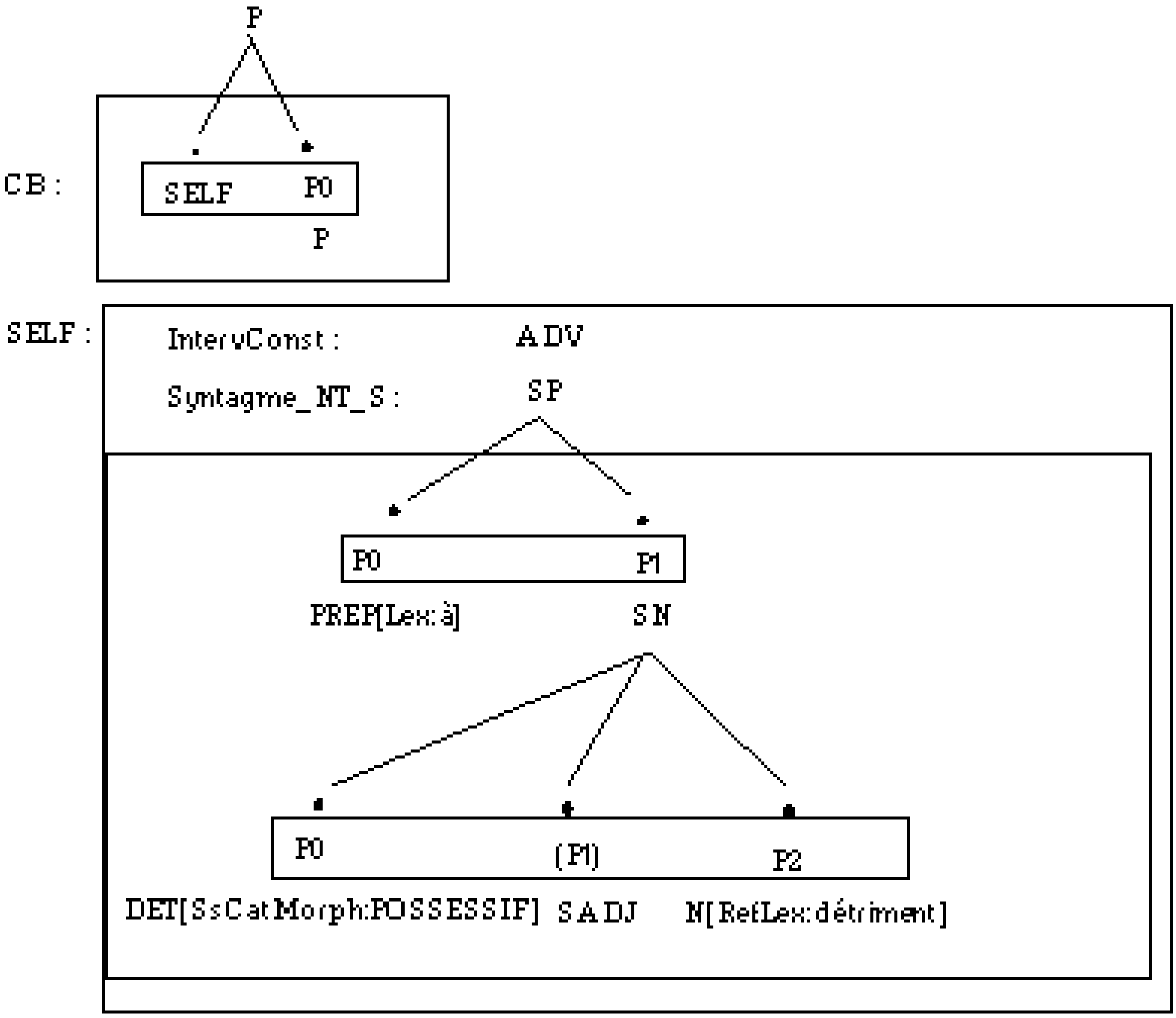
**Solution 3:** two syntactic units, one of which covering the classical idiom "au dŽtriment de " (to the detriment of).

***<u>First Usyn</u>***


SP
CB :
SELF
P0
SN

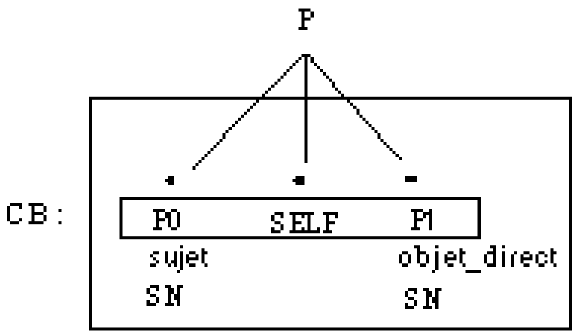

SELF :
IntervConst :
PREP
Syntagme_NT_S :
SANS_E
P0
P1
P2
PREP[Lex:à]
SN
Prep[Lex:de]
P0
(P1)
P2
DET[SsCatMorph:DEFINI]
SADJ
N[RefLex:détriment]

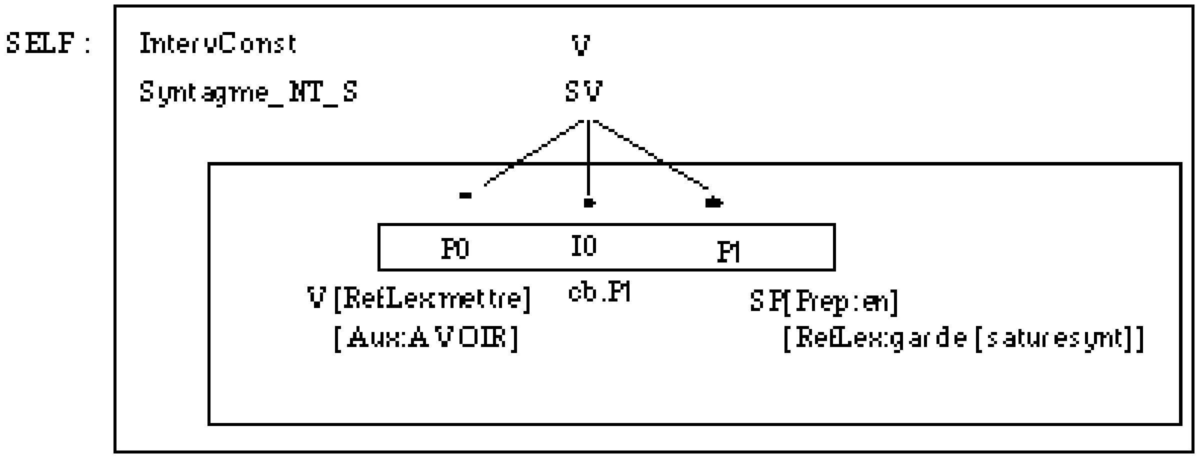
## ***Second Usyn***

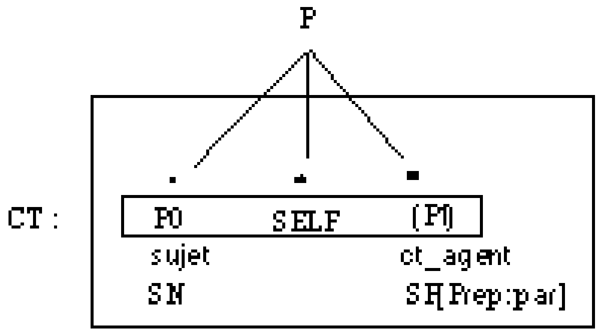


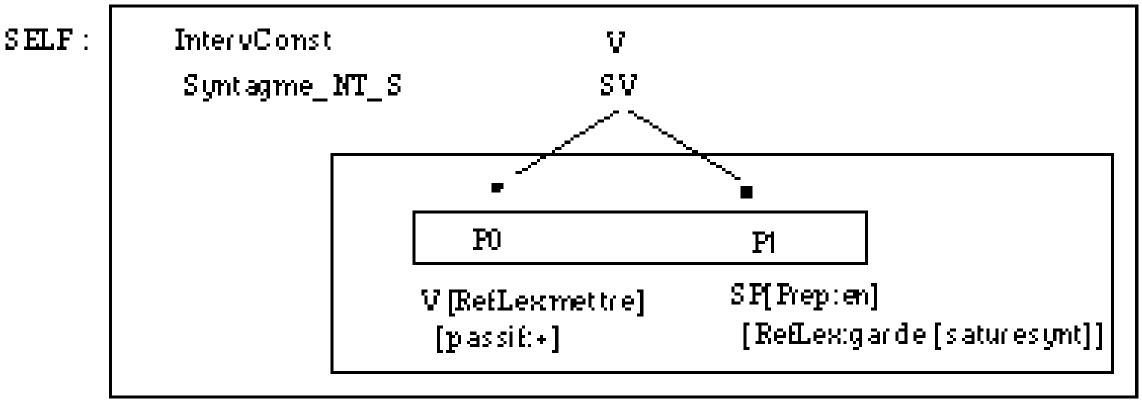
*Example (10) :* SN*1* ***mettre en garde*** SN*2* (to warn)

SN*2* ***•tre mis en garde par*** SN*1* (to be warned by)









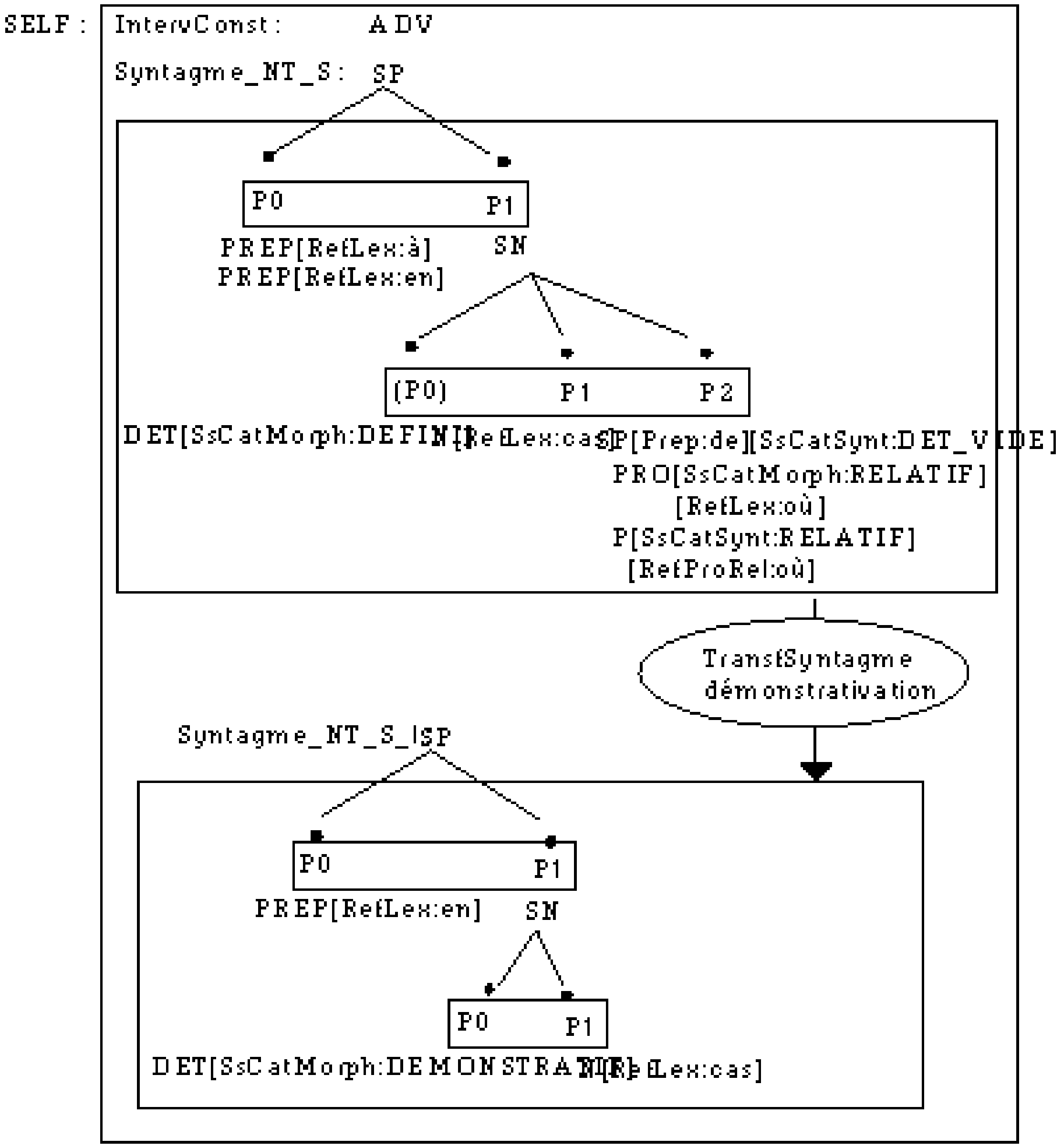
*Example (11) :* ***Au cas o•*** (if)

***Au cas o•*** *P* (if)

***En cas de*** *SN* (in case of)

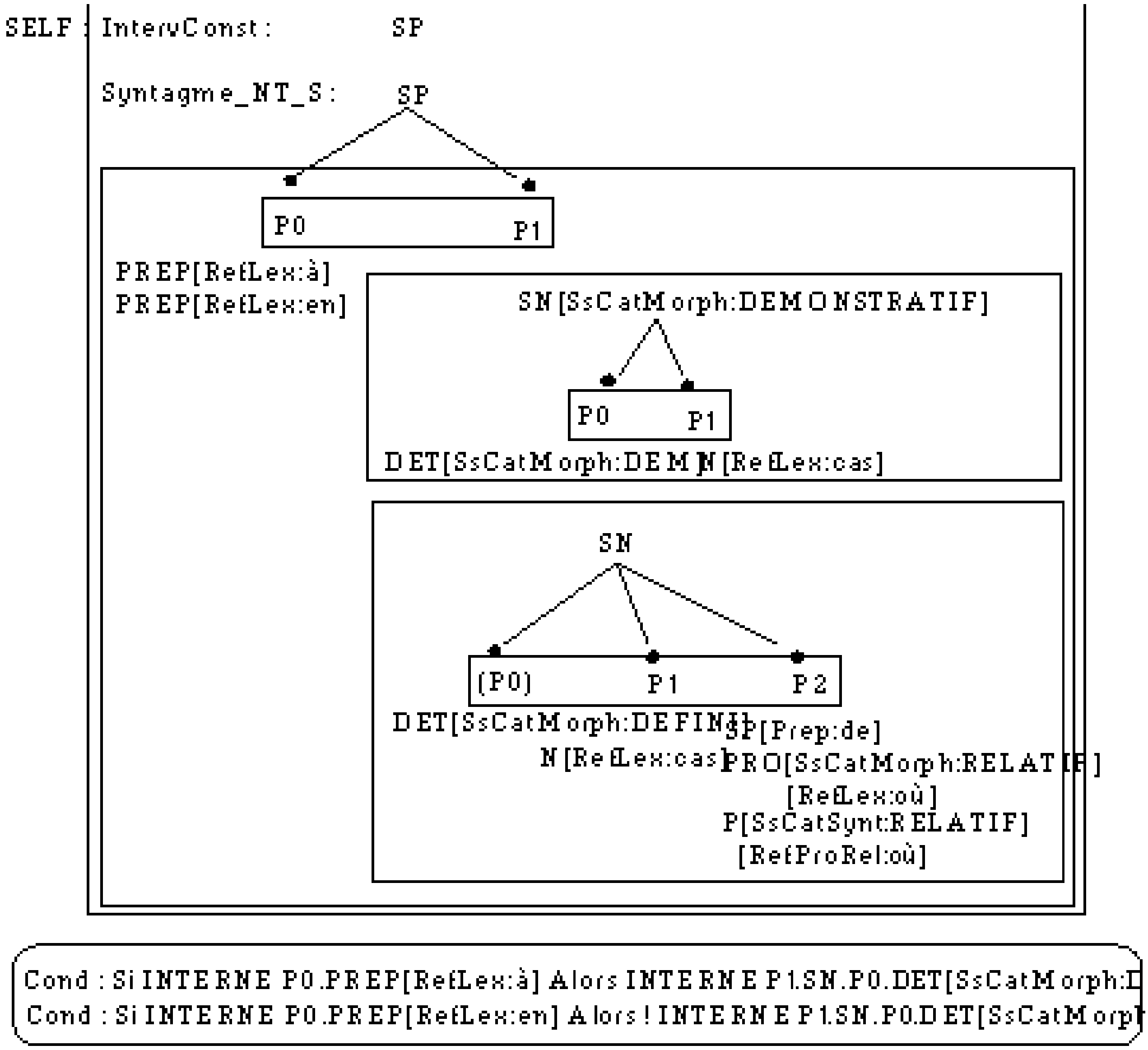
***En ce cas*** (in that case)

**Solution 1:** Relation of transformation between two phrases:



Cond : Si P0.PREP[RefLex:en] Alors !P1.SN.P0 & P1.SN.P2.SP[Prep:de][SscatSynt:DET_VI
Cond : Si P0.PREP[RefLex:à] Alors P1.SN.P0 & P1.SN.P2.SP[Prep:de][SsCatSynt:DET_VID

**Solution 2:** Inter-conditioned positions



Cond : Si INTERNE P0.PREP[RefLex:à] Alors INTERNE P1.SN.P0.DET[SsCatMorph:D
Cond : Si INTERNE P0.PREP[RefLex:en] Alors ! INTERNE P1.SN.P0.DET[SsCatMorph

### Graphic conventions on Composition Modes

To facilitate reading, Composition Modes are split in two: composition toward the INSIDE, composition toward the OUTSIDE. On the contrary, all the composition modes of a compound Usyn are gathered in the same pattern- i.e. its composition modes on each of its component Usyns- and the whole is miscalled "MdC".

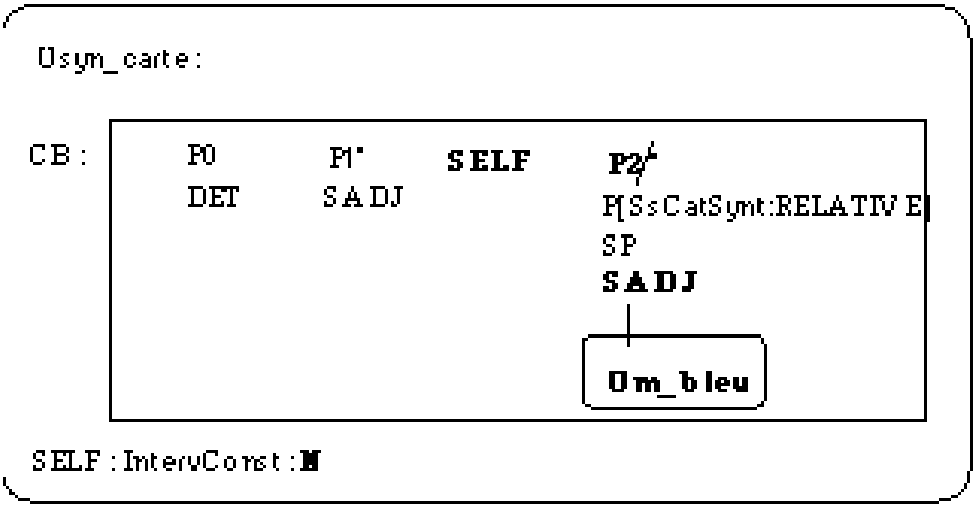
Um and Usyn components are in round frames.

Calls for Usyn or Um components are expressed by a line linking the calling phrase and the component.

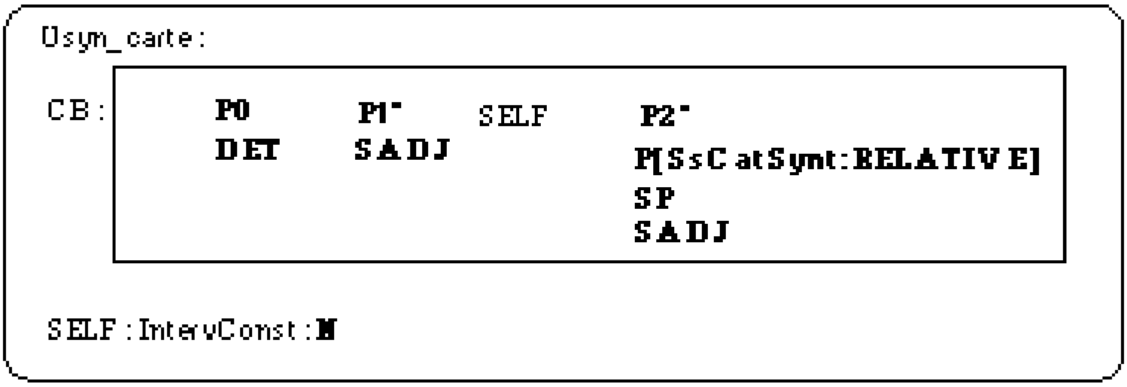
The Positions and Phrases we want to inherit from are in bold characters. The Phrases or features added during composition are in round frames. The loss of repeatability (asterisk) is indicated by a crossed asterisk.

*Example (12) :* ***carte bleue*** (Visa card)

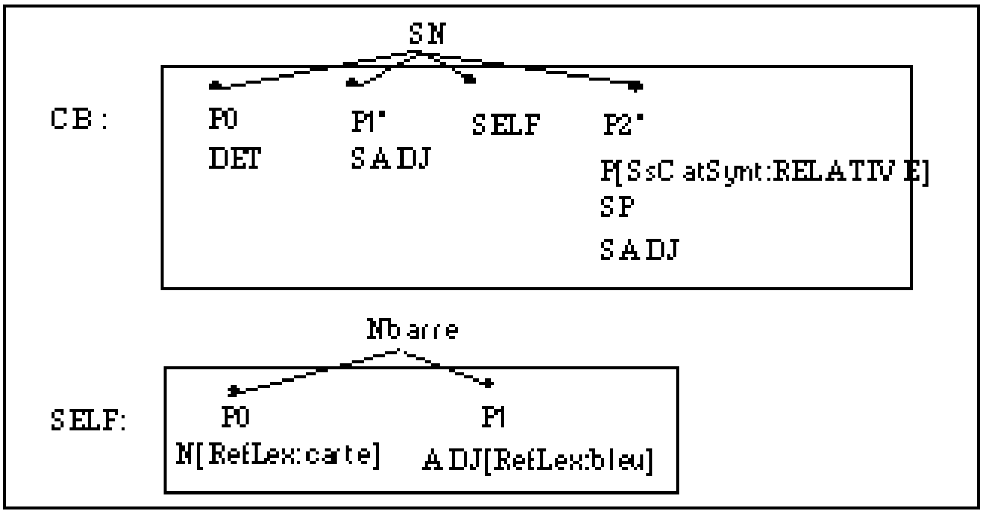
*Mode de Composition INTERNE de : CARTE_BLEUE*



*Mode de Composition EXTERNE de : CARTE_BLEUE*



*Unité syntaxique composée : Carte_Bleue :*

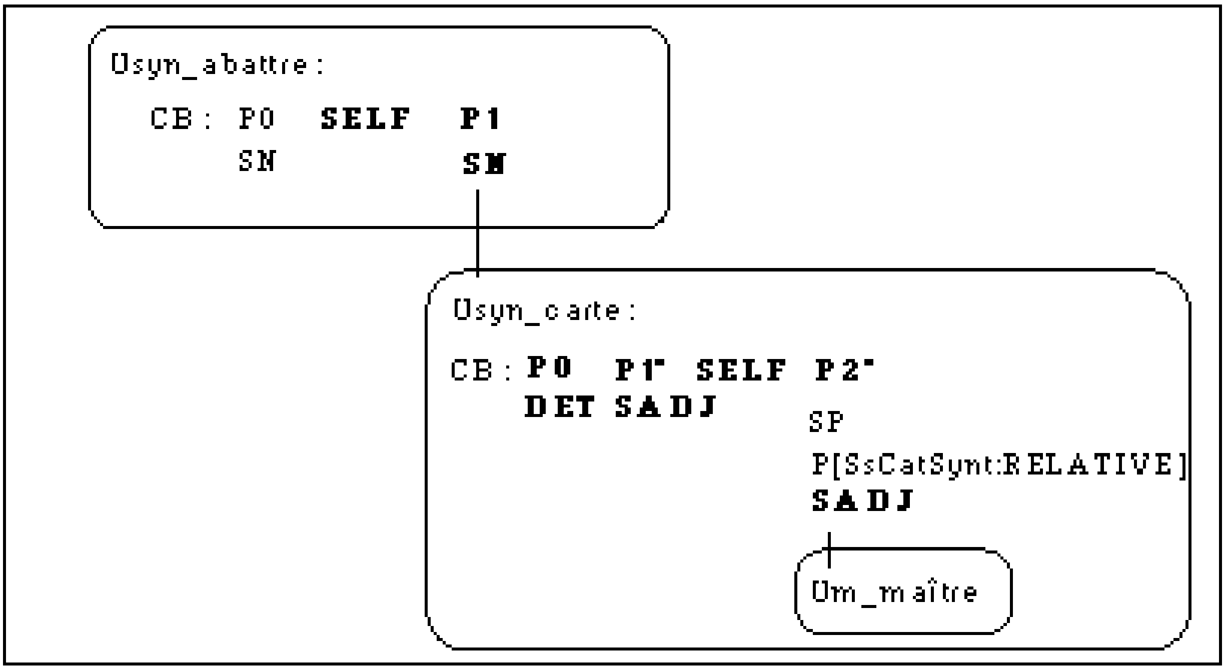


*Example (13) :* ***abattre carte maitresse*** (to show one's master card)

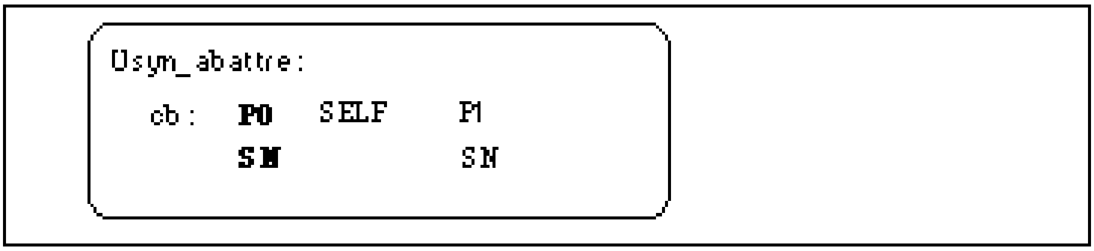
**Note: this example is explained in section H-III of this document (SGML DTD - Examples of tagged data)**

**Solution 1:** 3 components (Usyn abattre, Usyn carte and Um ma"tre)

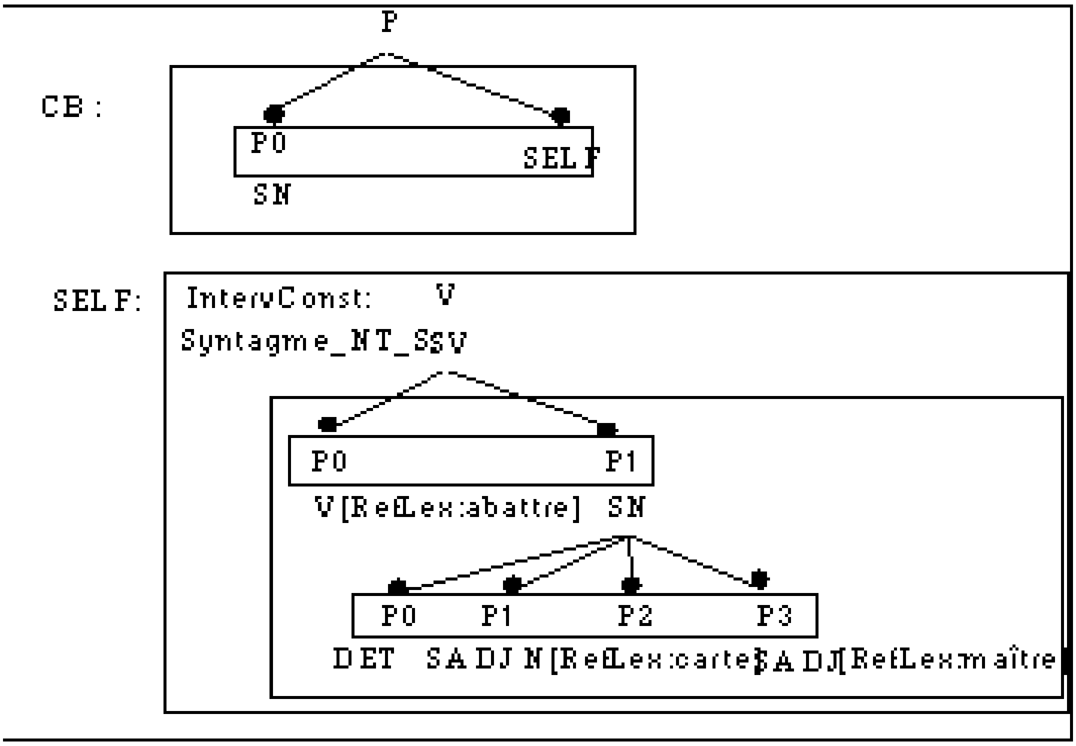
*Mode de composition INTERNE de : ABATTRE_CARTE_MAITRESSE*



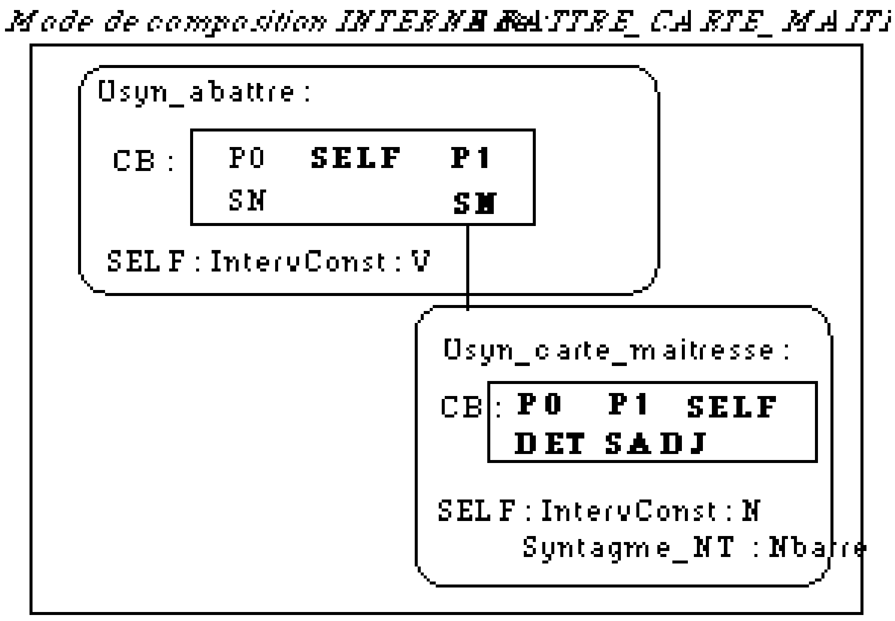
*Mode de composition EXTERNE de : ABATTRE_CARTE_MAITRESSE*

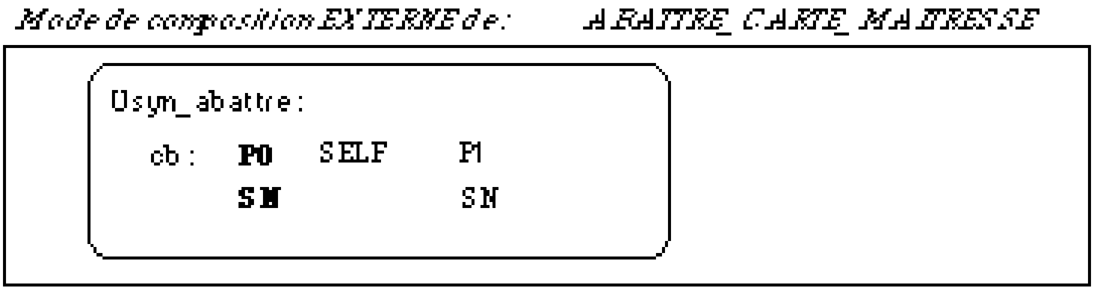


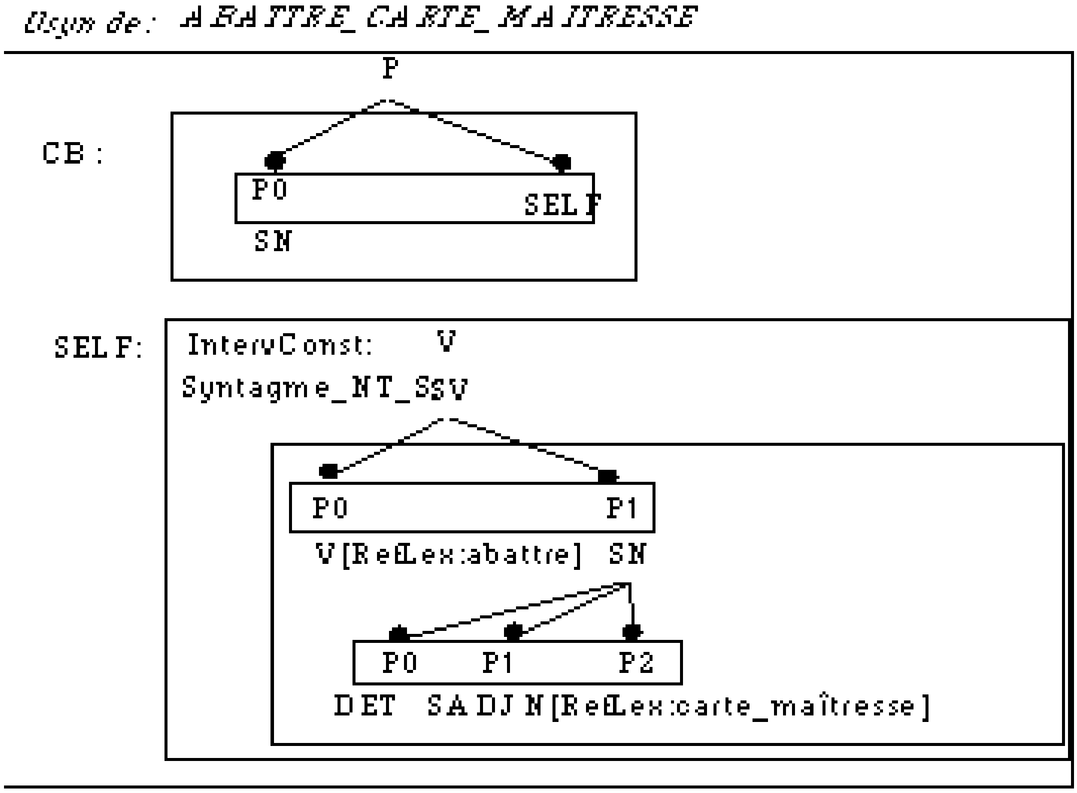
*Usyn de : ABATTRE_CARTE_MAITRESSE*



**Solution 2:** 2 components, one of them being a Usyn compound *carte maitresse.*

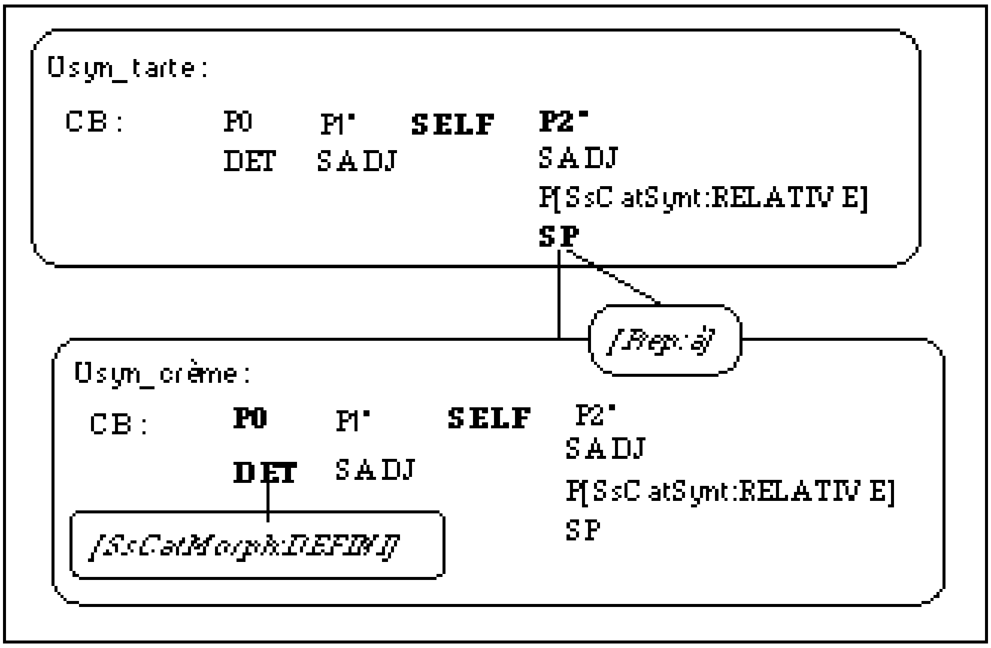




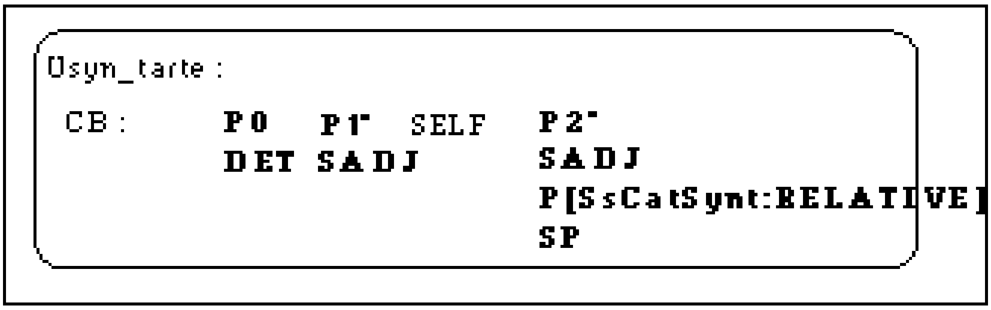


*Example (14) :* ***tarte ˆ la cr•me***

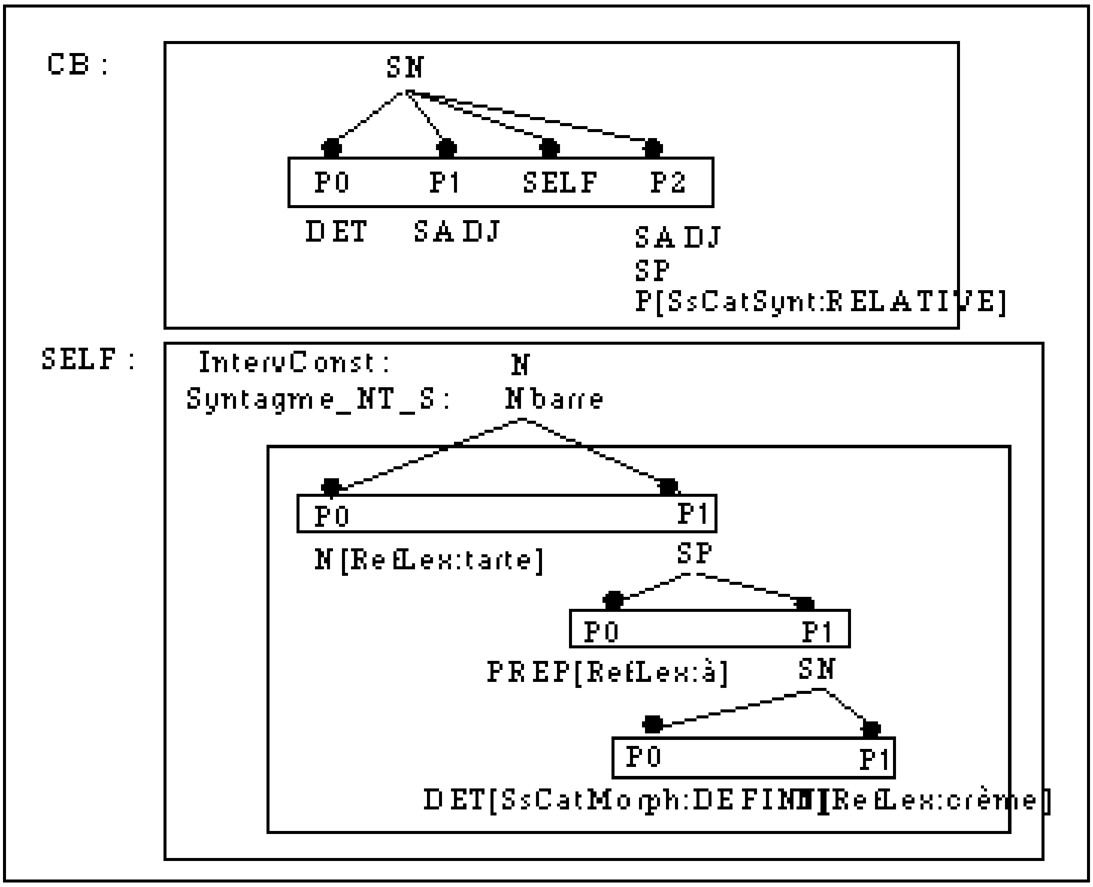
*Mode de composition INTERNE de : TARTE_A_LA_CREME*



*Mode de composition EXTERNE de : TARTE_A_LA_CREME*



*Usym composée : TARTE_A_LA_CREME*



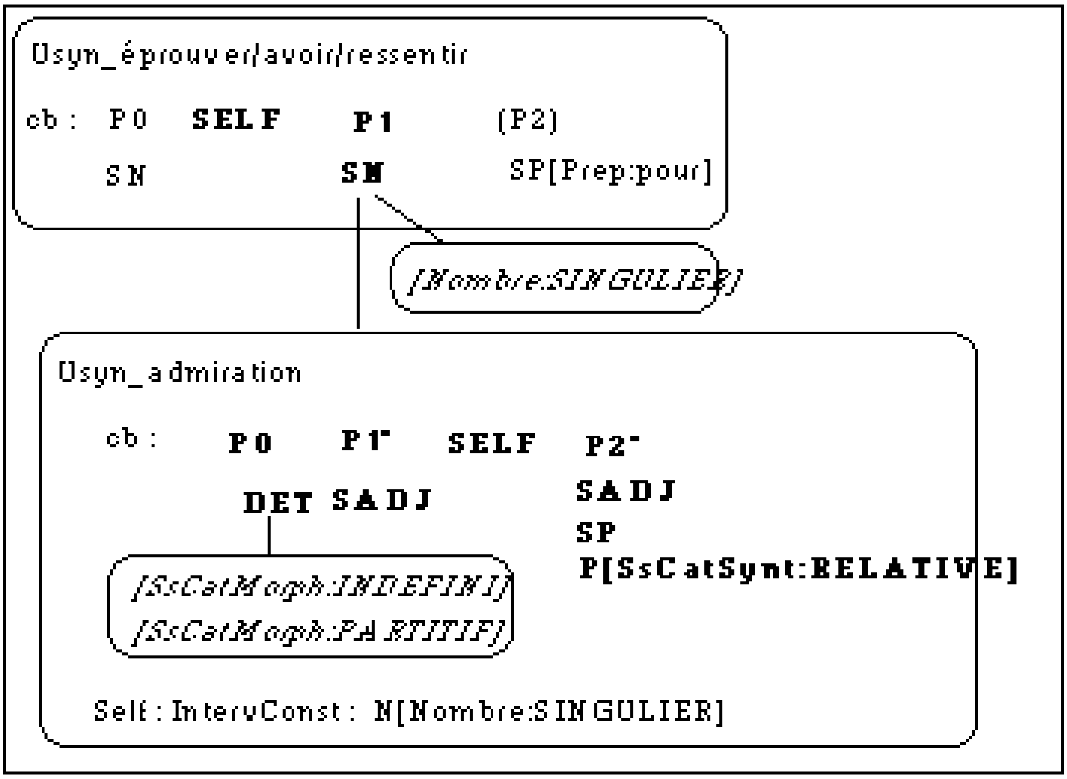
*Example (15¡:* SN ***avoir admiration*** pour SN

SN ***Žprouver admiration*** pour SN

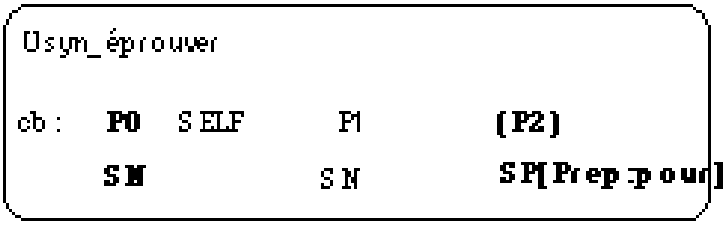
SN ***ressentir admiration*** pour SN

SN ***vouer admiration*** ˆ SN

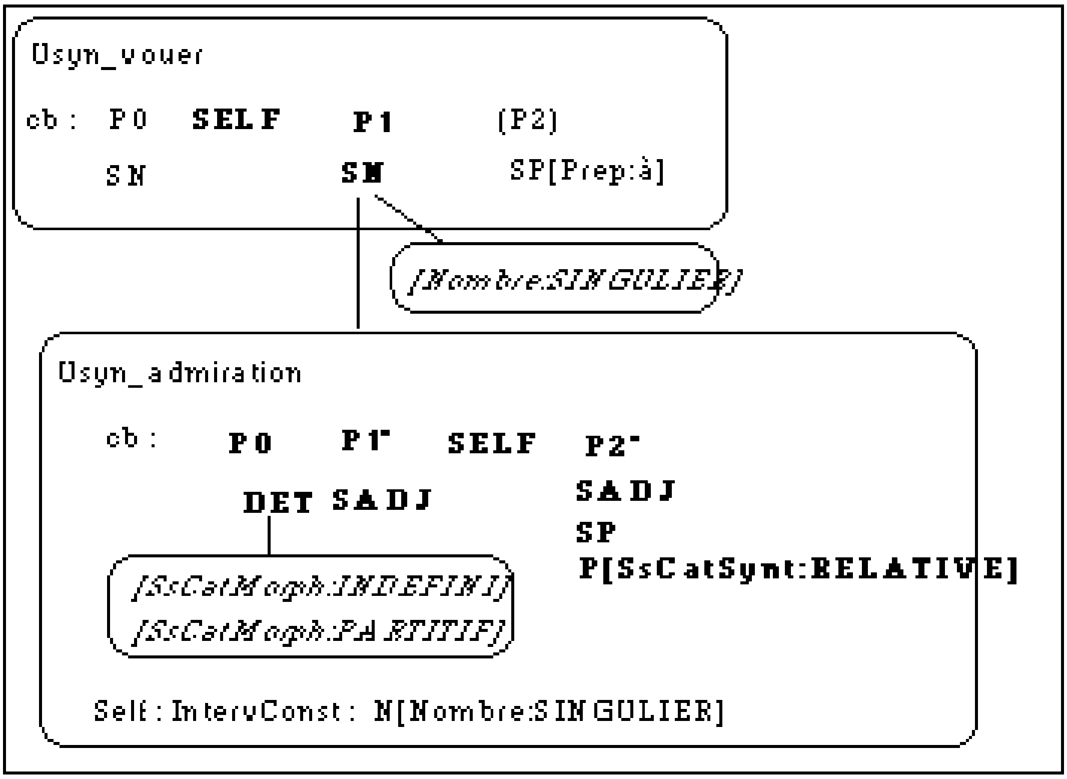
*Mode de composition INTERNE de : EPROUVER/.../..._ADMIRATION*



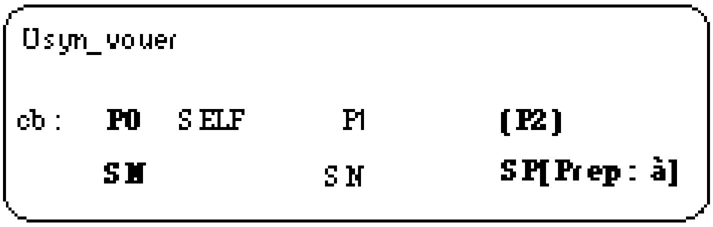
*Mode de composition EXTERNE de : EPROUVER_ADMIRAI*

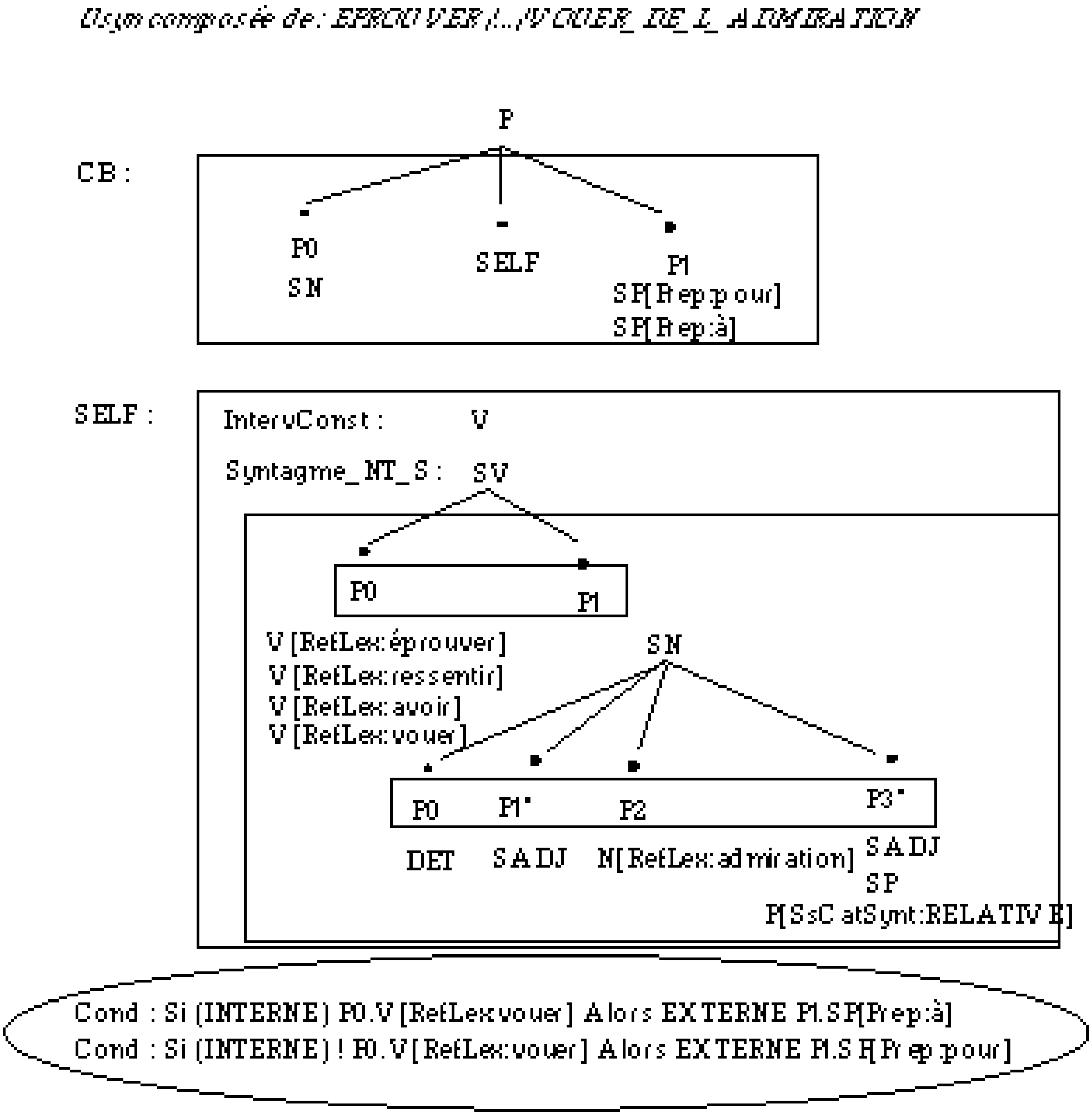


*Mode de composition INTERNE de : VOUER_ADMIRATION*



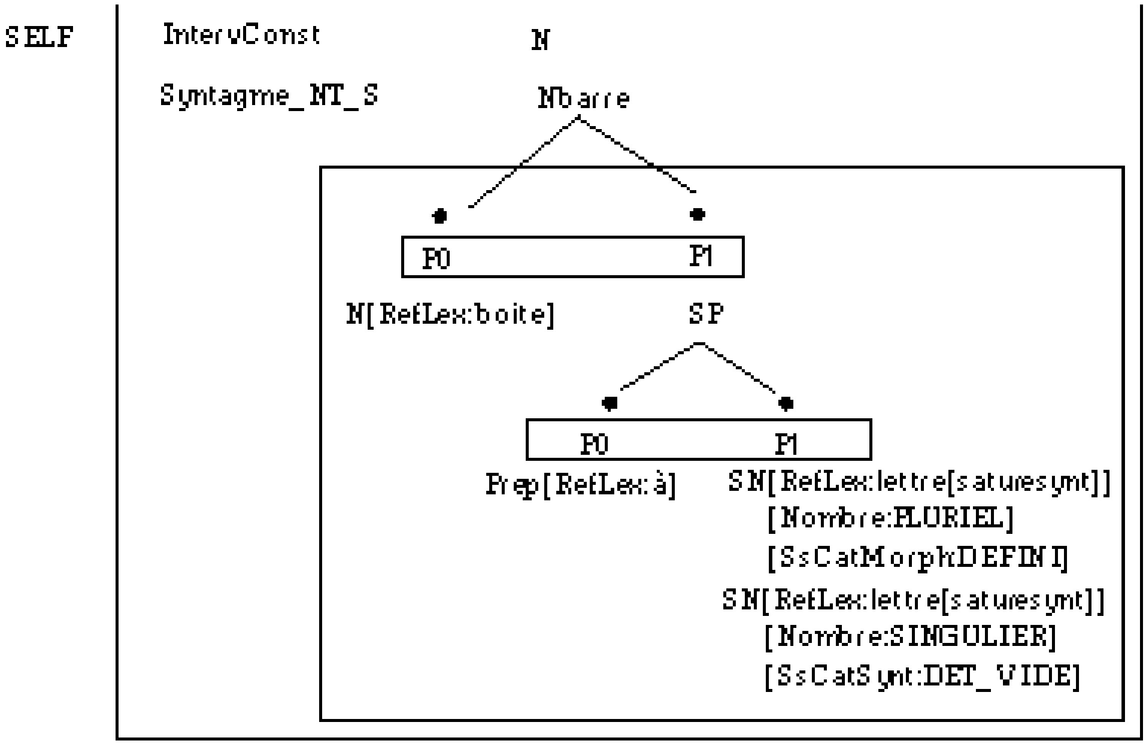
*Mode de composition EXTERNE de : VOUER_ADMIRAT*

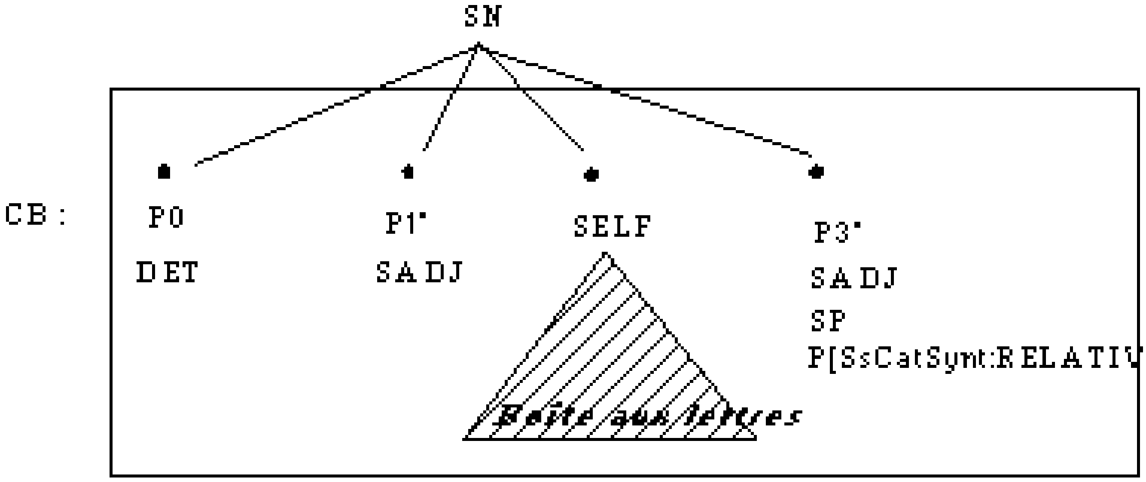




*Example (16) :* ***bo"te aux lettres*** (mailbox)

***bo"te ˆ lettres*** (mailbox)

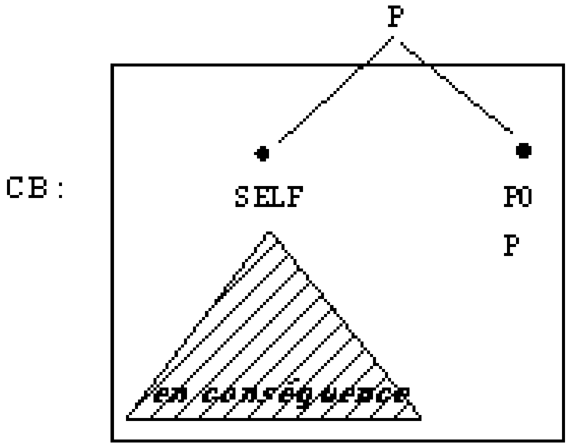




*Example (17) : la **plupart** des gens* (most people)

*Example (18) : **En consŽquence** il n'est pas venu.*

(in consequence he did not come)



# E - Bibliography

A. AbeillŽ

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

Probl•mes de linguistique gŽnŽrale; tome 2. Gallimard

Paris, 1974.

G.B•s & K.Bashung & A.Lecomte

Une modŽlisation des entrŽes lexicales.

Projet EUREKA GENELEX Janv. 91.

G.B•s & M.Emorine

Une lecture des tables du Ladl en vue de la dŽfinition 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.Lecl•re

La structure des phrases simples en fran•ais : 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 dŽfinition : de la lexicographie ˆ la sŽmantique*.

Langue Fran•aise 43; p.60-83.

Paris, 1979

Gazdar & Klein & Pullum & Sag

Generalized phrase structure grammar.

Harvard University Press, 1985.

M.Gross

MŽthodes en syntaxe: rŽgime des constructions complŽtives.

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 ˆ une science du langage

Seuil 1989.

A. Poncet-Montange

Les composŽs NˆN.

Communication GENELEX. Paris 1990

L. Tesni•re

ƒlŽments de syntaxe structurale.

Klincksieck 1959

# F - User's Manual

## 1: Usyn

### 1.1: General

The Syntactic Unit (**Usyn**) is the access point to the syntactic layer of the model.

Simple Usyns are not shared by Ums.

All of them may bear a **CombVE,** a combination of four use values: level of language (*niveaulgue*), frequency (*frŽquence*), geographical variant (*vargeog*) and *datation*.

Usyns obligatorily have a **Base Description** (attribute *description*) and 0 to N **Transformed Descriptions** (attribute *description_l*). A same Description cannot have both the status of base and that of transformed on the same Usyn.

The **Transformation** relations between Usyns and between Descriptions are borne at the level of Usyns (**TransfUsyn** element and attribute transfdescription_l ).

### 1.2: Simple Usyn

A simple Usyn allows to associate one Um with one or several syntactic behaviors and to attach specific information such as attestation. It is useless actually, unless one wants to pollute the dictionary or make it worthless, to record on an entry virtual characteristics that are not attested in language. Hence the importance of the field *attestation* that prevents from these problems if it is documented.

A Um is in relation with 0 to N (for instance affixes and agglutinates) or 1 to N Usyns.

A simple Usyn is in relation with a single Um for which it describes a single syntactic behavior. Since Usyns are not shared, there are at least as many simple Usyns as Ums described by these Usyns.

A Usyn is minimally described by a Base description and 0 to N Transformed descriptions.

### 1.3: Compound Usyn

A compound Usyn is a totally or partially lexicalized well-constructed Phrase. It is in relation with Ums and/or Usyns that lexicalize it and make it up (element **Composition**).

## 2: Descriptions

A **Description** is an entity gathering the pair **Construction** + **Self**. Therefore Constructions may be shared independently from the properties of Self.

A same Description may be shared by different Usyns.

A Description may not bear a Construction if one wants to describe a behavior of called without specifying the context of occurrence.

On the other hand, Self is compulsory.

A Description may bear 0 to N **Conditions**, that are also part of its definition.

A same Description may be a Base Description for a Usyn, yet a Transformed Description for another.

## 3: Self

The entity **Self** allows to record the properties connected with the described entry when this entry occurs in a given syntactic construction. Self does not point at the Um it figures and may consequently be shared by different Descriptions.

### 3.1: Simple Self

The properties of Self are attached to it for a Construction (**IntervConst**), or received from an element external to its Construction (**ComportAppele**). IntervConst and ComportAppele are additional non exclusive information. The first, the second or both elements may be documented, but at least one of them has to be documented.

#### 3.1.1: Self intervening in the Construction (IntervConst)

An **IntervConst** allows to specify and constrain Self as Intervener in the Construction. It gathers:

- **terminal Phrases**

These are pairs (syntagmatic label/list of features) describing the syntagmatic realization(s) of Self. All types of restricting features may apply to Self, except lexical features.

*Ex : **s'•tre moquŽ / avoir moquŽ***

(to have laughed at)

```
Self :

IntervConst :

syntagme_t_l :

V[Pronominal:SE][Aux:ETRE]

V[Aux:AVOIR]
```

- **a Function**

Function taken by Self as an intervener in the construction, the value of which is "TETE" if a major behavior of Self is described, or any other function received from an element of its construction if one of its minor behaviors is described.

GENELEX proposes a list of values that may be enriched, modified or entirely overhauled:

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

A function is not obligatorily documented (attribute *fonction*).

*Ex : La **majoritŽ** des N*

(most of N)

```
Self :

IntervConst :

syntagme_t_l : N[Nombre:SINGULIER]

fonction : SPECIFIEUR
```

- **Thematic Roles**

Self may bear in itself one or several Thematic roles (**RoleTh**) in a given Construction.

GENELEX proposes a list of values for Thematic Roles that may be enriched, modified or entirely overhauled:

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

Thematic roles are not obligatorily documented (attribute *roleth_l*).

### 3.1.2: Self called by an element external to its Construction (ComportAppele)

The same type of properties is noted on a **ComportAppele** as on an IntervConst, yet these properties are expressed on Self by an element not described in the Construction. Contrary to IntervConst, a ComportAppele only references one terminal Phrase. All restricting features are allowed, except lexical features. A same Self may have several different behaviors of called (attribute *comportappele_l*).

```
Ex : Self :

comportappele_l :

syntagme_t : ADJ

fonction : ATTRIBUT_SUJET
```

## 3.2: Compound Self

The distinction between Self intervening in an external construction and Self functioning as a called also applies to compound syntactic units. But the category indicated in *IntervConst* and *ComportAppele* corresponds to the (external) functional category of the compound.

*Ex : mettre en marche (VERBE)*

What is specific to compound syntactic Units is the fact that Self is not atomic, yet composite (multi-lexical). To document the Internal Structure of Self, that is a partially or entirely lexicalized Phrase, its field *syntagme_nt_s* is documented by a Phrase (**Syntagme_NT_S**) that must at least have a syntagmatic label. This Phrase may also be rewritten, if necessary.

*Ex : mettre en marche (SV)*

# 4: Constructions

## 4.1: General definition

A **Construction** describes the context or syntactic frame specific to the entry described.

It enters in the definition of a Description. If it is a Base Description, its field *squelettique* must have the value NON.

A same Construction may be shared by different Descriptions.

A Construction is basically defined by an ordered list of **Positions** (attribute *position_c_l*) that comprises at least one element.

In a "syntactist" vision, in which Constructions correspond to Phrases, it is possible to associate them with a **non-terminal syntagmatic label** (attribute *etiquettesynt*) and the set of restricting features allowed for this Phrase (attribute *trait_l*).

The fields *optionnalite*, *insereself* and *solidarite* take also full part in defining a Construction :

- **Optionality**

The field *optionnalite* must always be documented, i.e. it must reference an element **Optionnalite** that comprises at least a wording. This wording gives the ordered list of **Positions** (as stored in the attribute *position_c_l*) and indicates in brackets which **Positions** are optional.

The optional character of each **Position** is not attached to it because it depends upon the Construction in which it occurs.

**Positions** are identified by a Symbol Pi. The **Position** rank is a relative information, not attached to the **Position**. Nevertheless, in the GENELEX model, since a **Position** is obligatorily referenced in the context of a Phrase or a given Construction, it will always be documented by its rank.

```
Ex : P0 P1 (P2)
```

Self is not in this list.

- **Insereself**

  - Atomist vision

  This field is not obligatorily documented.

  Since the list of **Positions** only describes the complementation pattern of **Self**, and never a level of syntagmatic rewriting, it is always on the same syntactic level as **Self**.

  - Syntactist vision

  In this vision, the lists of **Positions** are lists of syntagmatic rewriting.

  When the list of **Positions** corresponds to the rewrite level of **Self**, this filed must obligatorily be documented.

  If **Self** does not enter into this rewrite level, then this field is obligatorily not documented. This list of Positions has a higher or lower hierarchic level than **Self** and describes a partially restricted prototypic syntactic context, and not the complementation of **Self.**

The field *insereself* indicates the insertion point of **Self** at the beginning, in the middle or at the end of the list of **Positions**. The number given is the index of the **Position** BEFORE which **Self** inserts. If **Self** inserts at the end of the list, the index is that of the last 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
```

On P, the insertion point of **Self** is not documented.

On SADJ, it is 1 (0+1).

- **Interdependency**

The text field *solidarite* is optional. It indicates with a dash couples that are dependent from one another, i. e. the elements Position, Self, or Insertion that are an undividable sequence.

```
Ex : SELF-P1-P2
```

## 4.2: Construction Skeletons

The Transformation Modes between Constructions (**ModifConstruction**) use Construction skeletons, i.e. Constructions that are fully or partially documented and filled from the source Construction.

These Construction skeletons are identical to true Constructions, except that they must have a field *squelettique* with the value OUI, that they authorize their **Positions** to have a non-documented distribution and that they have to be connected with a ModifConstruction that allows to obtain a result Construction from a source Construction.

# 5: Positions

**Positions** are defining criteria for Constructions. They identify a syntagmatic paradigm required by an entry, and allow to associate a **Function** and thematic roles (**RoleTh**) with this paradigm. Typically, a verb complement is defined by a **Position**.

**Positions** are defined by a set of three data: distribution, function and thematic roles.

A distribution is a list of 1 to N **Phrases** (attribute *syntagme_c_l*).

Function and thematic roles (attributes *fonction* and *roleth_l*) are not obligatorily documented.

Besides, one indicates on each **Position** whether it can be repeated or not (field *repetable*, value OUI or NON).

Last, a **Position** may indicate which Phrases in its distribution have a relation of Transformation (attribute *transfsyntagme_l*).

**Positions** are usually identified by their rank in the list of **Positions** under the Construction. However, the rank of the **Position** is not a defining element. It is only a value taken in a context - i.e. in the list of **Positions** in which it occurs.

**Positions** may be shared by a **Construction** or another.

We will see subsequently that they also make it possible to rewrite **Phrases**.

Only the **Positions** that take part in **Construction** skeletons (field *squelettique* with the value OUI) may have an empty distribution.

## 6: Insertions

**Insertions** deal with re-entering phenomena (with a meaning other than in unification grammars) between the inside and the outside of the compound, i.e. cases where a **Position** of the external **Construction** appears in surface among the **Positions** of the internal Structure.

*Ex :* SN ***mettre*** SN ***en Ïuvre***

Therefore **Insertions** only occur in **Phrases** describing the Internal Structure of a compound (**Syntagme_NT_S,** attribute *insertion_l*). They are inserted in their correct place in the list of **Positions** describing a structure at a N level of its rewrite tree (location indicated in the field *insereinsertion_l*).

An Insertion is consequently both a pointer at a Position of the external Construction and a mark for the insertion point. There may have several Insertions. As for Positions, they are numbered from 0 on.

Some **Insertions** are obligatory, whereas they are only possible in most cases (attribute *obligatoire* with the value OUI or NON).

## 7: Conditions

### 7.1: General properties

The aim is to be able to express **Conditions** on **Positions** to constrain their option or the whole of their possible occupants. To do so, one uses the logic implication expressed by the predicates **Si** (If) and **Alors** (then). Since implications are oriented, one has to be careful to express as many Conditions as required by the phenomenologic coverage.

Predicates Si and Alors allow to select a **Position** or a phrase, depending on needs.

Conjunction, disjunction and negation of Predicates may be expressed:

- **Conjunction** is expressed by a list of Predicates (list of Si, list of Alors). A Conjunction of Si is a set of surface realizations to be fulfilled. A conjunction of Alors is the set of the consequences resulting in surface: optional position(s) obligatorily realized or absent, selection or exclusion of a subset of Phrases that may occupy a **Position**.

- **Disjunction** is expressed by a list of Conditions. It is to be reminded that actually both formulas are equivalent:

```
(1) Si (A ou B) Alors C

If (A or B) Then C

(2) (Si A alors C) et (Si B Alors C)

(If A Then C) and (If B Then C)
```

In GENELEX, formula (2) is used.

- **Negation** is expressed by the field *negation* (OUI, NON) on each Predicate **Si** or **Alors**. The negation of a **Position** indicates that it is not realized in surface. The negation of a **Position** occupant indicates that this occupant does not actualize the **Position**.

Here are the two types of **Conditions** to be differentiated: Conditions on the **option** of Positions and Conditions on the **realizations** of Positions.

## 7.2: Conditions on the option of Positions

These Conditions (**ConditionOpt**) allow to express constraints on the option of **Positions**, using the predicates **SiOpt** and **AlorsOpt**. It will then be possible to give account of phenomena such as :

- **Positions** that are optional together: the absence of the one **Position** involves the absence of the other.

```
Ex : P0 (P1) (P2)

Cond : Si !P1 Alors !P2

Cond : Si !P2 Alors !P1
```

- **Positions** that are mutually exclusive: the presence of the one **Position** involves the absence of the other and vice-versa.

```
Ex : P0 (P1) (P2)
```

```
Cond : Si P1 Alors !P2

Cond : Si P2 Alors !P1
```

- Any other more or less complex constraint connected with the fact that an optional **Position**, when realized (or not) in surface, may have consequences on the actual realization of the other optional **Positions**.

These **ConditionOpt** select **Positions** using their rank (field *nieme_position* de **SiOpt** et **AlorsOpt**). They are borne by the element Optionality that may have zero or several ConditionOpt to be fully documented (refer to Positions that are optional together or mutually excluding). It is to be reminded that Construction and Syntagme_NT_C have a field *optionnalite*.

## 7.3: Conditions on the realizations of Position

Some realizations of **Positions** are inter-conditioned, i.e. if Pi is realized in such or such a way, then Pj can only be realized in such or such other way. Therefore, each realization of **Position** may exclude or, on the contrary, select a subset of **Phrases** that can occupy another **Position**.

Mutual selection and exclusion must always be expressed by a double implication.

```
Ex : Si Pi == SyntagmeX Alors Pj = SyntagmeY

Si Pj == SyntagmeY Alors Pi = SyntagmeX
```

A predicate selects:

- a Phrase or a **Position** of the external **Construction** (value EXTERNE of the field *portee* ),

- a **Phrase** or a **Position** of the structural **Phrase** of **Self** (value INTERNE of the field *portee*), if the Usyn is compound,

- a realization of Self as an intervener in the Construction (value INTERVENANT of the field *portee*).

To select or exclude, proceed as follows:

- for a Phrase, using **CheminSyntagme**:

  this latter selects a Position and one of its Phrases (attributes *nieme_position* and *syntagme*) then, if necessary, it goes further down in rewriting using a recursive link on another CheminSyntagme.

- for a Position, using **CheminPosition**:

  this latter selects a Position (attribute *nieme_position* of **PositionBut**) after going further down in the rewriting by a CheminSyntagme, if necessary.

- for a realization of Self as an intervener in the Construction, using **SelectIntervConst** :

  this latter selects one of the Phrases referenced by IntervConst (attribute *syntagme_t*).

These Conditions are borne by Descriptions.

# 8: Phrases

**Phrases** occur :

- in the caracterization of **Self**,
- in the distribution of a **Position**,

**Phrases** may be shared.

A **Phrase** is defined by:

- its **syntagmatic label** (attribute *etiquettesynt*),
- the list (that may be empty) of its **restricting features** (attribute *trait_l*),
- if it is a non-terminal Phrase, the list (that may be empty) of the **Positions** that describe its rewriting (in a syntactist vision ) on the first level (attribute *position_c_l* or *position_s_l*)
- an attribute *listepositions* (*OUVERTE, FERMEE*) that indicates whether the list of rewrite Positions is closed or open, in other words, if the rewriting of a Phrase is fully oir partially specified. When the list of Positions is empty, it is of course an OPEN list.

Here are the two subcategorizations of Phrases to be differentiated:

1. Terminal Phrases (**Syntagme_T**) the syntagmatic label of which is a morpho-syntactic category and non-terminal Phrases (**Syntagme_NT**). Terminal Phrases do not allow any of the features Prep, Conj, ProRel, ProIntrog, Introd, so that the only lexical feature allowed is the feature **Lex**. This rule applies in reflection of the features of the family RefLex (see paragraph on features).

2. Phrases entering in the definition of a Construction Position (**Syntagme_C**) and Phrases entering in the definition of a Structure Position (**Syntagme_S**). Only Syntagme_S allow Insertions in their list of Positions as well as lexical features from the family RefLex on their Phrases. In both cases, the rewrite list is an ordered list. The order of elements in the list corresponds to the canonical order. These elements may subsequently be referenced by their rank in the list, independently from their type **.**

These two distinctions may of course be combined (**Syntagme_T**, **Syntagme_NT_C**, **Syntagme_NT_S**).

The syntagmatic labels proposed by GENELEX are as follows:

- **Labels for non-terminal Phrases**: P, SN, Nbarre, SV, SADJ, SADV, SP.
- **Labels for terminal Phrases**: NOM, ADJECTIF, ADVERBE, VERBE, PREPOSITION, CONJONCTION, PARTICULE, DETERMINANT, INTERJECTION, PRONOM.

- **e category**: or "ŽlŽment vide" (empty category) in generative grammar, applicable to a terminal Phrase, that allows to express traces, thus it may bear all restricting features, except features concerning morphological or syntactic sub-category.

It is a minimum list that guarantees genericity. The richness of the restricting features that may be associated with the syntagmatic label make up for the poorness of this list. Users may nevertheless enrich this list with new values; the only drawback is that this list will then lose in genericity.

Phrases may be described as N-depth syntactic trees. In GENELEX, a syntactic tree is expressed by the embedding of lists of Positions as follows: a Phrase is rewritten on the first level by a list of Positions, each Phrase that occupies a Position may itself be rewritten by a list of Positions, and so on. If a Phrase is rewritten by a list of Positions, all the above-mentioned comments on the fields *optionnalite, insereself* and *solidarite* of the Construction apply.

# 9: Features

Combined to the different Phrases, **restricting features as such** should allow to specify any Position occupant with as much accuracy as possible

`Ex: P[SsCatSynt:RELATIVE] =>` relative

`P[Mode:INFINITIF]` => infinitive

`P[SsCatSynt:COORDONNE]` => phrase coordonnŽe

`P[SsCatSynt:SUBORDONNEE]` => subordonnŽe

`P[SsCatSynt:COMPLETIVE]`

`[Mode:SUBJONCTIF]` => complŽtive

Features are given a graphic form (user interface) that does not indicate anything on their internal structure. Features are recorded in square brackets, within which the name of the feature is first specified, followed by a colon, and its value. Only some features require a more complex notation: it is the case of Aux and of all features belonging to the RefLex family.

*Ex: Trait_Genre*

```
[Genre:FEMININ]
```

*Trait_Aux*

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

*Trait_RefLex*

```
[Reflex:1.3]
```

The name of the Feature given in appearance is deduced from its type or specified when creating the Feature.

*Ex: Trait_Personne*

**name** : Personne

*Trait_Libre*

**name**: couleur

*Trait_Bin*

**name**: animŽ

## 9.1: Lexical Features

## 9.1.1: Family of Lex features

These features give lexicalization information in specifying a Um by its written form and/or its identifier (which removes any homography problem). This direct (SGML identifier) or indirect (written form) reference to a Um is done using two fields:

**value:** *string*

**um:** *pointeur*

**Note**: a gap (based on the distance between the morpho-syntactic category and the functional category) is allowed between the category of the Phrase that bears Trait_Lex and the category of the Um referred by Trait_Lex.

```
Ex : Um id:um04 CatGram:ADJECTIF Umg:courageux

NOM[Lex:courageux]
```

Within this family, one has to differentiate introducers and the Lex feature itself.

1. **Introducers: Prep, Conj, ProRel, ProIntrog, Introd**

These features lexicalize the introducers for non-terminal Phrases.:

- Prep for the preposition that recategorizes a SN, introducing a Phrase or being the first element for rewriting a Prepositional Phrase.

- ProRel for the relative pronoun introducing a Sentence.

- ProIntrog for the interrogative pronoun introducing a Sentence.

- Conj for the conjunction introducing a Sentence.

- Introd for any other introducer of a non-terminal Phrase that would have not been provided for.

2. **Lex**

This feature lexicalizes:

- a syntagmatic leaf, if it applies to a terminal Phrase

- the head of a Phrase, if it applies to a non-terminal Phrase. Since introducers are available, it is to be assumed de facto that the syntagmatic heads are:

N for SN

N for SP

ADJ for SADJ

ADV for SADV

V for SV

V for P if we assume that P=V''' in the X-bar theory.

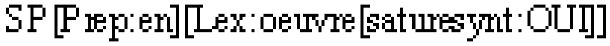
The attribute *saturesynt* of the **Lex** feature indicates whether the lexicalizing element is or not (including an eventual introducer), the only leaf of that Phrase. Consequently, this attribute:

- must always have the value *OUI* (default value) if the phrase is terminal,

- may have the value *OUI* or *NON* for non-terminal phrases for which it is intended.

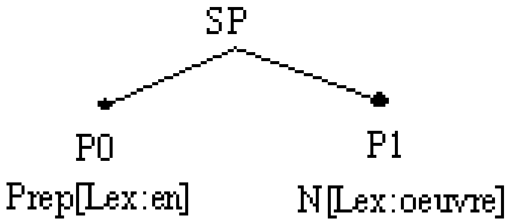
*Ex : en Ïuvre*

*Solution 1 :*



SP[Prep:en][Lex:oeuvre[saturesynt:OUI]]

*Solution 2 :*



## 9.1.2: Family of the Reflex Features

These features are defined in reflection of the features of the Lex family. Therefore this family comprises the following features: **RefLex, RefPrep, RefProRel, RefProIntrog, RefConj** and **RefIntrod**.

What differentiates them is:

- the mode for accessing units,

- the type of accessed units that are lexicalizing (Um) and/or structuring (Usyn),

- tha fact that they are only used within the Structures describing the inside of syntactic compounds.

The attributes *nieme_cposition* and *nieme_cposant* reference by their indices Ums or Usyns contained in a two-level list: list of alternatives (**Composition**) borne by the compound Usyn, list of components (**R_ComposeUm, R_ComposeUsyn**).

**nieme_cposition:** *integer*

**nieme_cposant:** *integer*

*Ex : ((avoir admiration pour)(porter admiration ˆ))*

```
RefLex[2.1] => porter
```

## 9.2: Morphological features

- **Trait_Mode**

**value**: INDICATIF, SUBJONCTIF, CONDITIONNEL, PARTICIPE, INFINITIF, IMPERATIF.

```
Ex :[Mode:INFINITIF]
```

- **Trait_Temps**

**value**: PRESENT, IMPARFAIT, FUTUR, PASSE, PASSE_SIMPLE, COMPOSE.

`Ex :[Temps:PASSE]`

The tense feature allows to express tense restrictions relating to certain constructions.

- **Trait_Personne**

**value**: 1, 2, 3

`Ex :[Personne:1]`

- **Trait_Genre**

**value**: masculin, fŽminin, neutre.

`Ex :[Genre:FEMININ]`

- **Trait_Nombre**

**value** singulier, pluriel.

Ex :`[Nombre:SINGULIER]`

- **Trait_NombrePosseur**

**value**: SINGULIER_POSSEUR, pluriel_POSSEUR.

Ex :`[NombrePosseur:SINGULIER_POSSEUR]`

# 9.3: Morpho-syntactic features

- **Trait_Accord**

**value**: I, J, K, L, NON_I, NON_J, NON_K, NON_L.

```
Ex : [Accord:I]
```

*cf. Coref*

- **Trait_SsCatMorph**

**value**: 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**

**value**: ETRE, AVOIR.

**tense**: PRESENT, IMPARFAIT, FUTUR, PASSE, PASSE_SIMPLE, COMPOSE.

**mood**: INDICATIF, SUBJONCTIF, CONDITIONNEL, PARTICIPE, INFINITIF, IMPERATIF.

This feature allows to associate with a given verb (the entry described or a verb in the context of the entry), its auxiliary(ies) allowed for a use.

*Ex : **se lever** (•tre levŽ) // **lever** (avoir levŽ)*

```
V[Lex:lever] V[Lex:lever]

[Aux:ETRE] [Aux:AVOIR]

[Pronominal:SE]
```

The occurrence of Trait_Aux along with Trait_Temps that bears the value COMPOSE indicates that the auxiliary verb is of necessity present in the use under description.

*Ex : **•tre arrivŽ** socialement*

```
V[Lex:arriver]

[Aux:ETRE]

[Temps:COMPOSE]
```

The attributes tense and mood of Trait_Aux specify the tense and the mood of the auxiliary, if necessary.

*Ex :* ***Žtant donnŽ***

```
V[Lex:donner]

[Temps:COMPOSE]

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

- **Trait_Pronominal**

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

This feature allows to associate with a given verb (the entry described or a verb in the context of the entry) the NON-REFERENTIAL preverbal particle that must be associated with a use.

*Ex :* ***s'en aller***

```
V[Lex:aller]

[Pronominal:SE_EN]
```

- **Trait_Passif**

**value**: PLUS, MOINS.

```
Ex : [Passif:PLUS]
```

## 9.4: Syntactic Features

- **Trait_Neg**

**value**: 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.

The presence of a Trait_Neg indicates that the use described is in the negative form. It is also possible to specify in the "valeur" field a restriction on the lexicalization of the negation

```
Ex : [Neg:NE_PAS]
```

- **Trait_SsCatSynt**

**value**: 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**

**value**: INTERROGATIVE, EXCLAMATIVE.

```
Ex : [Tournure:INTERROGATIVE]
```

## 9.5.: Syntactic-semantic Features

- **Trait_Coref**

**value**: I, J, K, L, NON_I, NON_J, NON_K, NON_L

Ex `: [Coref:I]`

Co-reference indices may be used at any time. Like any other features, they may apply to any Position occupant. But it must always be possible to solve coreference: if a feature with the value I exists, there is at least another matching feature with the value I or NON_I.

Coref features do not force the co-realization of the Phrases that bear them; if one wants to impose this co-realization, it will be done as usual by using Conditions.

When a Coref feature applies to the category P, it implicitly indicates the subject of P.

*Ex : Jean* ***dit*** *partir en vacances*

```
cb : P0 SELF P1

P0 : SN[Coref:I]

...

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

...
```

## 9.6: Semantic Features

- **Trait_Aspect**

**value**: STATIF, PROCESSIF, RESULTATIF.

```
Ex : [Aspect:STATIF]
```

- **Trait_Libre**

**name**: *string*

**value**: *string*

This type of feature may be used to specify semantic classes or families.

```
Ex : [couleur:rouge]

color: red

[classe:v•tement]

class: clothes
```

- **Trait_Bin**

**name**: *string*

**value**: PLUS, MOINS.

This type of feature allows to express "denotative conditions" for instance.

```
Ex : [animŽ:PLUS]
```

# 10: Transformations

The attributes *appellation*, *commentaire*, *exemple* borne by **Transf** gloss the transformations themselves and not the element to be transformed.

## 10.1: TransfUsyn and TransfDescription

The Transformation process described by TransfUsyn and TransfDescription is basically the same. The three following phenomena are covered:

1: modification of the realizations of Self. (attribute *modifintervconst*)

```
Ex : Self : IntervConst : V
```

**--passivation-->**

Self : IntervConst : V[Aux:ETRE]

[Temps:COMPOSE]

[Passif:PLUS]

2: modification of the Construction (attribute *modifconstruction*).

Ex : P0[RoleTh:AGENT] P1[RoleTh:THEME]

SN SN

**--passivation-->**

P0[RoleTh:THEME] P1[RoleTh:AGENT]

SN SP[Prep:par]

3: restructuring of the structural Phrase describing the inside of a compound (attribute *modifsyntagme_nt*)

*Ex : pour les beaux yeux de SN*

-> *pour ses beaux yeux*

Nevertheless, in the case of TransfDescription, the work is done under the same Usyn, whereas in the case of TransfUsyn the work is done under two different Usyns, for linguistic reasons.

1. **TransfUsyn**: Transformations operating between two Syntactic Units

• stemming from the same Um,

Ex : neutrality

• stemming from different Ums

Ex : syntactic derivation

2. **TransfDescription**: Transformation operating between two Descriptions (i.e. two pairs Self / Construction).

Ex : passivation

Besides, formally TransfUsyn and TransfDescription are both pointed at by the source Usyn. Yet:

• TransfUsyn points at the result Usyn

- TransfDescription points on the one hand at the source Description (base or transformed to give account of Transformation linkings) and on the other hand at the result Description.

Besides, the constraints on the cardinality of lists are different:

- a Usyn may bear 0 to N TransfUsyn for which it is the source.

- the cardinality of the list of TransfDescription (*transfdescription_l*) is greater than or equal to that of the transformed Descriptions (*description_l*). If N is the number of transformed Descriptions, then the number of TransfDescription is lower than or equal to the number of possible combinations of two descriptions among N+1 (since the base must also be taken into account).

## 10.2: TransfUsyn

**TransfUsyn** connects a source Usyn, given in the call context of TransfUsyn, and a result Usyn pointed at by the attribute*usyn_resultat* of TransfUsyn.

**TransfUsyn** may make it possible to give information on the SHIFT from the source to the result, documenting a ModifDescription on the base Description of the source Usyn.

## 10.3: TransfDescription

**TransfDescription** establishes a relation of transformation between a source **Description** and a result **Description**, respectively pointed at by the attributes *description_origine* and *description_resultat*.

As regards the Transformation of the **Construction** contained in the **Description**, it si possible to choose between a descriptive mode (minimum use) or a computation mode (maximum use) :

- in the descripton mode, the **Construction** pointed at by the result Description is fully described: **ModifConstruction** establishes a correspondence between Positions or Phrases at any hierarchic levels between the source **Construction** and the result **Construction**;

- in the calculation mode, the **Construction** pointed at by the result **Description** is declared *squelettique*: The result **Construction** is not really recorded. Only a squeleton of result is recorded. This latter is enriched with elements or attributes stemming from the source Construction or given in **ModifConstruction**. It is to be noted that in the computation mode, it is impossible to record transformation linkings.

## 10.4: TransfPosition

**TransfPosition** establishes a relation of transformation between a source **Position** and a result **Position**, respectively pointed at by two **CheminPosition**.

To transform a Position (**TransfPosition**) in the computation mode, one has to identify the source Position (first CheminPosition) and a Position of the squeleton (second CheminPosition) between which one wants to establish a relation of Transformation. The number of TransfPosition is lower than or equal to the number of Positions of the squeleton.

# 10.5: TransfSyntagme

Phrases occupying a same Position or characterising Self may have Transformation relations, the most typical of which being pronominalization and the reduction of a completive into an infinitive clause.

Ex : P0 ***aimer*** P1

P1 : SN

PRONOM[Lex:le]

[SsCatMorph:PERSONNEL_FAIBLE]

P[Mode:INFINITIF]

PRONOM[Lex:•a]

Since these transformational properties are only valid in the context of a Position (it is not a property of the Phrase in itself), they are borne by the Position and not by the Phrase; the Position indicates which pairs of Phrases that occupy it are connected to one another by a Transformation relation.

If N is the number of Phrases entering in the same deistribution, then the number of TransfSyntagme is lower than ot equal to the number of possible combinations of two Phrases among N.

**TransfSyntagme** establishes a relation of Transformation between a source **Phrase** and a result **Phrase**, respectively pointed at by the attributes *syntagme_origine* and *syntagme_resultat* and allows to give information on the SHIFT from the source to the result in documenting a ModifSyntagme on the source (attribute *modifsyntagme*).

**TransfSyntagme** is also used for describing the possible transformation applied to a Phrase when the Construction in which it occurs is transformed. In that case, *syntagme_origine* points at a Phrase of the source Construction and *syntagme_resultat* points at a Phrase of the result Construction.

# 11: Modifs

Elements whose name starts by Modif are used to modify the attributes values of a so-called "source" element (before being modified). A pair Modif + source element (known in context) must allow to deduce a "result" element (after the specified modifications are applied).

All or part of the attributes of an element may be modified according to a protocol based on the type of attribute:

- attributes IDREF(S), CDATA and NUMBER(S). When they are not documented in Modif, they are inherited as such from the origin. When they are documented, the new value supersedes the old one.

- **Special case** for some IDREFS, the double-list mechanism (**retire +**

**ajoute**): when no element is pointed at in Modif, neither by retire (remove) nor by ajoute (add), the origin element(s) is(are) inherited. Otherwise, a new attribute value is obtained by removing or adding element(s) on the basis of the value of the source attribute.

- attribute with enumerated values. When the value is HERITAGE (inheritance) in Modif, these fields are inherited as such from the origin. Any other value supersedes the old one.

## 11.1: ModifDescription

Given a **Description** gathering by definition a **Construction** and a **Self**, **ModifDescription** allows to modify:

- the **Construction**, in calling **ModifConstruction** (attribute *modifconstruction*)
- the **IntervConst** of **Self**, in calling **ModifIntervConst** (attribute *modifintervconst*)
- the **Phrase** representing the internal Structure of a compound **Self**, in calling **ModifSyntagme_NT** (attribute *modifsyntagme_nt*).

## 11.2: ModifConstruction

Given a **Construction**, **ModifConstruction** allows to fully modify it, i.e.:

- rewrite its gloss that covers the attributes *appellation*, *commentaire, exempl*e ;
- modify its *etiquettesynt*, *optionnalite* and *solidarite* ;
- specify the rstricting features to be withdrawn and/or added (*ajoute_trait_l, retire_trait_l*) ;
- change the insertion point of **Self** (attribute *insereself*) ;
- withdraw (attribute *retire_position_l*) or modify (element **TransfPosition**) **Positions**. In that case, one has to make sure that the attributes *optionnalite, insereself* and *solidarite* are consistent with the new list of **Positions**.

## 11.3: ModifPosition

Allows to fully modify a **Position**, i.e. :

- its gloss that covers the attributes *appellation*, *commentaire*, *exemple* ;
- its values of *repetabilite, fonction* and *roleth_l* ;
- its distribution, by:
  - the removal and/or addition of **Phrases**,

(attributes *retire_syntagme_l* and *ajoute_syntagme_l*),

- the withdrawal and/or addition of **Transformations** between these **Phrases**,

(attributes *retire_transfsyntagme_l* and *ajoute_transfsyntagme_l*).

- the modification of the Phrases that occupy it by Transformation on these **Phrases** (attribute *transfsyntagme_l*).

## 11.4: ModifSyntagme

Given a **Phrase** indicated by the call context of **ModifSyntagme**, **ModifSyntagme** allows:

1. for any type of Phrase (**Syntagme_T, Syntagme_NT, Syntagme_NT_S**) :

- to specify the restricting **Features** to be withdraw and/or added (attributes *retire_trait_l* et *ajoute_trait_l*),

2. For a non-terminal Phrase (**Syntagme_NT**) :

- to modify its etiquettesynt, its *optionnalite* and its *solidarite* ;

- to withdraw (attribute *retire_position_l*) or modify (element **TransfPosition**) rewrite **Positions**, if any. In that case, one has to make sure that the attributes *optionnalite, insereself* and *solidarite* are consistent with the new rewriting.

3. For a Construction Phrase (**Syntagme_NT_C**) :

- to change the insertion point of **Self** (attribute *insereself*) ;

4. For a Structure Phrase (**Syntagme_NT_S**) :

- to withdraw (attribute *retire_insertion_l*) or modify (element **TransfInsertion**) **Insertions**, if any.

- to change the insertion point of Insertions, if any (attribute *insereinsertion_l*).

A **ModifSyntagme** makes virtually a copy of the **Phrase** to be modified. Consequently, if two **ModifSyntagme** operat'e on the same **Phrase**, it is to be noted that they will produce two result **Phrases** on the basis of one source **Phrase**.

## 11.5: ModifIntervConst

Knowing that **Self** is defined by a single **IntervConst** that describes its characteristics as an intervener in the Construction, **ModifIntervConst** allows to:

- modify its *fonction* and its *r™les thŽmatiques*,

- withdraw and/or add **Phrases** to those that define **Self**.

- modify these **Phrases** by **Transformation**.

# 12: R_Compose and Composition

**R_Compose** and **Composition** give account of the lexicalization of compound **Usyns**. **R_Compose** specify the components, **Compositions** specify lexicalization alternatives. Therefore, each **Composition** contains a list of **R_Compose**.

*Ex : [(avoir admiration)*

*(Žprouver admiration)*

*(porter admiration)]*

3 Compositions, each of them comprising 2 R_Compose

These two levels of list are used in the features of the RefLex family that indirectly reference the elements of those lists:

• either in the internal structural Phrase of Self,

• or in the Composition Mode MdC

according to the following syntax:

```
[RefLex:[nieme_composition:i][nieme_composant:j]]
```

The lists are numbered from 1 on, and 0 on nieme_composition (nth composition) is reserved for the selection of the set of alternatives of an element of the list.

```
Ex : RefLex[1.1] = avoir
```

```
RefLex[0.1] = avoir/Žprouver/porter
```

The components pointed at by R_Compose may be of two types:

• Um component,

• Usyn component and its MdC, if any.

To each type of components corresponds a sub-type of R_Compose: R_ComposeUm and R_ComposeUsyn.

The attribute *type* indicates whether the component is only calling, calling and called or only called. The compound head is only calling.

A component of the Um type is always called.

# 13: Composition Mode

## 13.1: MdC

The Composition Mode (**MdC**) allows to specify the mutual organization of component Usyns, by indicating which component saturates which **Position** of another component, and to express constraints of inhibition, filtering, or inheritance of the properties of the Usyn making up that compound.

Given a component of the Usyn type, it is possible to associate it with an MdC to fully or partially inherit its complementation pattern or its occurrence context; this composition mode allows to:

- specify a minima the **Positions** one wants to inherit from and filter, if necessary, these **Positions** and their occupants ;

- specify the lexicalization of these **Positions** by other components (only for a calling behavior) using the features of the RefLex family on Position occupants.

- if necessary, filter the properties of the component Self.

For a syntactic compound on which one wants to specify all this information, there are as manyMdCs as components of the Usyn type.

## 13.2: HeritePosition

Any **Position** not referenced by **HeritePosition** is considered inhibited. **HeritePosition** is necessary to indicate that a **Position** is inherited during composition, and it specifies:

- which is the **Position** concerned (element **CheminPosition**),

- where this **Position** is inherited, i.e. INSIDE or OUTSIDE the compound (attribute *destination*).

- how this **Position** is inherited, i.e. with or without filtering (attributse *optionnel* et *modifposition*).

These modifications on the source **Position** may affect all the attributes that define it as well as its *optionnalitŽ*. They are expressed on **HeritePosition** by :

- the attribute *optionnel* that has the value *HERITAGE*, *OUI*, *NON*. The value *HERITAGE* indicates that the option value defined for the **Position** in the field "optionnalitŽ" of its Construction has to be inherited. The values *OUI* and *NON* delete the option values of the source **Position**.

- the attribute *modifposition* that refers to the element **ModifPosition**. This latter factors the modifications to apply to all the attributes -of a **Position** knowing that any element that is not modified is inherited unchanged.

When a **Position** is inherited inside AND outside the compound, it has to be associated with two **HeritePosition**, one for the inside, the other for the outside.

When a **Position** is only inherited inside the compound, in the atomist vision, it is said to be saturated by the component that occupies it.

If all **Positions** of a higher level than Self (in the syntactist vision, with a rewrite tree) are inhibited, then the level at which they occur us inhibited, without indicating anything on the lower levels. If all Positions of a lower level than Self are inhibited, then the level at which they occur is inhibited and all lower levels than that level are inhibited.

## 13.3: FiltreSelf

Unless otherwise specified, **IntervConst** and **Syntagme_NT_S** of the **Self** of the component **Usyn** are inherited WITHOUT CHANGES during the composition process and there is therefore no inheritance on Self to specify.

If one wants to inherit then by filtering (to modify them slightly when the component fills the compound), one has to define a **FiltreSelf**.

**FiltreSelf** allows to specify the inheritance of **Self** by filtering, which consists in taking back its **IntervConst**s and its **Syntagme_NT_S** (if it is itself a compound) to modify them respectively by **ModifIntervConst** and **ModifSyntagme_NT**.

# 14: Structure (Syntagme_NT_S)

A **Structure** is borne by the element **Self**. It is used for describing the internal structure of a compound.

**Structures** share all the characteristics of **Phrases**, except the following:

- their lists of rewrite Positions are a priori fully specified, the field *listepositions* should therefore always have the value FERMEE

- furthermore they have a list of Insertions (Insertion of **Positions** of the external structure into the internal structure),

- the lexical features borne by the Phrases occupying the Positions belong to the Lex or **RefLex** family.

- their field *solidarite* may have the reference to Insertions but it excludes Self.

Just as **Positions**, **Insertions** are numbered from 0 on. The location of Insertions in the middle of Poistions in noted in the field *insereinsertion_l*

- Atomist vision

This field is not documented.

- Syntactist vision

The field *insereinsertion_l* indicates the insertion point of each *Insertion* in the beginning or middle of the list of **Positions**. The number given to each, respecting their canonical order is the index of the **Position** IN FRONT OF which they fit. If several Insertions are in front of the same **Position**, it means that they follow one another in this order.

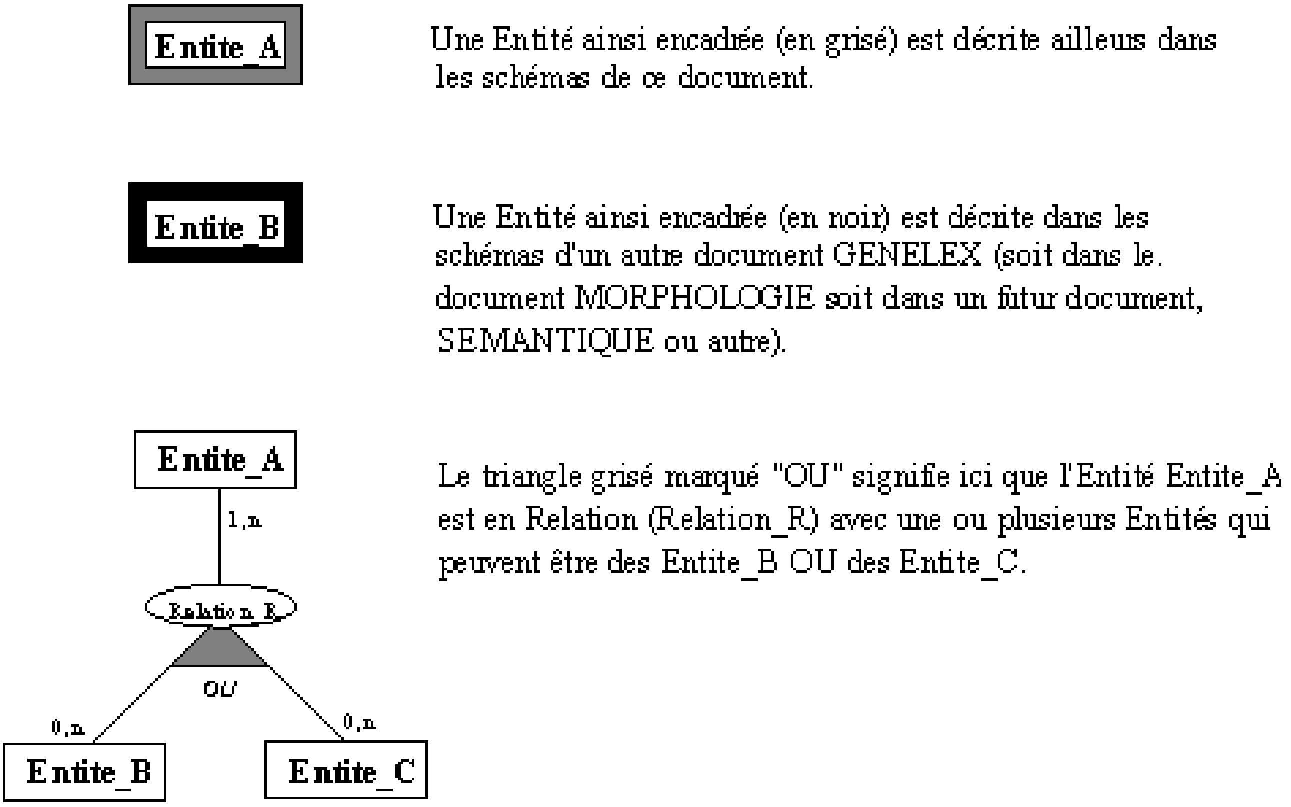
## 15: Appellation, commentaire, exemple

Thanks to the fields *appellation*, *commentaire* and *exemple*, the lexicographer may gloss the element described, speed up access to it and make maintenance easier. The *appellation* allows to associate with an element of GENELEX the name generally used in grammar tradition, the linguistic world or even the editorial team. Thus a construction may bear the *appellation* "transitif direct" (direct transitive). The *commentaire* allows to indicate right from the highest level the specificities of an element without having to go through it. It also makes it possible to record all information relating to the lexicographic work. Thus the status "en cours" (in progress), "ˆ valider" (to be validated), "validŽ" (validated) of an element may be recorded there, just as justification on the coding choice, if any. Last *exemple* should be self-explaining...

# G - Entity-Relation diagrams

**SPECIAL GRAPHICAL CONVENTIONS**

**used in the Entities/Attributes/Relations diagrams of this document:**

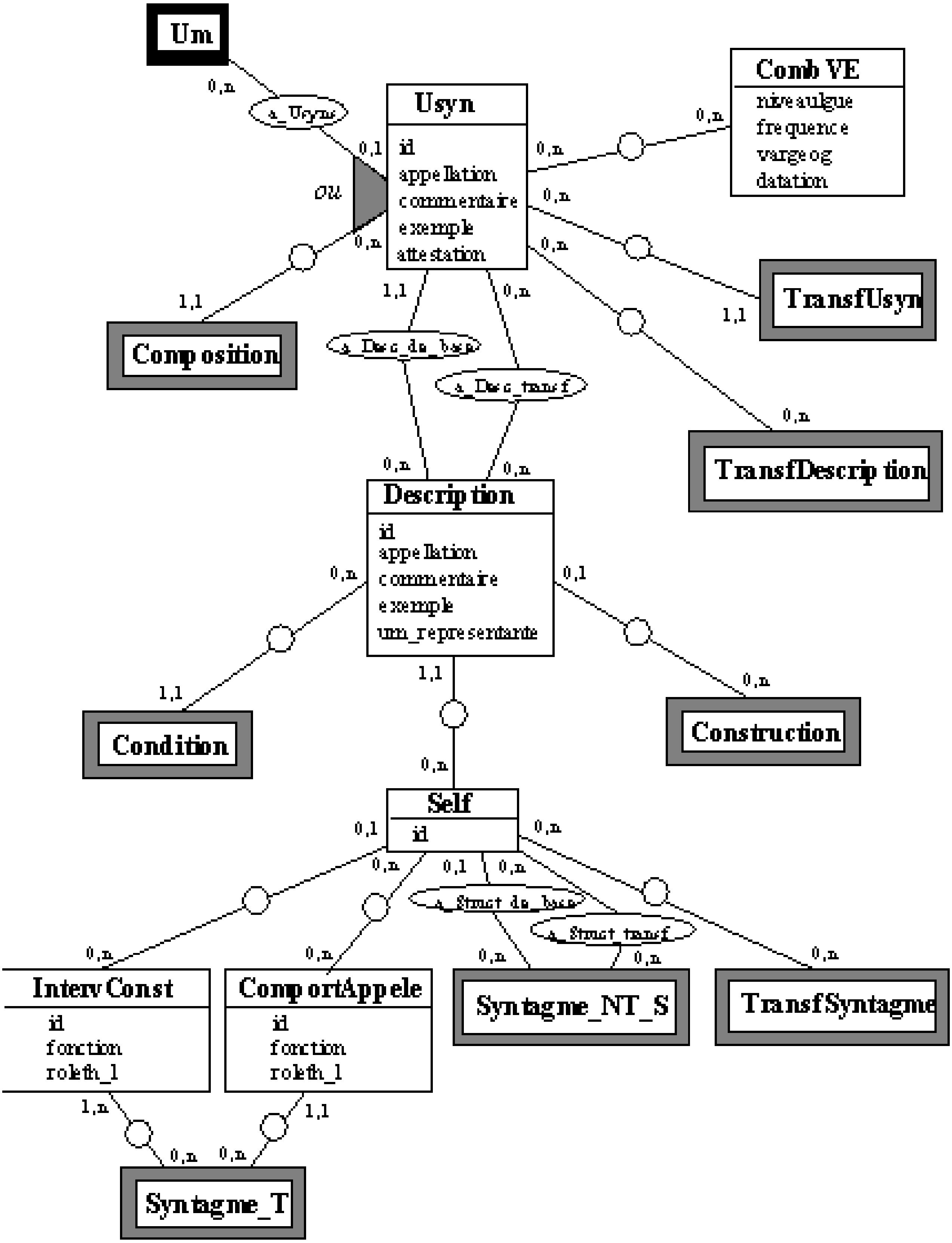


Une Entité ainsi encadrée (en grisé) est décrite ailleurs dans les schémas de ce document.

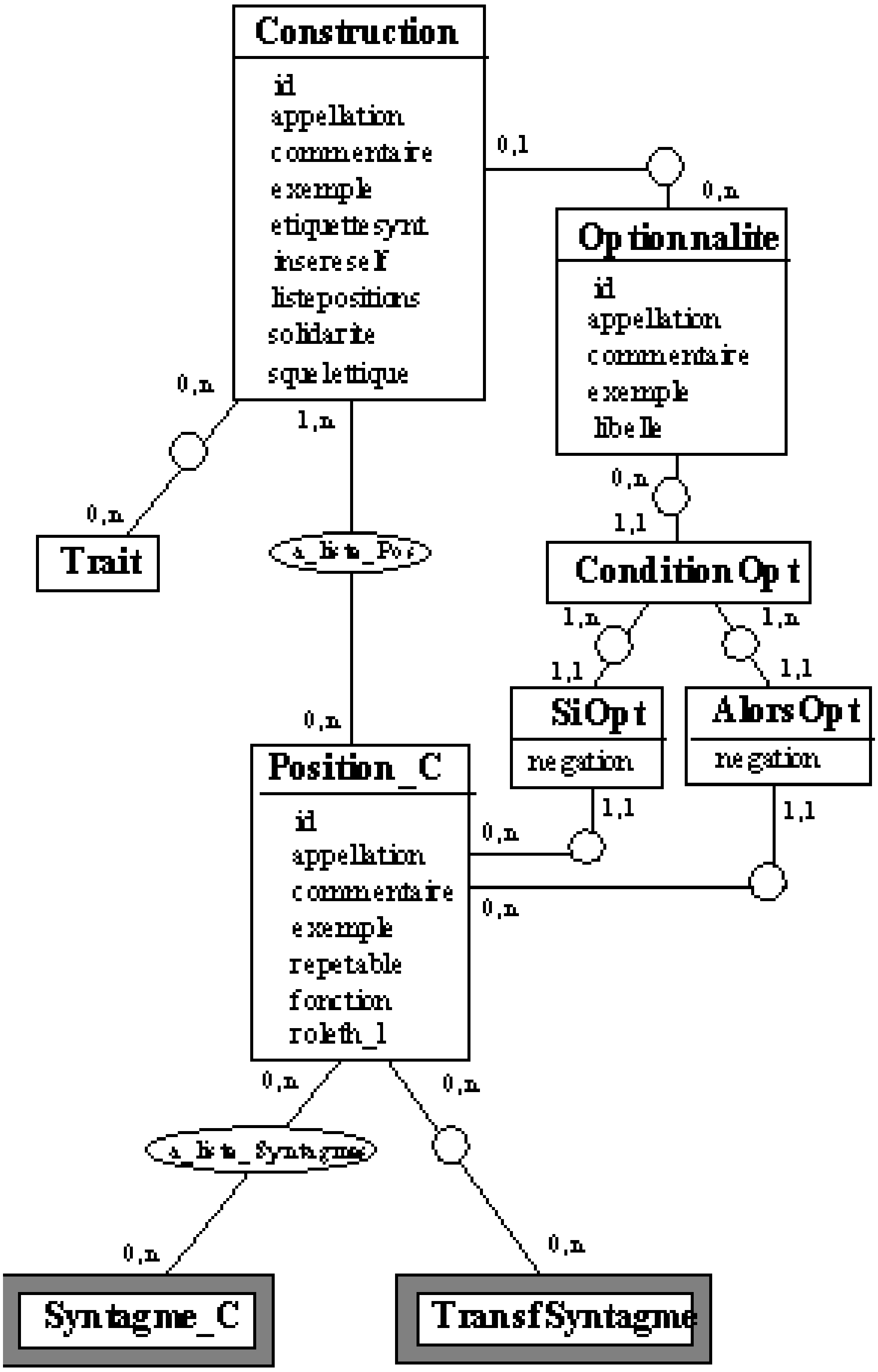
Une Entité ainsi encadrée (en noir) est décrite dans les schémas d'un autre document GENELEX (soit dans le. document MORPHOLOGIE soit dans un futur document, SEMANTIQUE ou autre).

Le triangle grisé marqué "OU" signifie ici que l'Entité Entite_A est en Relation (Relation_R) avec une ou plusieurs Entités qui peuvent être des Entite_B OU des Entite_C.

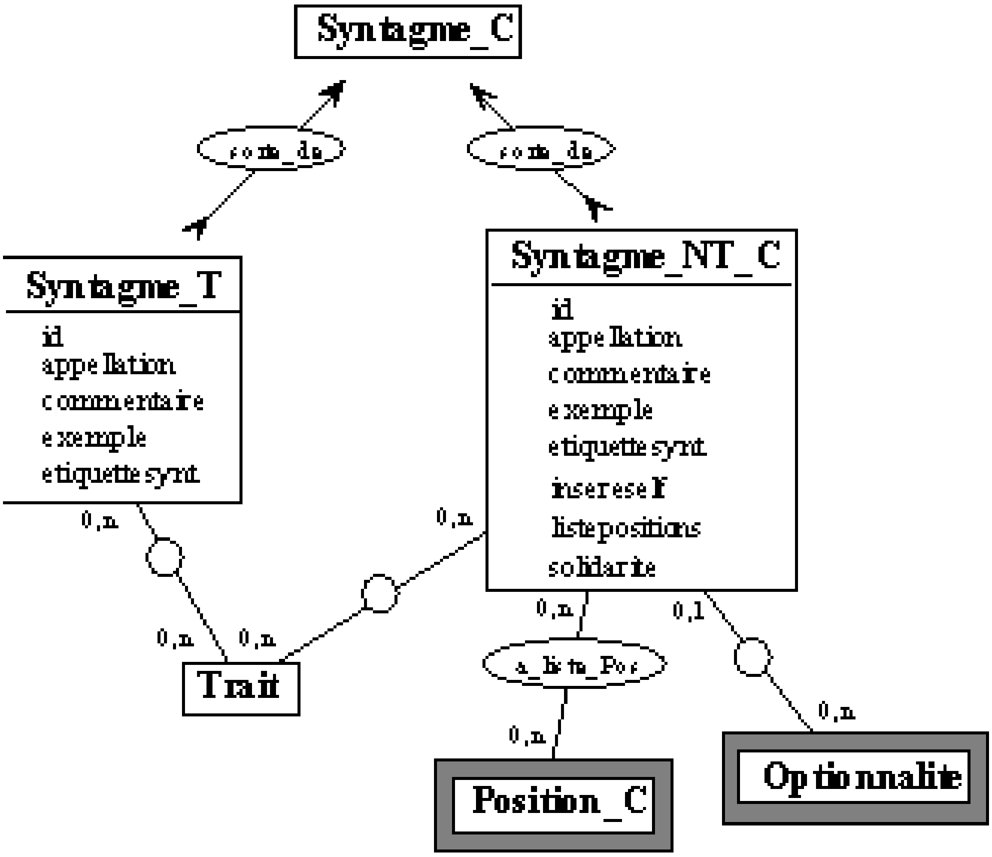
1 : Syntactic Unit, Description and Self



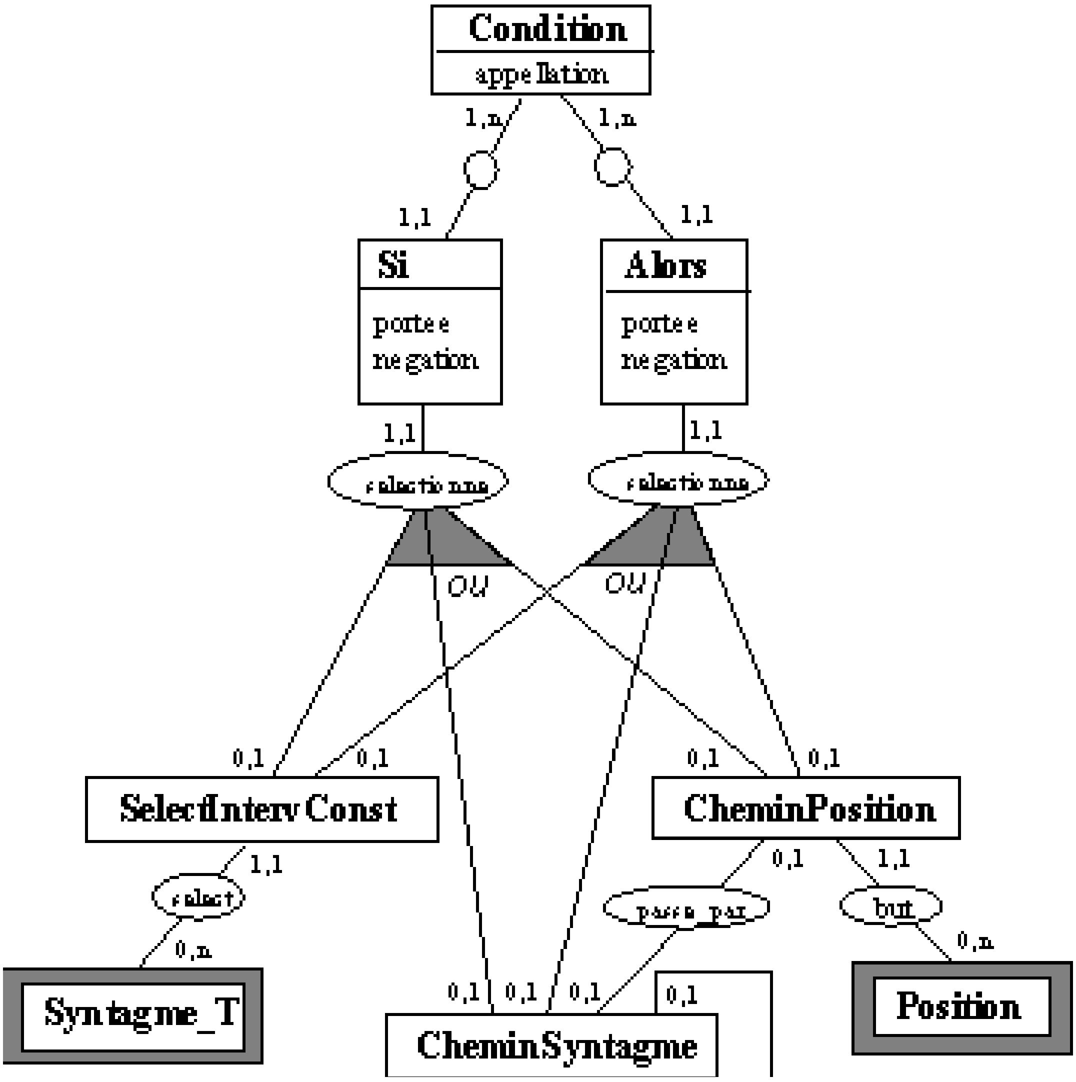
2 : Construction, Position_C and Optionality

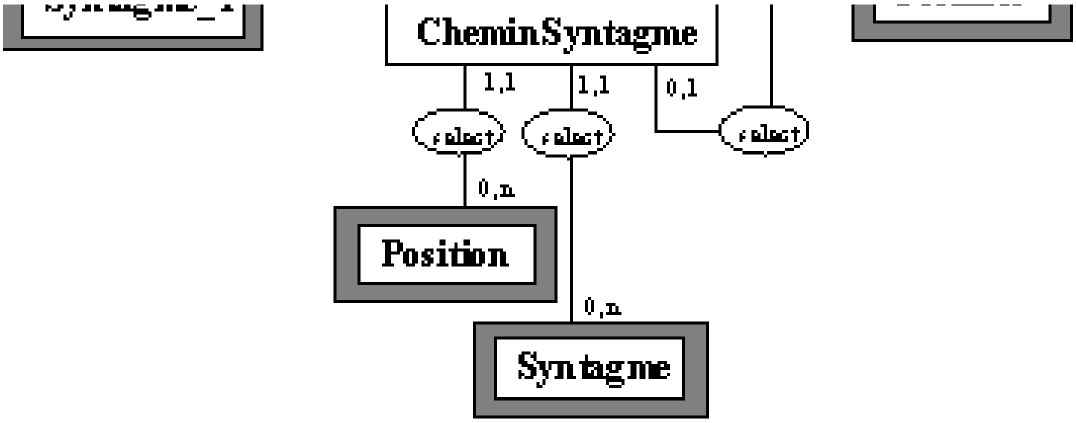


3 : Syntagme_T and Syntagme_NT_C

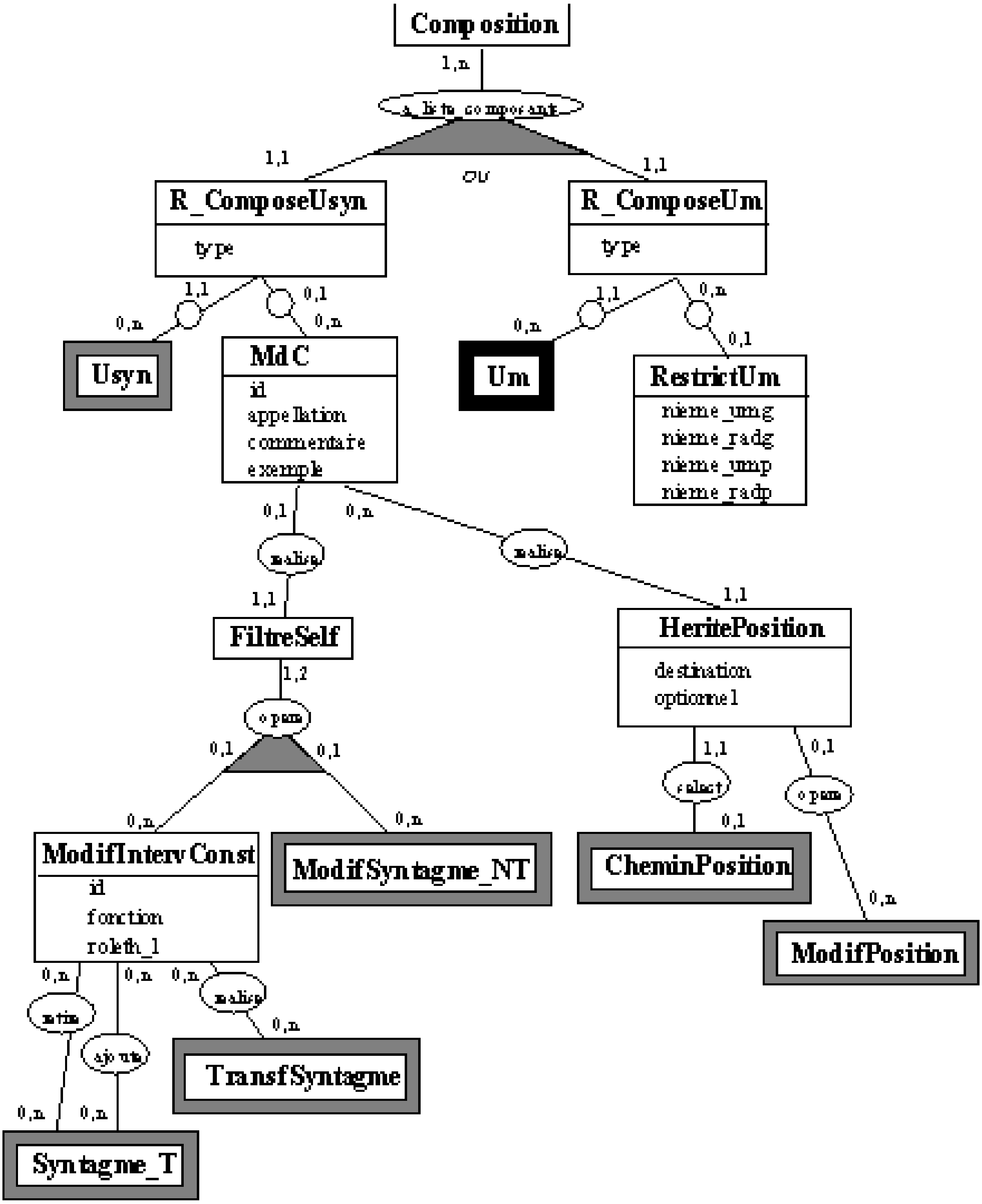


4 : Condition, CheminPosition, CheminSyntagme

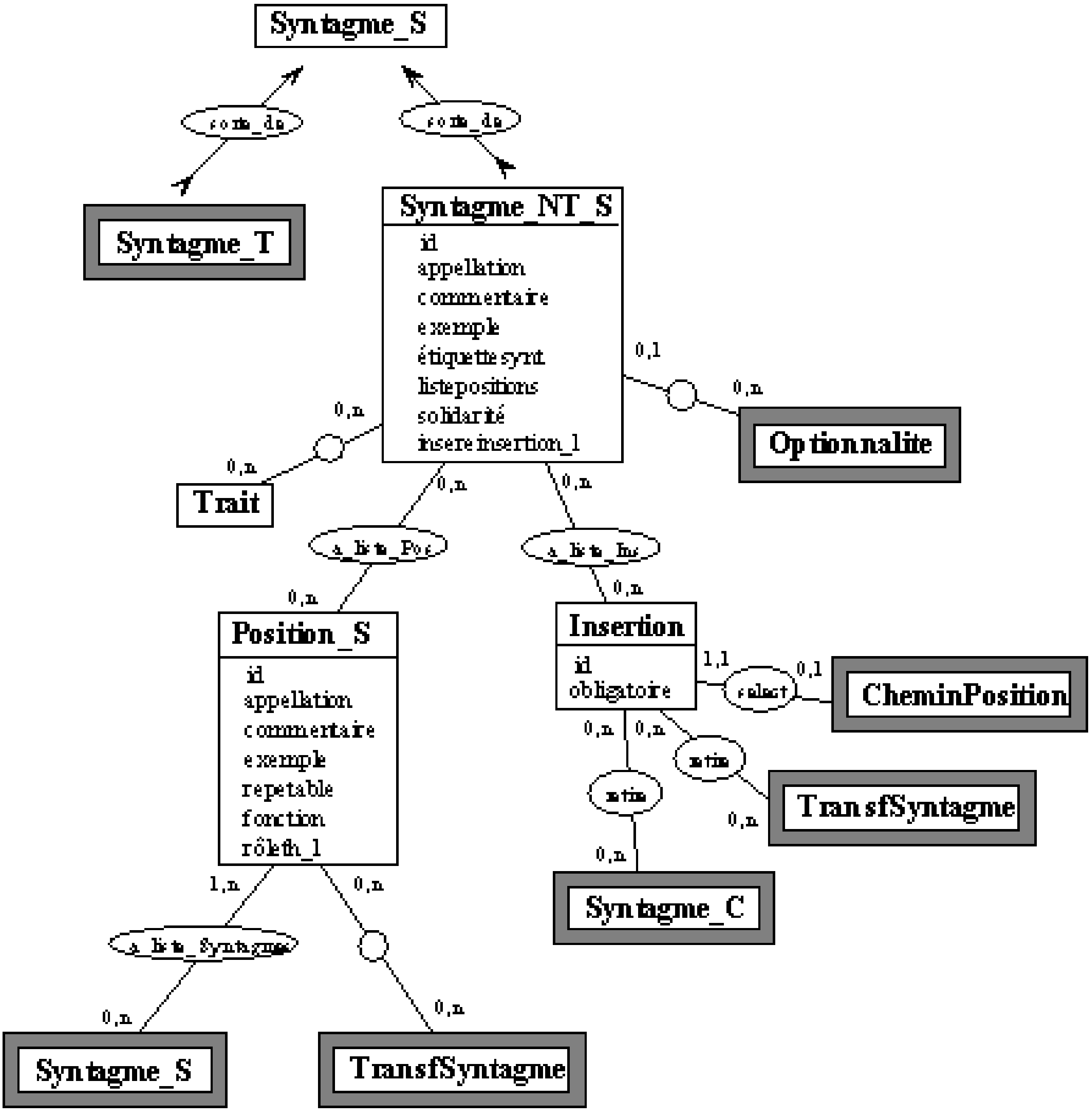




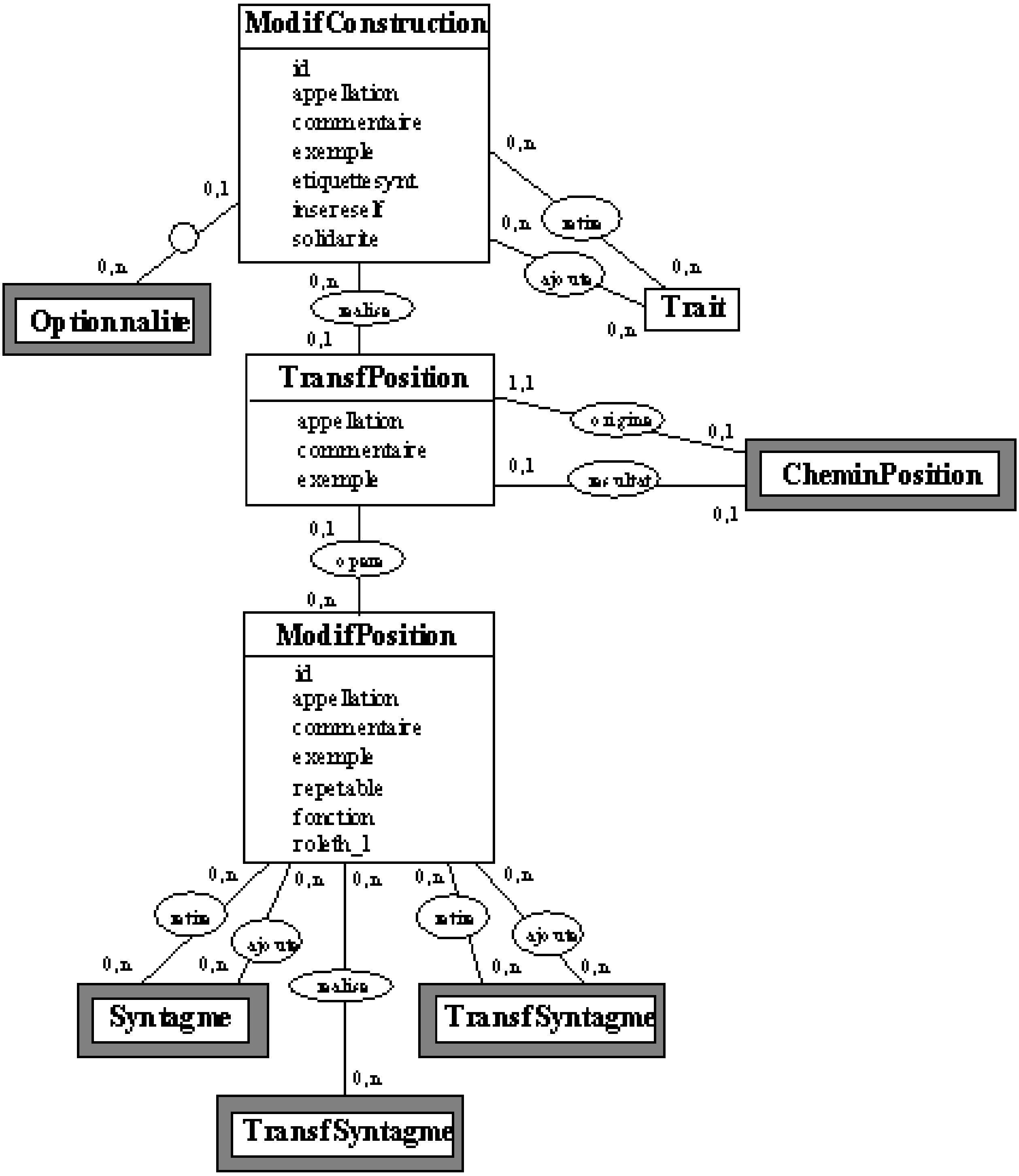
5 : Composition, MdC and ModifIntervConst



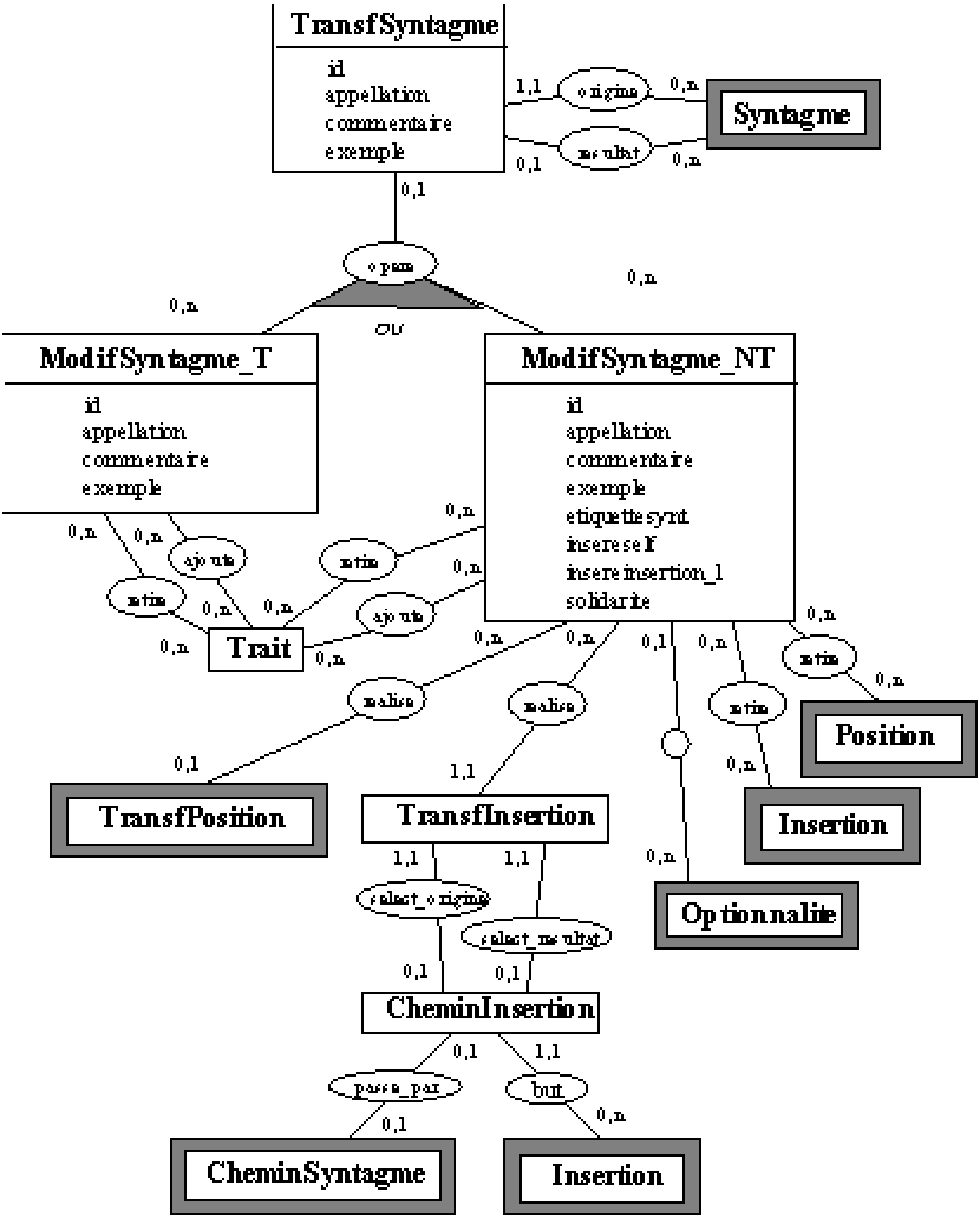
6 : Syntagme_NT_S, Position_S and Insertion



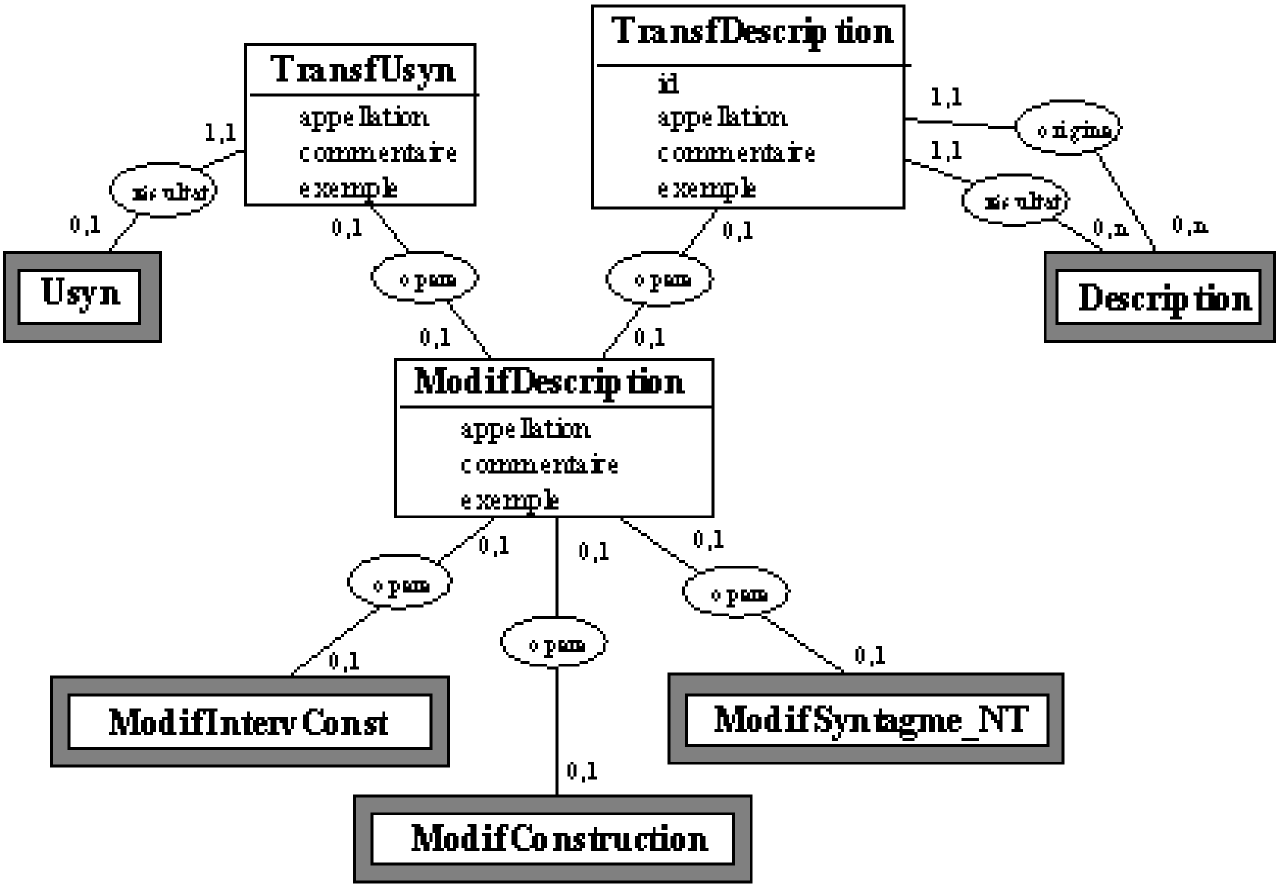
7 : ModifConstruction, TransfPosition and ModifPosition



8 : TransfSyntagme, ModifSyntagme_T and ModifSyntagme_NT



9 : TransfUsyn, TransfDescription and ModifDescription



# H - SGML DTD

## I : Introduction - Translation of the conceptual model

The conceptual model of GENELEX has been expressed to a great extent in terms of Entity-Attribute-Relation models (Merise).

Many constraints of integrity are expressed in this formalism: type of objects, type of relations, cardinality of relations, etc. However, since the model was not conceived to express rules - doing so gives rise to extreme complications - certain constraints had to be expressed in the accompanying document (restriction on the combination of values). It follows that the conceptual model of GENELEX is a combination of the Entity-Attribute-Relation (EAR) formalism and of natural language comments.

An SGML DTD (Document Type Definition) is a physical model of grammar that describes the marking of data.

When shifting from the Conceptual Model to the GENELEX DTD we have attempted to systematically translate the EAR models and to formally express most of the integrity constraints described in natural language.

Certain rules of translation from the EAR formalism into SGML have been implemented:

(i) EAR entities become SGML elements.

(ii) Attributes of EAR entities become Attributes of Elements.

> When the values of an Attribute are exclusive from one another and when they constitute a closed vocabulary, these values are represented in the form of listed

SGML attributes.

(iii) Non-attributed Relations pointing at a non-shared EAR entity are expressed by hierarchical links between the Elements of the DTD. Their cardinality is expressed by SGML occurrence indicators: **? + ***

(iv) Non-attributed Relations pointing at a shared EAR entity are expressed by reference links between Elements.

(v) Attributed Relations are expressed in the form of SGML attributed elements connected (hierarchy or reference relation) with the Elements translating the EAR Entities.

A file of constraints ("syntaxe.ctr") has been created to facilitate the reading of cross-references. These constraints appear as comments and are therefore not taken into account by an SGML parser; they show the typing of references in an intuition-based syntax.

# II: DTD GENELEX commented

## 1: DTD *genelex.dtd*

```
<!--Consortium GENELEX @(#) genelex.dtd 3.0 -->

<!-- **************FOR THE BENEFIT OF USERS ******************

Your comments concerning this DTD will be studied by the GENELEX

consortium. It will ensure the circulation of the new version that

may result from it.

*************************************************************** -->

<!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

<!--
```

```
A Genelex document is made of several parts:
- the morphological description
- the syntactic description
- ...
To select the chosen part, you just have to specify the appropriate
keyword (INCLUDE or IGNORE) in the following entity declarations:
-->
<!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">
```

```
<!-- As a general rule, in the whole file:

- appellation : allows to name the object in a comprehensible

and, if possible, univocal way

- exemple : allows to illustrate the use (quotation, example

taken from a dictionary or a linguist)

- commentaire : free field for the user -->

<!-- ********************************************************** -->

<!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 -->

<!--

<!-- **************FOR THE BENEFIT OF USERS ******************

Your comments concerning this DTD will be studied by the GENELEX

consortium. It will ensure the circulation of the new version that

may result from it.

**************************************************************-->

<!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*)>

<!-- ********************************************************* -->

<!-- ***** SYNTACTIC UNIT AND 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>

<!-- The field 'attestation' allows to specify the source of the use

encountered (name or title of the dictionary, author text,

or linguistics article -->
```

```
<!-- The attribute 'description' records the base description,

the list 'description_l' record the transformed descriptions,

the list 'transfdescription_l' records the transformations between the

descriptions associated with the Usyn; these transformations may

operate between the base description and the transformed descriptions

but also between transformed descriptions -->

<!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>

<!-- The field 'syntagme_nt_s' is only instantiated for compound Usyns,

it expresses their internal structure, if necessary reduced to

the syntagmatic label, by a 'Syntagme_NT_S' with or without rewriting.
```

```
Ex : mettre en marche SV
In the same way, the fields 'syntagme_nt_s_l' and 'transfsyntagme_l'
only concern compound Usyns and are used to record possible
transformations on the internal structure.
The field 'intervconst' gives the realizations of Self intervening
in the external construction:
- as occupant of the construction if this latter describes
a context of occurrence into which Self inserts,
- as predicate associated with a construction describing a
complementation pattern
The list comportappele_l allows to indicate the behavior
alternatives of Self as called by an element not described
in its construction-->
<!ELEMENT IntervConst - O EMPTY>
<!ATTLIST IntervConst
id ID #REQUIRED
fonction IDREF #IMPLIED
roleth_l IDREFS #IMPLIED
syntagme_t_l IDREFS #REQUIRED>
<!-- The element 'IntervConst' has terminal phrases.
It is possible to express variations of realizations of Self:
Ex : N + [Nombre:PLURIEL]
Ex : V
V + [Pronominal:SE]
V + [Pronominal:SE_EN]
For simple words, a difference between the functional category
```

```
and the grammatical category of the morphological unit is allowed.

Ex : description of the behavior as an adjective of the

Morphological Unit of the NOUN abricot

For syntactic compounds, the functional (external) category of the

compound that may differ from its label of internal structural phrase

is expressed

Ex : mettre en oeuvre (VERBE / SV)

Besides, it is possible to indicate on this element a function and

thematic roles: these are the values borne by Self when it inserts

into the construction. -->

<!ELEMENT ComportAppele - O EMPTY>

<!ATTLIST ComportAppele

id ID #REQUIRED

fonction IDREF #IMPLIED

roleth_l IDREFS #IMPLIED

syntagme_t IDREF #REQUIRED>

<!-- 'ComportAppele' records a behavior of Self as called by an element

not described in its construction.

A behavior gathers:

- a terminal syntagmatic label and a list of features giving

the functional category of the called element and the associated

restricting features:

this association is the same as a terminal phrase

Ex : PREP + [SsCatSynt:LIEU]

(if the Self bears an IntervConst, the syntagmatic label

of the phrase of the 'ComportAppele' must be the same as

the label of one of the phrases of that IntervConst)
```

```
- a function as a called
- a list of thematic roles -->
<!-- ********************************************************* -->
<!-- ***** CONSTRUCTION ***** -->
<!-- ********************************************************* -->
<!ENTITY % pConstSyntNT
"%pGlose
etiquettesynt (SANS_E
|%pEtiquetteSynt_NT
|%pEtiquetteSynt_cust) SANS_E
solidarite CDATA #IMPLIED
optionnalite IDREF #IMPLIED">
<!-- The attribute 'solidarite' indicates with dashes the pairs
of interdependent positions;
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>
<!-- Optionality is expressed:
```

```
- on the one hand, by a wording indicating in brackets for
each position if it can be deleted in a realization
of the construction; all the positions of the construction or of
the phrase are given in this field, the insertion point of Self
is not given here
Ex : P0 (P1) P2 (P3)
- on the other hand, by conditions expressing possible
interdependencies between positions-->
<!ELEMENT Construction - O EMPTY>
<!ATTLIST Construction
id ID #REQUIRED
%pConstSyntNT
squelettique (%pBooleen) NON
listepositions (%pTypeListPos) FERMEE
insereself NUMBER #IMPLIED
trait_l IDREFS #IMPLIED
position_c_l IDREFS #REQUIRED>
<!-- The attribute 'squelettique' indicates whether the element is a
construction skeleton - thus, to be connected with a
'ModifConstruction' to obtain a full Construction.
The attribute listepositions indicates whether the list
'position_c_l' is to be understood as being FERMEE (closed)
- all the positions are given - or OUVERTE (open).
The attribute 'insereself' indicates, when it is documented, the
insertion point of Self in the list of positions: before the rank
position corresponding to its value.
The list 'position_c_l' refers to 'Position_C'.
```

```
It is an ordered list, the order of elements in the list corresponds
to the canonical order (initial value = 0),
and the elements may subsequently be referenced by their rank in that
list -->
<!-- ********************************************************* -->
<!-- ***** POSITION AND INSERTION ***** -->
<!-- ********************************************************* -->
<!ENTITY % pPosition
"%pGlose
repetable (SANS_B|%pBooleen) SANS_B
fonction IDREF #IMPLIED
roleth_l IDREFS #IMPLIED">
<!-- The attribute 'repetable' indicates if a position can be
realized several times-->
<!ELEMENT (Position_C|Position_S) - O EMPTY>
<!ATTLIST Position_C
id ID #REQUIRED
%pPosition
syntagme_c_l IDREFS #IMPLIED
transfsyntagme_l IDREFS #IMPLIED>
<!ATTLIST Position_S
id ID #REQUIRED
%pPosition
syntagme_s_l IDREFS #REQUIRED
transfsyntagme_l IDREFS #IMPLIED>
<!-- The attribute 'syntagme_c_l' refers the possible occupants
```

```
of 'Position_C':
- terminal phrase (S'yntagme_T')
- non-terminal phrase, the rewriting of which is or not
described ('Syntagme_NT_C') -->
<!-- The attribute 'syntagme_s_l' refers the possible occupants
of 'Position_S':
- terminal phrase (S'yntagme_T')
- non terminal phrase, the rewriting of which is or not
described ('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>
<!-- 'Insertion' in a structural phrase is only used to represent the
case of an insertion that refers to a position of a position
described in the external syntactic construction:
the element 'CheminPosition' gives access to this position.
The attributes of the Insertion indicate the possible modifications
that can be applied to the referenced position: removal of phrases and
transformations between phrases. The repeatability is that of the
referenced external position.
Ex: the compound "mettre en oeuvre" has in its external
construction a position of direct object containing for instance
a noun phrase and a personal pronoun:
mettre en oeuvre un processus, le mettre en oeuvre
```

```
(to implement a process, to implement it)

insertion is only possible for the noun phrase:

mettre un processus en oeuvre

The attribute 'obligatoire' indicates, whether, during a realization

of the referenced external position, the insertion phenomenon is

compulsory or optional -->

<!-- ********************************************************* -->

<!-- ***** CONDITION ***** -->

<!-- ********************************************************* -->

<!ELEMENT Condition - O (Si+ , Alors+)>

<!ATTLIST Condition

appellation CDATA #IMPLIED>

<!-- Conditions allow to express mutual constraints between realizations

of positions.

Ex : si P0 == P[SsCatSynt:COMPLETIVE]

alors P1 != P[SsCatSynt:COMPLETIVE]

The lists of Predicates (Si and Alors) allow to express

conjunctions on these predicates.

Disjunctions are expressed by the list of Conditions

borne by the Description (list "et"). -->

<!ELEMENT (Si|Alors) - O (CheminPosition|CheminSyntagme

|SelectIntervConst)>

<!ATTLIST (Si|Alors)

portee (%pPortee) EXTERNE

negation (%pBooleen) NON>

<!-- A predicate selects:
```

```
- a phrase or a position of the external construction
(field 'portee EXTERNE'),
- a phrase or a position of the structural phrase of Self
(field 'portee INTERNE'),
- a realization of Self as an intervener in the construction
(field 'portee INTERVENANT').
The selection is made
- for a phrase, using 'CheminSyntagme',
- for a position using 'CheminPosition',
- for a realization as an intervener using 'SelectIntervConst'.
If necessary, a predicate may bear a negation that expresses
the inhibition of the phrase or of the position pointed at. -->
<!ELEMENT CheminSyntagme - O (CheminSyntagme?)>
<!ATTLIST CheminSyntagme
nieme_position NUMBER 0
syntagme IDREF #REQUIRED>
<!-- This element allows to select a particular phrase.
Recursivity is used to go down in a possible rewriting. The result
of the element is always a phrase.
The positions are referred by 'nieme_position' that indicates
their rank in the list in which they occur;
the value 0 refers the first element of the list. -->
<!ELEMENT CheminPosition - O (CheminSyntagme?,PositionBut)>
<!-- For a given construction or a phrase, this element allows
to select one of its positions - it may occur in the
rewriting of the phrase -.
The element 'PositionBut' indicates the selected position;
```

```
if this latter occurs in a phrase rewriting, the element

'CheminSyntagme' is used to reach that phrase -->

<!ELEMENT PositionBut - O EMPTY>

<!ATTLIST PositionBut

nieme_position NUMBER 0>

<!ELEMENT SelectIntervConst - O EMPTY>

<!ATTLIST SelectIntervConst

syntagme_t IDREF #REQUIRED>

<!-- *********************************************************** -->

<!-- ***** TERMINAL AND NON-TERMINAL PHRASES ****** -->

<!-- *********************************************************** -->

<!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>
<!-- The list 'position_c_l' refers to 'Position_C'.
It is an ordered list, the order of elements in the list
corresponds to the canonical order and the elements may
subsequently be referred by their rank in this list. -->
<!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>
<!-- The list 'position_s_l' refers to 'Position_S'; the list 'insertion_l'
refers to Insertions. These are ordered lists, their elements
may subsequently be referenced by their rank of occurrence in their
list.
The list of integers 'insereinsertion_l' indicates the points where
the Insertions present in the list 'insertion_l' insert among the
Positions-->
<!-- 'Syntagme_T' is a terminal position occupant (label reduced
to the grammatical category or to "e" for tracing)
'Syntagme_NT_C/S' is a non-temrinal position occupant,
one will use its list of positions (#IMPLIED) if one wants to
specify the way it should be rewritten.
Besides, 'Syntagme_NT_S' is used to describe the internal structure
```

```
of a compound unit (field 'syntagme_nt_s' of the Self).

The list 'trait_l' refers to restricting features and thus it allows

to specify on a phrase a set of constraints:

lexical, morphological, morpho-syntactical, syntactico-semantic,

and even semantic.

The field 'appellation' allows to record the usual name of the phrase.

Ex : Syntagme_NT_C

etiquettesynt = "P"

trait_l -> [Mode:INFINITIF]

appellation = "phrase infinitive"

Category "e" allows adherents to the generative grammar to record

traces and to consider them as 'phantom' phrases

with which one can associate the necessary restrictions.

Ex : Syntagme_T

etiquettesynt = "e"

trait_l -> [Personne:3][Nombre:SINGULIER]

appellation = "elt vide" -->

<!-- ********************************************************* -->

<!-- ***** COMPOSITION ***** -->

<!-- ********************************************************* -->

<!ELEMENT Composition - O (R_ComposeUm|R_ComposeUsyn)+>

<!-- The elements 'Composition' borne by a syntactic unit

record the alternative lists of lexicalizations:

Ex : (avoir admiration pour)

(eprouver admiration pour)

(eprouver admiration envers)
```

```
(porter admiration a) -->
<!-- The list of 'R_ComposeUm' and 'R_ComposeUsyn' gives the list of
components for a given composition alternative. -->
<!-- The components will be referred by RefLex features bearing
two index values:
- the composition index in the list of compositions
- the component index in the list of components
Ex. : [RefLex:1,2]:
This mechanism will be used:
- either in the internal structural phrase of Self,
- or in the MdC calculation mode. -->
<!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>
<!-- If one wants to record the formation of the compound according
to the component Usyn:
- lexicalization of one of its positions by other components
(only for a behavior of caller)
and/or
```

```
- inheritance of its positions with addition of restrictions,
if necessary,
an MdC that describes all these phenomena is associated with it.
Note: the lexicalization of a position of a calling Usyn by another
component is done via the introduction of a phrase bearing
a feature [RefLex:nieme_cposition,nieme_cposant] in the
Usyn composition mode. Besides, this phrase may bear other
restricting features; in that case, one has to make sure
that these features are compatible with the MdC and the Self
of the Usyn behaving as a called component.
Ex : the MdC of the calling unit indicates:
SN[RefLex:3,1][SsCatSynt:DET_VIDE]
The called unit (3,1) cannot impose in its MdC
a lexicalization of its position Determiner. -->
<!ELEMENT MdC - O (HeritePosition* & FiltreSelf?)>
<!ATTLIST MdC
id ID #REQUIRED
%pGlose>
<!-- The MdC describes the composition mode of a compound Usyn on one of
its component Usyns. It allows to specify:
- the constraints expressed by the compound Usyn on the component
Usyns,
- the mutual organization of the component Usyns, indicating
which component occupies which position in another component.
The MdC applies to the component Usyn and allows to
- inherit positions from its construction
- filter its Self
A position that is not referred by the MdC is inhibited.
When no filter is specified on the Self, this latter is
default inherited-->
<!ELEMENT HeritePosition - O (CheminPosition)>
<!ATTLIST HeritePosition
destination (%pDestination) EXTERIEUR
optionnel (HERITAGE|%pBooleen) HERITAGE
modifposition IDREF #IMPLIED>
<!-- The attribute 'destination' indicates whether the inherited position
is to be found outside or inside the compound.
The attribute 'optionnel' allows to note eventual modifications
on the option of the inherited position. -->
<!ELEMENT FiltreSelf - O EMPTY>
<!ATTLIST FiltreSelf
modifintervconst IDREF #IMPLIED
modifsyntagme_nt IDREF #IMPLIED>
<!-- 'FiltreSelf' must at least realize one of the two operations:
- Modification of IntervConst
- Modification of the internal structural phrase.-->
<!-- ********************************************************** -->
<!-- ***** TRANSFORMATIONS ***** -->
<!-- ********************************************************** -->
<!-- There are three types of transformations:
- 'TransfUsyn': transformations operating between two Syntactic Units
- stemming from the same morphological unit
Ex: neutrality
```

```
- stemming from different morphological units

Ex: syntactic derivation

- 'TransfDescription': transformations operating between two

Descriptions

(i.e. two pairs Self/Construction).

Ex: passivation

- 'TransfSyntagme': transformations operating between Position

occupants, i.e. terminal or non-terminal Phrases.

Ex: pronominalization -->

<!ELEMENT TransfUsyn - O (ModifDescription?)>

<!ATTLIST TransfUsyn

%pGlose

usyn_resultat IDREF #REQUIRED>

<!-- 'TransfUsyn' are pointed at by the source syntactic units and indicate

the result syntactic units

They operate between the base Description of the source Usyn

and the base Description of the result Usyn-->

<!ELEMENT TransfDescription - O (ModifDescription?)>

<!ATTLIST TransfDescription

id ID #REQUIRED

%pGlose

description_origine IDREF #REQUIRED

description_resultat IDREF #REQUIRED>

<!-- 'TransfDescriptions' are borne by the Usyn and operate between the

descriptions of that Usyn.

The origin of a 'TransfDescription' may be the base Description
```

```
but also a transformed Description of the Usyn. -->
<!ELEMENT ModifDescription - O EMPTY>
<!ATTLIST ModifDescription
%pGlose
modifconstruction IDREF #IMPLIED
modifintervconst IDREF #IMPLIED
modifsyntagme_nt IDREF #IMPLIED>
<!-- 'ModifDescriptions' may give account of three phenomena:
- transformation on the construction
- transformation on the realizations of Self associated
with the external construction as occupant or predicate
(field 'intervconst' of Self)
Ex : V -> V[Passif:PLUS]
- transformation on the internal structural phrase describing
a compound
Ex : pour les beaux yeux de SN
-> pour ses beaux yeux
In the first case, it is possible to choose either a calculation
or description mode:
- in the calculation mode, the Construction pointed at by the
result Description is said 'squelettique': the result
construction itself has to be constructed from that skeleton
by applying to it the modifications indicated on
'ModifConstruction'
- in the description mode, the Construction pointed at by the
result Description is fully described: then the
'ModifConstruction' indicates correspondences,
with varying fineness levels, between the source Construction

and the result Construction.

In the other cases, the results - 'IntervConst' and

'Syntagme_NT_S' - are fully described: we are in a description

mode-->

<!ELEMENT ModifConstruction - O (TransfPosition*)>

<!ATTLIST ModifConstruction

id ID #REQUIRED

%pGlose

etiquettesynt (HERITAGE

|%pEtiquetteSynt_NT

|%pEtiquetteSynt_cust) HERITAGE

insereself NUMBER #IMPLIED

solidarite CDATA #IMPLIED

optionnalite IDREF #IMPLIED

retire_trait_l IDREFS #IMPLIED

ajoute_trait_l IDREFS #IMPLIED>

<!-- In the description mode, this element allows to record, partially

if necessary - information on the shift from the source

construction to the result construction; it may minimally concern

correspondences of positions by TransfPosition*.

In the calculation mode, this element allows to construct the result

construction from the source construction and the skeleton:

the skeleton is enriched with elements or attributes coming from the

source construction or specified on ModifConstruction.

If the attributes 'etiquettesynt', 'insereself', 'solidarite',
'optionnalite' and those grouped together by 'pGlose'
- are documented: they fill (even delete) the corresponding
attributes of the skeleton
- are not documented: the corresponding attributes that are not
documented on the skeleton construction are inherited
from the source construction
The features of the source Construction are inherited with the
withdrawals and additions recorded here.
The list of TransfPositions expresses the formation
of the positions of the result construction from the positions
of the skeleton and the positions of the source construction. -->
<!ELEMENT TransfPosition - O (CheminPosition , CheminPosition?)>
<!ATTLIST TransfPosition
%pGlose
modifposition IDREF #IMPLIED>
<!-- Selection of the source Position
The first element of 'CheminPosition' points at a position
of the construction which caused the transformation.
Selection of the result Position
The result position may be selected by the second element
of 'CheminPosition'
The attribute 'ModifPosition' indicates the modifications to
be applied to shift from the source position to the result
position -->
<!ELEMENT ModifPosition - O EMPTY>
<!ATTLIST ModifPosition
id ID #REQUIRED
```

```
%pGlose

repetable (HERITAGE|%pBooleen) HERITAGE

fonction IDREF #IMPLIED

roleth_l IDREFS #IMPLIED

retire_syntagme_l IDREFS #IMPLIED

retire_transfsyntagme_l IDREFS #IMPLIED

ajoute_syntagme_l IDREFS #IMPLIED

ajoute_transfsyntagme_l IDREFS #IMPLIED

transfsyntagme_l IDREFS #IMPLIED>

<!-- 'ModifPosition' functions like 'ModifConstruction'; one will have

the same alternative between a calculation mode - the result

position is to be built from the position pointed at on the skeleton

and the information recorded here - and a description mode -

the result position is already fully documented.-->

<!ELEMENT TransfSyntagme - O EMPTY>

<!ATTLIST TransfSyntagme

id ID #REQUIRED

%pGlose

syntagme_origine IDREF #REQUIRED

syntagme_resultat IDREF #IMPLIED

modifsyntagme IDREF #IMPLIED>

<!-- 'TransfSyntagme' record the transformation relations existing between:

- phrases that occupy the same position - terminal or

non-terminal phrases-, the phrases at the origin and the result

of the transformation are both already documented,

- phrases connnected during a transformation of the highest level;
```

```
in that case, the result phrase need not necessarily be

documented if a Construction transformation operating in

calculation mode occurs-->

<!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

insereself NUMBER #IMPLIED

insereinsertion_l NUMBERS #IMPLIED

retire_position_l NUMBERS #IMPLIED

retire_insertion_l NUMBERS #IMPLIED

solidarite CDATA #IMPLIED

optionnalite IDREF #IMPLIED>

<!-- Modifying a Phrase consists in:

- deleting or adding features,

- and for a non-terminal phrase:
```

```
- changing the label,

- changing the option, the interdependency and the

recorded insertion points, if any.

- modifying the positions and insertions of the rewriting,

if any-->

<!ELEMENT TransfInsertion - O (CheminInsertion, CheminInsertion)>

<!-- The element 'TransfInsertion' connects an insertion of the source

structural phrase and the corresponding insertion of the result

structural phrase. -->

<!ELEMENT CheminInsertion - O (CheminSyntagme?,InsertionBut)>

<!ELEMENT InsertionBut - O EMPTY>

<!ATTLIST InsertionBut

nieme_insertion NUMBER 0>

<!-- The attribute 'nieme_insertion' selects an insertion using its rank

in the list 'insertion_l' in which it appears -->

<!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>

<!-- The element 'ModifIntervConst' allows to record the modifications

applied in transformation or composition, by the IntervConst

of Self. -->
```

```
<!-- ************************************************************ -->
<!-- ***** RESTRICTING FEATURES ***** -->
<!-- ************************************************************ -->
<!-- Restricting features allow to specify the phrases with which
they are combined
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] -->
<!--******************** LEXICAL Features **********************-->
<!--************************************************************-->
<!ENTITY % pTrait_Lexical
"id ID #REQUIRED
valeur CDATA #IMPLIED
um IDREF #IMPLIED">
<!-- These features allow to express a lexical restriction on a Phrase,
- either by entering in the field "valeur" a character string
representing the written form - this means does not disambiguate
homographs -,
- or by referring a morphological unit (by its identifier).
A gap is allowed (based on the distance between the morpho-
syntactic category and the functional category) between the
category of the phrase that bears the lexical feature and the
category of the Um referred by this lexical feature
```

```
Ex : NOM[Lex:courageux[um:UM04]]

with in Morphology:

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">
```

```
<!-- These features refer by their coordinates ('nieme_cposition'

indicates the rank in the Composition list borne by the Usyn

and nieme_cpsosant the rank in the mixed list of 'R_ComposeUm/Usyn'

borne by the Composition) the Um or Usyn entering in the composition

of the Unit. The value 0 on 'nieme_cposition' refers

all the components of 'nieme_cpsant' rank independently from

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>

<!--****************** MORPHOLOGICAL Features*****************-->
```

```
<!--************************************************************-->

<!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>

<!-- The feature 'temps' allows to express restrictions on tense

relating to certain forms.

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_NombrePosseur - O EMPTY>
<!ATTLIST Trait_NombrePosseur
id ID #REQUIRED
valeur (%pNombrePosseur) SINGULIER_POSSEUR>
<!--***************** MORPHO-SYNTACTIC Features ******************-->
<!--**************************************************************-->
<!ELEMENT Trait_SsCatMorph - O EMPTY>
<!ATTLIST Trait_SsCatMorph
id ID #REQUIRED
valeur (%pSsCatGram) COMMUN>
<!-- The values for the morphological sub-category features
are predefined -->
<!ELEMENT Trait_SsCatSynt - O EMPTY>
<!ATTLIST Trait_SsCatSynt
id ID #REQUIRED
valeur (%pSsCatSynt
|%pSsCatSynt_cust) COORDONNE>
<!-- The syntactic sub-category features may concern terminal or non-
terminal categories and are defined according to the needs,
in addition to certain values already provided for by the GENELEX
model.
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>
<!-- This feature allows to associate a given verb (the described
entry or a verb in the context of the entry) with the auxiliary verb
that corresponds to a use and has to be associated with it.
Ex : se lever (etre leve) // lever (avoir leve)
V[Lex:lever] V[Lex:lever]
[Aux:ETRE] [Aux:AVOIR]
[Pronominal:SE]
The presence of 'Trait_Aux' together with a 'Trait_Temps' having
the value 'COMPOSE' indicates that the auxiliary is necessarily
present in the use under description.
Ex : etre arrive
V[Lex:arriver]
[Aux:ETRE]
[Temps:COMPOSE]
The attributes 'temps' and 'mode' specify, whenever necessary,
the tense and mood of the auxiliary verb itself.
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>
<!-- This feature allows to associate a given verb (the described entry
or a verb in the context of the entry) with the NON REFERENTIAL
preverbal particle that corresponds to a use and that has to be
combined to it (See "true" pronominal verbs).
Ex : s'en aller
V[Lex:aller]
[Pronominal:SE_EN] -->
<!ELEMENT Trait_Neg - O EMPTY>
<!ATTLIST Trait_Neg
id ID #REQUIRED
valeur (%pNeg) LIBRE>
<!-- The presence of a 'Trait_Neg' indicates that the use under description
is in the negative form; besides it is possible to specify in the
field 'valeur' a restriction on the lexicalization of the 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>

<!--******************* SYNTACTICO-SEMANTIC Features ***********-->

<!--************************************************************-->

<!ELEMENT Trait_Coref - O EMPTY>

<!ATTLIST Trait_Coref

id ID #REQUIRED

valeur (%pIJKL) I>

<!-- Coreference must always be resolvable: if a feature with the value I

exists, there is at least another feature I or NON_I that matches it.

Coref features do not force the co-realization of the phrases

that bear them; if one wants to force this co-realization,

Conditions have to be used-->

<!--***************** SEMANTIC Features ************************-->

<!--************************************************************-->

<!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>

<!-- This type of feature allows for instance to express "denotative

conditions",
Ex: anime (animated), humain (human) -->

<!ELEMENT Trait_Libre - O EMPTY>

<!ATTLIST Trait_Libre

id ID #REQUIRED

nom CDATA #REQUIRED

valeur CDATA #REQUIRED>

<!-- This type of feature can be used to specify semantic classes or

families

Ex : nom : classe

valeur : vetement -->

<!-- The features Bin and Libre can also be used for other features

not predefined in this DTD -->

<!--******************** THEMATIC ROLE ********************-->

<!--************************************************************-->

<!ELEMENT RoleTh - O EMPTY>

<!ATTLIST RoleTh

id ID #REQUIRED

valeur (%pRoleTh_cust) AGENT>

<!--*********************** FUNCTION **************************-->

<!--************************************************************-->

<!ELEMENT Fonction - O EMPTY>

<!ATTLIST Fonction

id ID #REQUIRED

valeur (%pFonction_cust) TETE>

<!-- The entities suffixed with _cust are defined in a file

"custom.ent" specific to the user; he may therefore add

the attribute values he wants to have at hand -->
```

## 3 : Entities *morpho.ent*

```
<!--Consortium GENELEX @(#) morpho.ent 3.0 -->

<!-- **************FOR THE BENEFIT OF USERS ******************

Your comments concerning this DTD will be studied by the GENELEX

consortium. It will ensure the circulation of the new version that

may result from it.

****************************************************************-->

<!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 : Entities *syntaxe.ent*

```
<!--Consortium GENELEX @(#) syntaxe.ent 4.0 -->

<!-- **************FOR THE BENEFIT OF USERS ******************

Your comments concerning this DTD will be studied by the GENELEX

consortium. It will ensure the circulation of the new version that

may result from it.

****************************************************************-->

<!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 : Entities *custom.ent*

```
<!--Consortium GENELEX @(#) custom.ent 3.0 -->

<!-- **************FOR THE BENEFIT OF USERS ******************

Your comments concerning this DTD will be studied by the GENELEX
```

```
consortium. It will ensure the circulation of the new version that
may result from it.
****************************************************************-->
<!-- The user may access this file to define additional values of
attribute in replacing the text "_VALEURS_A_DEFINIR_" from the
entity by the list of values he wants to have at hand:
"VALEUR1|VALEUR2|...|VALEURn" -->
<!-- For the moment, this feature is only possible for the syntagmatic
label, the function, the thematic role and the syntactic
sub-categories.
For the syntagmatic label and the sub-category, a minimal list of
values is defined in the file "syntaxe.ent".
This possibility might be extended to other features. -->
<!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 : Constraints *syntaxe.ctr*

```
<!--Consortium GENELEX @(#) syntaxe.ctr 4.0 -->
```

```
<!--CONTRAINTE Usyn
combve TYPE CombVE
(description
|description_l) TYPE Description
transfdescription_l TYPE TransfDescription -->
<!--CONTRAINTE Description
self TYPE Self
construction TYPE Construction -->
<!--CONTRAINTE Self
(syntagme_nt_s
|syntagme_nt_s_l) TYPE Syntagme_NT_S
transfsyntagme_l TYPE TransfSyntagme
intervconst TYPE IntervConst
comportappele_l TYPE ComportAppele -->
<!--CONTRAINTE IntervConst
fonction TYPE Fonction
roleth_l TYPE RoleTh
syntagme_t_l 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 : Examples of tagged data

The example of SGML tagging shown here gives a more detailed representation of the two ways for coding the compound Usyn *"abattre carte ma"tresse"*, as given in the example (13), part D of this document (Annexes to Compound Syntactic Units).

```
<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"></>
</>
</>
<!-- *************************************** -->
<!-- *************************************** -->
<!-- Unites Syntactiques -->
<!-- *************************************** -->
<!-- *************************************** -->
<!-- *************************************** -->
<!-- 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 composŽe 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>
```