# EAGLES

# Recommendations
# on
# Subcategorisation

EAGLES DOCUMENT EAG—CLWG—SYNLEX

Version of 31st August, 1996

# Contents

# Chapter 1

# Contributors

This document, as with all EAGLES documents, is the result of cooperative work involving a number of individuals, contributing in different ways.

## 1.1 Authors


| | |
|---|---|
| R. Barnett | (3.3) |
| Department of Linguistics and | |
| Modern English Language | |
| Lancaster University | |
| Lancaster | Tel: +44.1524.65201 |
| United Kingdom | Fax: +44.1524.843085 |
| LA1 4YT | E-mail: R.Barnett@lancs.ac.uk |
| | |
| N. Calzolari | (2) |
| ILC–CNR | |
| Via della Faggiola 32 | Tel: +39.50.560.481 |
| I–56100 Pisa | Fax: +39.50.589.055 |
| Italy | E-mail: glottolo@ilc.pi.cnr.it |
| | |
| S. Flores | (4.1, 4.4, 5.1) |
| GSI-ERLI | |
| 1, Place des Marseillais | Tel: +33.1.48.93.81.21 |
| F–94227 Charenton-Le-Pont Cedex | Fax: +33.1.43.75.79.79 |
| France | E-mail: sylvie.flores@erli.gsi.fr |
| | |
| P. Hellwig | (3.1, A.3) |
| Universität Heidelberg | |
| Karlstrasse 2 | Tel: +49.6221.543245 |
| D–6900 Heidelberg | Fax: +49.6221.543242 |
| Germany | E-mail: C87@vm.urz.uni-heidelberg.de |
| | |
| P. Kahrel | (3.3) |
| Department of Linguistics and | |
| Modern English Language | |
| Lancaster University | |
| Lancaster | Tel: +44.1524.65201 |
| United Kingdom | Fax: +44.1524.843085 |
| LA1 4YT | E-mail: P.Kahrel@lancs.ac.uk |

| | |
|---|---|
| G. Leech | (3.3) |
| Department of Linguistics and | |
| Modern English Language | |
| Lancaster University | |
| Lancaster | Tel: +44.1524.593036 |
| United Kingdom | Fax: +44.1524.843085 |
| LA1 4YT | E-mail: G.N.Leech@lancaster.ac.uk |
| | |
| M. Melera | (3.1, A.2) |
| GILCUB | |
| Avda. Vallvidrera 25 | Tel: |
| 08017 Barcelona | Fax: +34.3.2054656 |
| Spain | E-mail: maite@gilcub.es |
| | |
| S. Montemagni | (3.2, 4.5, 4.7, A.4) |
| ILC–CNR | |
| Via della Faggiola 32 | Tel: +39.50.560.481 |
| I–56100 Pisa | Fax: +39.50.589.055 |
| Italy | E-mail: simo@ilc.pi.cnr.it |
| J. Odijk | (4.6) |
| Institute for Perception Research (IPO) | |
| PO Box 513 | Tel: +31.40.2773843 |
| 5600 MB Eindhoven | Fax: +31.40.2773876 |
| The Netherlands | E-mail: odijkje@natlab.research.philips.com |
| | |
| V. Pirrelli | (3.2, 4.5, 4.7, A.4) |
| ILC–CNR | |
| Via della Faggiola 32 | Tel: +39.50.560.481 |
| I–56100 Pisa | Fax: +39.50.589.055 |
| Italy | E-mail: vito@ilc.pi.cnr.it |
| | |
| A. Sanfilippo | (2, 3.1, 4.3, 4.8, A.1) |
| Sharp Laboratories of Europe Ltd | |
| Edmund Halley Road | |
| Oxford Science Park | |
| Oxford | Tel: +44.1865.747711 |
| United Kingdom | Fax: +44.1865.747717 |
| OX4 4GA | E-mail: antonio@sharp.co.uk |
| | |
| S. Teufel | (3.3) |
| Centre for Cognitive Science | |
| 2 Buccleuch Place | |
| Edinburgh | Tel: +44.131.650.4667 |
| United Kingdom | Fax: +44.131.650.6626 |
| EH8 9LW | E-mail: simone@cogsci.ed.ac.uk |

| | |
|---|---|
| M. Villegas | (3.1, A.2) |
| GILCUB | |
| Avda. Vallvidrera 25 | Tel: |
| 08017 Barcelona | Fax: +34.3.2054656 |
| Spain | E-mail: marta@gilcub.es |

| | |
|---|---|
| L. Zaysser | (4.2) |
| GSI-ERLI | |
| 1, Place des Marseillais | Tel: +33.1.48.93.81.21 |
| F–94227 Charenton-Le-Pont Cedex | Fax: +33.1.43.75.79.79 |
| France | E-mail: laurence.zaysser@erli.gsi.fr |


## 1.2 Editors and assistants


| | |
|---|---|
| N. Calzolari | |
| Co-Chief Editor, EAGLES | |
| ILC–CNR | |
| Via della Faggiola 32 | Tel: +39.50.56.04.81 |
| Pisa I-56126 | Fax: +39.50.58.90.55 |
| Italy | E-mail: eagles@ilc.pi.cnr.it |

| | |
|---|---|
| J. McNaught | |
| Co-Chief Editor, EAGLES | |
| Centre for Computational Linguistics | |
| UMIST | |
| Manchester | Tel: +44.161.200.3098 |
| United Kingdom | Fax: +44.200.3099 |
| M60 1QD | E-mail: jock@ccl.umist.ac.uk |

| | |
|---|---|
| T. Paskiewicz | |
| EAGLES Editorial Assistant | |
| Centre for Computational Linguistics | |
| UMIST | |
| Manchester | Tel: +44.161.200.3097 |
| United Kingdom | Fax: +44.200.3099 |
| M60 1QD | E-mail: teresap@ccl.umist.ac.uk |

| | |
|---|---|
| T. Ayazi | |
| EAGLES Secretariat | |
| ILC–CNR | |
| Via della Faggiola 32 | Tel: +39.50.56.04.81 |
| Pisa I-56126 | Fax: +39.50.58.90.55 |
| Italy | E-mail: eagles@ilc.pi.cnr.it |

| | |
|---|---|
| A. Enea | |
| EAGLES Webmaster | |
| ILC–CNR | |
| Via della Faggiola 32 | Tel: +39.50.56.04.81 |
| Pisa I-56126 | Fax: +39.50.58.90.55 |
| Italy | E-mail: enea@ilc.pi.cnr.it |

# Chapter 2

# Introduction

It is not necessary here to provide a general motivation justifying the need for standards in the field of Language Engineering. We can simply state that standards are needed in the area of Computational Lexicons in order to:

- Pave the way towards the construction of commonly usable large-scale lexicons;
- Enable the merging of available, existing lexical material;
- Enable the linking of lexicons for different languages which follow the same recommendations; and
- Create the conditions for developing lexical tools based on common models.

Following the work done within the EAGLES Lexicon Working Group in the area of Morphosyntax for the standardisation of the lexical information at the morphosyntactic layer (EAGLES, 1996), the obvious next step was to move on to the syntactic layer and to deal with subcategorisation information. *Subcategorisation* — restricted to verbs for the moment — is broadly intended as referring to typical collocations sanctioned by strong syntactic/semantic selection (head/complement relation), thus leaving out other collocation types such as head/modifier, head/specifier, etc.
We state here a number of general issues which can be of help to better situate the work done by the Lexicon/Syntax group. The first issue concerns the characterisation of the EAGLES Lexicon work with respect to the way in which one can try to define standards. One can hypothesise different approaches to the task of reaching consensus, by providing standards at the levels of:

- Linguistic theories
- Linguistic frameworks
- Linguistic substance
- Choice/range of primitives
- Terminology
- Notation and formalism.

From this viewpoint, the most important concern for EAGLES is *linguistic substance*. Consequently, the group is building on the results of the ET–7 feasibility study (Heid & McNaught, 1991) which recommended the following methodology: to break up the complex descriptive devices into "minimal observable facts" in order to arrive at the most fine-grained, common set of features underlying different theoretical frameworks or systems. EAGLES results are therefore based on a careful and detailed analysis of different linguistic theories and frameworks, but aiming at reaching a consensus at the level of these "minimal observable facts".
Connected with this basic objective is the approach chosen towards its achievement, an approach which can be defined as looking for an *edited union*[1] of the features proposed in the various major theories and systems.

[1] We are grateful to Gerald Gazdar for suggesting this term as a characterisation of the approach to standardisation undertaken.

This approach tries to capture all the relevant distinctions made by the major lexical theories/systems, without taking a theoretical stand, thereby giving to features labels which are as neutral as possible. By employing this approach, the Lexicon/Syntax group is using the same methodology as that already established within the Lexicon/Morphosyntax group, thus adhering to a coherent path.
As background principles to our work, we aim to be:

- Flexible, in order to accommodate different levels of granularity in the description, still being conformant to the standards;
- Open, in order to accommodate, within the broad EAGLES framework, different theoretical options;
- Consensual, arriving at the proposal through an agreement reached by a number of different key groups working in the field, after a careful survey phase.

In an attempt to be as theory-compatible as possible, there are a few points where choices were left open, especially for those aspects of grammatical description which tend to be more theory-bound (e.g. grammatical relations and control). There are practical drawbacks to this decision — especially with regard to the implementation of the proposed standard — but, at least in this first phase, more importance was given to avoiding committment to specific theories of lexical description. We recognise that there is a tension between the decision to be flexible and open to more than one choice and the real and effective useability of the proposed standards. Without abandoning the principle of flexibility and openness, we can provide an indication of usage by exemplifying the implementation of critical choices.
The last point we want to mention concerns more the methodology of work within EAGLES. In general, the EAGLES results are achieved in a dynamic way, with a cyclical process of revision after one or more phases of testing and feedback, possibly in large projects. The difference between the European approach and other approaches to standards should be pointed out here, to be taken as a description of a general tendency. While in, say, the USA, a sort of *de facto* standard is somehow made available to the community through the provision of publicly available data, in Europe we try to arrive at *consensually agreed* standards. This implies a considerable effort in trying to involve the relevant experts in the different areas of concern, either in the phase of producing the standards, or at least in the successive phases of testing the proposals and providing feedback. This approach also involves a large amount of overheads in terms of activities and work necessary to arrive at a consensus as well as a slower process of arriving at the aimed-for results.

## 2.1 Background and objectives

The specification of subcategorisation structure is perhaps the most urgent and complex type of basic linguistic information which a lexicon suitable for NLP applications must provide. Urgency arises from the need to encode information concerning the local context of words in order to constrain the analysis and generation of natural language. Complexity results from the multidimensional aspect of subcategorisation structure. Because subcategorisation involves reference to diverse levels of grammatical description, the achievement of efficient standards in this area presupposes a minimal normalisation of

- Morphological, syntactic and semantic aspects of word specification; and
- The interface between these levels of grammatical description.

The standardisation of subcategorisation is therefore a crucial task in the provision of a protocol for

- Comparing and exchanging existing lexical resources; and
- Building new monolingual and multilingual lexica which are compatible with or easily adaptable to different grammatical platforms.

Work on subcategorisation has been carried out from a variety of perspectives by linguists, lexicographers and developers of NLP applications for a variety of languages. Consequently, the formulation of standards in this area necessitates a comparison of linguistic theories, practical NLP lexicons, dictionaries and bracketted

corpora as well as language-specific aspects of subcategorisation structure. The main aim of this document is to then assess how and to what extent such a comparison can be carried out in order to achieve an inter-theoretical specification which captures grammatical equivalences among system- or theory-specific coding schemes for the main European languages.
In carrying out this work, our goals are to:

- Define a scheme which standardises approaches to subcategorisation from —
  - Linguistic theories (GB, HPSG, LFG, Unification Categorial Grammar, Dependency Grammar)
  - Practical NLP lexicons (ACQUILEX, COMLEX, EUROTRA, GENELEX, ILC, PLNLP)
  - Bracketted corpora (UPENN, TOSCA, SUSANNE and HELSINKI treebanks)
- Prove the validity of our standardisation scheme for a variety of European languages.

## 2.2 Working procedures

The basic working procedure followed throughout consists in establishing grammatical equivalences relating pairs of system-specific coding schemes. These equivalences provide the basis for an inter-theoretical formalism which is particularly suitable for characterising the weak equivalence of various grammar frameworks and therefore pre-empts the danger that the common specification turns into yet another framework, rather than providing an edited union of existing frameworks. The general strategy adopted in implementing this working procedure can be summarised as follows:

- Standardise basic notions which are necessary in expressing subcategorisation, in the systems and languages considered, by —
  - Comparing the classificatory insights underlying a representative set of system-specific coding schemes for subcategorisation;
  - Establishing a class of equivalences across these classificatory insights; and
  - Using these equivalences to formulate new basic classificatory notions for subcategorisation.
- Use standardised basic notions to define standardised subcategorisation patterns.
- Integrate existing EAGLES work (e.g. Morphology group)
- Collaborate with ongoing EAGLES groups whose work is related to subcategorisation (e.g. Syntactic Annotation).

## 2.3 Summary of report

The work described in this report is the result of collaborative effort involving all contributors, as for all EAGLES work. Indications of individual contributions are given in the Contributors section: these indicate mainly which person was responsible for leading and summarising discussion.
The report is organised into five main chapters. Chapter 3 provides a review and comparison of different approaches to subcategorisation adopted in linguistic theories, practical NLP lexicons and bracketted corpora. In chapter 4, the results of the comparisons drawn in the preceding chapter are used to compile frame-based representations of subcategorisation structures. After a general description of the frame structure, specific details are given concerning the encoding of categorial and morphosyntactic properties, grammatical functions, order, verbal diatheses, control, thematic roles and semantic class. In chapter 5, a formalisation of the encoding scheme presented is given using an object-oriented specification written in SGML. In appendix, a set of macros is given which synthesises the formal frame descriptions presented in chapter 3; using these macros, a list of examples is compiled for each of the languages represented within the group.

# Chapter 3

# Comparing approaches to subcategorisation

## 3.1 Subcategorisation from a theoretical perspective

*Subcategorisation* is concerned with the lexical specification of a predicate's local phrasal context. In general, this specification involves reference to the semantic arguments of the predicate. For example, the verb *devour* subcategorises for two noun phrases insofar as, semantically, it can be characterised as the set of all participant pairs which stand in the *devouring* relation. Indeed, subcategorisation is often contrasted with *collocation* which describes the preferential or statistically salient local phrasal contexts of a word which are not readily expressible in terms of semantic selection — e.g. nominal compounds, adjective–noun pairs.
Semantic selection, however, is neither a sufficient nor a necessary condition for subcategorisation. It is in fact customary to include, in the subcategorisation frame of a predicate, phrases whose occurrence is obligatory in the local phrasal context of the predicate but are not semantically selected by it. Extraposition as well as subject and object raising verbs are the most well-known cases of predicates subcategorising for phrases which they do not select semantically. For example, the subject of an extraposition verb such as *bother* in (1) is a pleonastic element and cannot therefore be regarded as being semantically selected:

(1) It bothers Bill that Mary left

Similarly, a raising verb cannot be regarded as semantically selecting its subject simply because the semantic restrictions on this subject are solely determined by the complement verb. For example, the subject of a raising verb such as *seem* is either contentful or pleonastic (2) according to the complement's subject subcategorisation, e.g.

(2) Mary seems to be happy
It seems to bother Bill that Mary left

Analogous remarks apply to object raising verbs, as in (3):

(3) John believes Mary to be happy
John believes it to be possible that Mary left

Subcategorisation is thus concerned with the specification of phrases which are salient to the local context of a predicate because they are selected by the predicate both syntactically and semantically or simply syntactically. Although all linguistic theories have means to express (morpho)syntactic aspects of subcategorisation, direct reference to semantic selection can only be expressed in those frameworks which include a level of semantic representation (e.g. UCG or HPSG). Nevertheless, because of the importance of semantic selection in subcategorisation, frameworks such as GB or LFG which do not include a level of semantic representation provide a level of linguistic description which expresses the semantic argument structure of predicates in syntactic terms.
Linguistic theories differ in the amount of information which is provided in the subcategorisation frame of a verb. This is mostly due to diverging dispositions to using syntactic rules and principles to express linguistic generalisations, with a consequent shift of emphasis away from or towards lexical specification. For example, control links which obtain with raising and equi verbs can be included in the subcategorisation frame of the relevant predicate as in HPSG/UCG/LFG, or handled in terms of syntactic principles or rules as in GB.

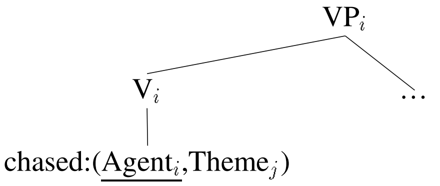
### 3.1.1 Subcategorisation in GB theory

The encoding of subcategorisation in GB can be characterised in terms of Williams' (Williams, 1980; Williams, 1981a) theory of argument structure. Argument structure (AS) is a level of syntactic description where the arguments of a predicate are collected into a set where each element corresponds to an indexed thematic role, as in (4):

(4) chased:(Agent$_i$, Theme$_j$)

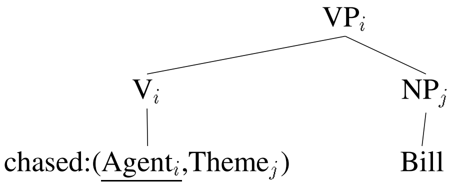
Within the argument structure of a predicate there can be a distinguished position which functions as the 'head thematic role' of the argument structure as a whole (Di Sciullo & Williams, 1987). This thematic role is referred to as the 'external argument' as it can only be assigned outside the maximal projection of its predicate. Given standard conventions about feature percolation (Lieber, 1980) and the notion 'head of a word' (Williams, 1981b; Di Sciullo & Williams, 1987), the index of the external thematic role is passed on to the maximal projection of its predicate as indicated in (5) where the external argument is underlined following Williams' notation.

(5)



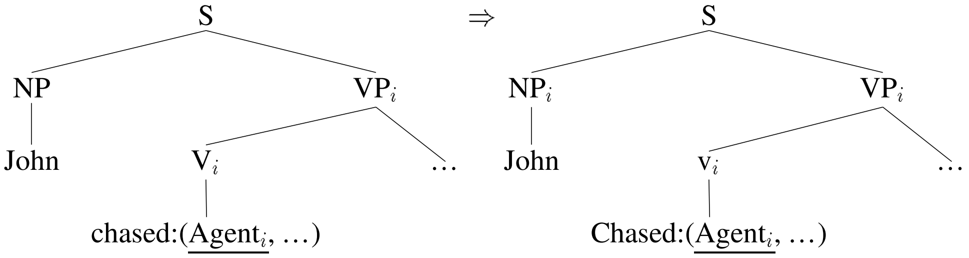

Information about the remaining thematic roles — the 'internal arguments' — is available only within the first projection of the predicate. As shown in (6), the thematic index of an internal argument does not percolate to the VP node, but is assigned within the first projection of the predicate. Such assignment is realised under government, i.e. mutual c–command[1].

(6)




The external thematic role is instead assigned outside the maximal projection of the predicate through *predication*. Predication is stated as a coindexing procedure which relates a predicative phrase with a c–commanding NP at S–structure, as in (7):

(7)




The syntactic realisation of thematic roles in argument structure is constrained and secured by the Projection Principle and the Θ-Criterion (Chomsky, 1981):

**Projection Principle** *Representations at each syntactic level (i.e. LF, and D– and S–structure) are projected from the lexicon, in that they observe the subcategorisation properties of lexical items.*

(Chomsky (1981, 29))

[1] A node *A* c–commands a node *B* iff *A* does not dominate *B* and the first branching node dominating *A* also dominates B.

**Θ–criterion** *Each argument bears one and only one θ-role, and each θ-role is assigned to one and only one argument.*

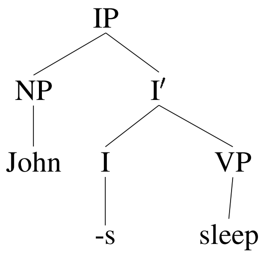
(Chomsky (1981, 36))

In general, it is assumed that categorial selection (c–selection) can be derived as the Canonical Structural Realization (CSR) of its semantic category (Chomsky, 1986b). For example, CSR(patient) is NP. Consequently, the claim is made that only semantic selection (s–selection) needs expressing in the lexicon.
Control with equi verbs is handled syntactically. In brief, the subject of non-finite VPs is structurally represented as the empty category PRO whose relation to its controller is regulated by the binding theory in terms of c–command[2], as in (8):

(8) $\text{John}_i$ wants [$\text{PRO}_i$ to sleep]

This implies that verbal (clausal) subcategorisation is always expressed in terms of sentences rather than VPs. Note that, unlike most other theories of grammar, sentences are assumed to be the maximal projection of the *(I)nflectional (P)hrase* introducing verbal morphology (e.g. tense and aspect) in later versions of GB (Chomsky, 1986a), as in (9):

(9)




Verbal dependencies arising in expletive and subject raising constructions are also handled syntactically (Chomsky, 1981). For example, a raising verb such as *seem* subcategorises for a sentence but has no external argument. If the subcategorised clause is non-finite, the subject moves into matrix subject position to satisfy the *Case Filter* since only a tensed VP can assign nominative case to its subject, as in (10):

**Case Filter** Every lexical NP must be assigned case.

(10) $\text{John}_i$ seems [$\text{t}_i$ to sleep]

If the subcategorised clause is finite, the pleonastic element *it* is inserted in the matrix subject position to satisfy the *Extended Projection Principle* which in addition to the Projection Principle above requires that all clauses have subjects (Chomsky, 1981), as in (11):

(11) __ seems that John sleeps $\rightarrow$ it seems that John sleeps

Object raising constructions are also viewed as involving sentential subcategorisation. A verb such as *believe* subcategorises for an infinitive sentence whose subject is assigned case by the matrix verb across sentential boundaries, as in (12), an occurrence described as *exceptional case marking* (Chomsky, 1986b).

(12) Mary believes [$_S$ John to be intelligent]

### 3.1.2 Subcategorisation in Lexical-Functional Grammar (LFG)

As in GB, subcategorisation in LFG is based on a syntactic representation of predicate argument structure. In addition, the notion of grammatical function occupies a central role in determining which of the arguments semantically selected by a predicate are syntactically realised and how.
In Bresnan (1982), *grammatical functions* are defined as universal syntactic primitives of the grammar and classified according two main parameters: *subcategorisability* and *semantic restrictedness*. Subjects, objects

[2]See Manzini (1983), Chomsky (1986b) and references therein.

and sentential/VP–complements are subcategorisable functions in that they can be assigned to the arguments of lexical items. Nonsubcategorisable functions correspond to adjunct phrases, and — as the term suggests — may not be associated with the arguments of lexical items. The subcategorisability of grammatical functions such as *Topic* and *Focus* is subject to parametric variation and distinguishes between subject-oriented and topic-oriented languages.

Subcategorised functions may differ with respect to the range of argument types with which they can be associated. Grammatical functions which are not inherently tied to specific selectional restrictions are *semantically unrestricted*, while those which can only be paired with arguments of specific semantic types are *semantically restricted*. For example, the subject function can be linked to any thematic role and can also occur as a non-thematic function when encoding the subject-subcategorisation of raising verbs such as 'seem' and 'appear', while oblique functions are always thematic (i.e. they are never associated with pleonastic elements) and are generally more sensitive to selectional restrictions (they are inherently tied to a thematic role (Rappaport, 1983)).

In addition, complements and adjuncts may occur as *closed* or *open* functions. Closed complement and closed adjunct functions (COMP, ADJ) are assigned to clausal expressions which have a controller of their own, e.g. the underlined NPs in (13) and (14):

(13) John believes $[that\ \underline{Bill}\ is\ a\ genius\ ]_{COMP}$

(14) $[\underline{John}\ being\ angry]_{ADJ}$, Mary left

Clausal expressions which are assigned the open complement and open adjunct functions (XCOMP, XADJ) are instead controlled from without, as in (15) and (16):

(15) <u>John</u> wants $[to\ be\ a\ genius\ ]_{XCOMP}$

(16) $[Being\ angry\ at\ John]_{XADJ}$, <u>Mary</u> left

A schematic picture of the grammatical functions adopted in LFG is given in table 3.1.

<table>
<tr><th colspan="3">GRAMMATICAL FUNCTIONS</th></tr>
<tr><th colspan="2">Subcategorisable</th><th>Nonsubcategorisable</th></tr>
<tr><td>Unrestricted</td><td>Restricted</td><td rowspan="2">ADJ (controlled from within)<br>XADJ (controlled from without)</td></tr>
<tr><td>SUBJ<br>OBJ<br>OBJ2</td><td>$OBL_\theta$<br>COMP (controlled from within)<br>XCOMP (controlled from without)</td></tr>
<tr><td></td><td colspan="2">TOPIC<br>FOCUS</td></tr>
</table>

Table 3.1: Grammatical functions in LFG.

*Grammatical relations* are associations of grammatical functions with thematic roles or with non-thematic values. These associations are encoded in the lexicon, where each verb is represented as a *lexical form* consisting of a *predicate argument structure* and a *grammatical function assignment*. The predicate argument structure of a lexical form is a list of the arguments for which there are selectional restrictions. The grammatical function assignment of a lexical form is a list of its syntactically subcategorised functions.

a. predicate argument structure: break⟨agent, theme⟩

b. grammatical function assignment: ((SUBJ), (OBJ))

c. lexical form:

'break⟨ agent theme ⟩'
| |
SUBJ OBJ

The assignment of functions is subject to a number of universal conditions. For example, all monadic predicates are assigned a SUBJ, and all dyadic predicates are assigned a SUBJ and an OBJ. A very important condition on grammatical function assignment is the *Biuniqueness of Function-Argument Assignments* which establishes a one-to-one relation between grammatical functions and arguments within the predicate–argument structure of a lexical form:

> *Biuniqueness of Function-Argument Assignments*
> $G = g_1,\ldots,g_n$ is a possible grammatical function assignment to $P(1,\ldots,m)$ if and only if the mapping from $1,\ldots,m$ to $G$ defined by $i \mapsto g_i$ is injective (one-to-one and into).
>
> (Bresnan (1982, 163))

Grammatical function assignment lists serve as subcategorisation frames. Subcategorisation is checked in *Functional Structure* for *Completeness* and *Coherence* (Kaplan & Bresnan, 1982). *Completeness* ensures that all subcategorised arguments are present in functional structure (e.g. it rules out sentences like **John devours*, **eats a cookie*), while *Coherence* restricts the occurrence of subcategorisable grammatical functions to those listed in the verb's lexical form (e.g. it rules out sentences like **John arrives Bill*).

#### 3.1.2.1 Lexical Mapping Theory

More recently, the theory of grammatical relations in LFG has been revised to make grammatical relation assignment and changing monotonic (Levin, 1987; Bresnan & Kanerva, 1988; Bresnan & Moshi, 1989; Alsina & Mchombo, 1988). Grammatical functions such as SUBJ, OBL, etc., are no longer viewed as atomic specifications, but are defined in terms of more primitive functional features. The emerging theory, *Lexical Mapping Theory*, consists of four basic components, which we now explain.

##### 3.1.2.1.1 Hierarchy of lexical roles

The hierarchy includes the following roles in descending order: agent, beneficiary and maleficiary, recipient and experiencer, instrumental, patient and theme, locative, motive:

**Universal Thematic Hierarchy**

$$ag > ben/mal > recip/exp > ins > pt/th > loc > mot$$

##### 3.1.2.1.2 Decomposed syntactic functions

Syntactic functions are decomposed according to the features [± r] (thematically restricted or unrestricted) and [± o] (objective or not) as follows:

$$\begin{bmatrix} -r \\ -o \end{bmatrix} \text{ SUBJ} \qquad \begin{bmatrix} +r \\ -o \end{bmatrix} \text{ OBL}_\theta$$

$$\begin{bmatrix} -r \\ +o \end{bmatrix} \text{ OBL} \qquad \begin{bmatrix} +r \\ +o \end{bmatrix} \text{ OBL}_\theta$$

Individually, each value for the two features [± r] and [± o] defines a partial grammatical function as indicated below.

| | |
|---|---|
| [-r] | SUBJ/OBL |
| [+r] | $\text{OBL}_\theta$/$\text{OBL}_\theta$ |
| [-o] | SUBJ/$\text{OBL}_\theta$ |
| [+o] | OBL/$\text{OBL}_\theta$ |

#### 3.1.2.1.3 Lexical Mapping Principles

Semantic roles are associated with partially specified grammatical functions according to the following *Lexical Mapping Principles*:

**Intrinsic Role Classifications** For example:

$$\begin{array}{ll} ag & \underbrace{ben/mal > recip/exp > \underbrace{ins > pt/th > loc > mot}_{[+o]}}_{[-r]} \\ | & \\ [-o] & \end{array}$$

**Morpholexical Role Classifications** For example, locative inversion:

$$\begin{array}{ll} \langle th \ldots & loc\rangle \\ & | \\ & [-r] \end{array}$$

**Default Role Classifications**

- Roles lower than theme are [+r]
- Restricted roles are [-o]
- the highest role is [-r], next highest [+o], and the third highest [+r], e.g.

$$\left( \begin{array}{c} top\ \theta \\ | \\ [-r] \end{array} \left( \begin{array}{c} next\ \theta \\ | \\ [+o] \end{array} \left( \begin{array}{c} next\ \theta \\ | \\ [+r] \end{array} \right) \right) \right)$$

#### 3.1.2.1.4 Well-formedness Conditions

After mapping principles have applied, any remaining underspecified grammatical function is fully instantiated. This instantiation is free as long as *Biuniqueness* (17) and the *Subject Condition* (18) are observed.

(17) *Biuniqueness of Function–Argument Assignments*
Within the predicate-argument structure of a lexical form, there is a one-to-one relation between grammatical functions and arguments.

(18) *The Subject Condition*
Every lexical form must have a subject

### 3.1.2.2 Lexical Mapping Theory at work

As an illustrative example, consider the treatment of passive proposed in Bresnan & Kanerva (1988). Prior to passivisation, the theme and agent roles of a verb such as *find* are intrinsically associated with partially specified grammatical functions, as shown in (19).

(19)
$$\begin{array}{lcccccc} & \text{find} & \langle & ag & th & \rangle \\ & & & | & | & \\ intrinsic: & & & [-o] & [-r] & \end{array}$$

As indicated above, the passive rule introduces the functional specification [+r] (i.e. thematically restricted) for the highest role of a lexical form, $\hat{\theta}$, as seen in 20:

(20) passive:
$$\begin{array}{c} \hat{\theta} \\ | \\ [+r] \end{array}$$

When passive applies to the predicate argument structure in (19), the agent role acquires the specification [+r] which in conjunction with [-o] defines an oblique function. The agent argument of a passive verb is thus realised as an oblique complement, while the theme can be either subject or object. Well-formedness constraints induced by the *subject condition* require that the subject option be chosen in this case. Example (21) below provides a schematic representation of the process described as a whole.

(21)

$$
\begin{array}{rlccc}
 & \text{find} & \langle \; ag & th \; \rangle & \\
 & & | & | & \\
intrinsic: & & [-o] & [-r] & \\
passive: & \text{found} & [+r] & & \\
\hline
 & & \text{OBL}_{\hat{\theta}} & \text{OBJ/SUBJ} & \\
wf.: & & \text{OBL}_{\hat{\theta}} & \text{SUBJ} &
\end{array}
$$

#### 3.1.2.3 Functional control

Finally, control relations with raising and control verbs are handled lexically with reference to grammatical functions. For example, subject control with both raising and equi verbs would be stated in the lexicon as indicated by the functional control equations in the relevant parts of of the lexical entries for *seem* (22) and *try* (23)[3]

(22) [PRED = 'seem $<$ ($\uparrow$XCOMP)$>$($\uparrow$SUBJ)'
XCOMP [SUBJ = ($\uparrow$SUBJ)]]

(23) [PRED = 'try $<$ ($\uparrow$XCOMP) ($\uparrow$SUBJ)$>$'
XCOMP [SUBJ = ($\uparrow$SUBJ)]]

Since control is dealt with lexically and no empty categories are used to bind the complement subject, it follows that both raising and equi verbs subcategorise for VPs rather than sentences as in GB.

### 3.1.3 Subcategorisation in HPSG

In HPSG, the repository of information about the subcategorisation of signs is the local syntactic feature SUBCAT. The feature SUBCAT is used to encode the various dependencies that hold between a lexical head and its complements (including subjects). SUBCAT takes a list of (partially specified) *synsems* as its value. Given that *synsems* bear *local* values for CATEGORY and CONTENT[4], lexical signs can exert restrictions on category selection and government as well as case and role assignment.
In HPSG the flow of subcategorisation information up projection paths is handled by a principle of universal grammar — the Subcategorisation Principle — expressed in the following terms:

> In a headed phrase, the list value of
> DAUGHTERS | HEAD-DAUGHTER | SYNSEM | LOCAL | CATEGORY |SUBCAT
> is the concatenation of the list value of
> SYNSEM | LOCAL | CATEGORY | SUBCAT
> with the list consisting of the SYNSEM values (in order) of the elements of the list value of
> DAUGHTERS | COMPLEMENT-DAUGHTERS

Broadly speaking, the Subcategorisation Principle establishes that the SUBCAT value of a phrase is the SUBCAT value of the lexical head minus those specifications already satisfied by some constituent in the phrase. The Subcategorisation Principle, then, works much the same way as cancellation in categorial grammar.
A more recent version of HPSG distinguishes between two features SUBJ and COMPLS, which were conflated in the former feature SUBCAT.

[3] The raising/equi distinction is represented by encoding pleonastic arguments outside the angle brackets which delimit arguments which are semantically selected.

[4] Broadly speaking the CATEGORY attribute of a sign contains information about its part of speech, subcategorisation requirements and possible markers. The CONTENT attribute provides information about its argument structure.

#### 3.1.3.1 Argument structure

One essential function of the SUBCAT feature is to set up the correspondence between grammatical relations (subject, object, etc.) and the roles that are present in the situation described by the verb. In HPSG, then, role assignment is the connection between the constituents of a utterance and the constituents of the thing the utterance is about. Thus, the lexical entry for a ditransitive verb such as *give* assigns semantic roles to its subcategorised-for dependents

$$\begin{bmatrix} \text{pho } \textit{give} \\ \text{synsem} \mid \text{loc} \mid \text{cat} \begin{bmatrix} \text{subcat } \langle\ \text{np}_{\boxed{1}},\text{np}_{\boxed{2}},\text{np}_{\boxed{3}} \rangle \\ \text{content} \begin{bmatrix} \text{reln give} \\ \text{giver } \boxed{1} \\ \text{receiver } \boxed{2} \\ \text{given } \boxed{3} \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

The variables associated with the elements of the SUBCAT list unify with the corresponding variables of the roles in the CONTENT description. Thus the subject variable (first element of the SUBCAT list) unifies with the variable filling the giver role, the first object variable (second element of the SUBCAT list) unifies with the variable corresponding to the giver rol, and, finally, the second object variable (third element in the SUBCAT list) unifies with the variable corresponding to the given role in the verb's described situation.

#### 3.1.3.2 The obliqueness hierarchy

The theory of subcategorisation in HPSG makes essential use of a hierarchical conception of grammatical relations. Leaving aside the subject which has its own list feature, other syntactic functions (such as first object, second object,…) are defined in terms of the order of the corresponding elements on the head's COMPLS list. That is, order on this list corresponds to the traditional grammatical notion of obliqueness of grammatical relations, with more oblique elements occurring further to the left.
Four different classes of linguistic generalisations provide motivation for the hierarchical theory of grammatical relations:

1. **Constituent order generalisations**:
   In many — but not all — languages, the surface order of constituents and their grammatical relations appear to be subject to mutual constraints.

2. **Generalisations involving the theory of control**:
   Controlled complements find their controller in a less oblique co-argument.

3. **Generalisations about the binding of pronouns and reflexives**:
   O–command relations (the HPSG *version* of c–command relations in GB) are expressed in terms of obliqueness hierarchy.

4. **Generalisations about the functioning of lexical rules**:
   For example, passivisation may promote a last — or second-last — NP to a subject position.

#### 3.1.3.3 Category selection

In HPSG, lexical dependencies crucially involve category selection. It is well known that there are subcategorisation restrictions which involve differences of syntactic category which cannot be reduced to semantic or functional distinctions. Examples (24) to (27) below show that verbs, which are semantically very close, impose specific restrictions on the syntactic category of their arguments.

(24) Sandy trusts Kim.
*Sandy trusts on Kim.

(25) Sandy relies on Kim.
*Sandy relies Kim.

(26) Terry is a complete madman.
Terry is out of his mind.

(27) Terry became a complete madman.
*Terry became out of his mind.

Verbs of *trusting* such as *trust* and *rely* in (24) and (25) may have similar argument structures but display different category selection. In HPSG, category selection is achieved in the SUBCAT list specifications. Thus, in (28), the SUBCAT description for *trust* specifies that the category of its second complement is N:

(28) trust: SUBCAT$\langle \ldots,$ SYNSEM|LOC|CAT|MAJ NP$\rangle$

*Relay* not only specifies the category of its complement as prepositional but also demands the preposition be *on*, as in (29):

(29) rely: SUBCAT$\langle \ldots,$ SYNSEM|LOC|CAT [MAJ P, PFORM *on*]$\rangle$

#### 3.1.3.4 Selection of certain morphosyntactic features: vform, case,...

Similar variation is exhibited for finer syntactic selections. For instance, English verbs (including auxiliaries) differ with respect to the inflectional form required on the verb they subcategorise for, as in (30) and (31):

(30) Sandy made Kim throw up.
*Sandy made Kim to throw up.

(31) Sandy forced Kim to throw up.
*Sandy forced Kim throw up.

In order to account for (30) and (31), the corresponding lexical signs must specify different values for the feature VFORM on the xcomp:

$$\begin{bmatrix} \text{phon } \textit{made} \\ \text{synsem} \mid \text{loc} \mid \text{cat} \begin{bmatrix} \text{head} \begin{bmatrix} \text{maj } \textit{v} \\ \text{vform } \textit{finite} \end{bmatrix} \\ \text{valence} \begin{bmatrix} \text{subj} \left\langle \text{NP} \right\rangle \\ \text{compls} \left\langle \text{VP}\left[\textit{base}\right]\text{,NP} \right\rangle \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

$$\begin{bmatrix} \text{phon } \textit{force} \\ \text{synsem} \mid \text{loc} \mid \text{cat} \begin{bmatrix} \text{head} \begin{bmatrix} \text{maj } \textit{v} \\ \text{vform finite} \end{bmatrix} \\ \text{valence} \begin{bmatrix} \text{subj} \left\langle \text{NP} \right\rangle \\ \text{compls} \left\langle \text{VP}\left[\textit{inf}\right]\text{,NP} \right\rangle \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

Likewise, in languages which inflect for case, semantically close verbs may require objects in different cases. Since CASE is a head feature, it follows (in virtue of the Head Feature Principle, which percolates the head features from daughter to mother node) that whenever a lexical form selects a phrasal complement specified

as
|SYN|LOC|HEAD|CASE *ACC* or
|SYN|LOC|HEAD|CASE *NOM*
the lexical head of that complement is also specified in this way.
For languages which lack case inflection (like English or Spanish) an analogous situation obtains with respect to government of the particular preposition that heads a PP complement, as in (32):

(32) Kim relies/depends on Sandy.

Semantically related verbs may assign closely corresponding roles to different PP complements, as in (33):

(33) The authorities blamed Greenpeace for/*with/*of the bombing.
The authorities accused Greenpeace *for/*with/of the bombing.
The authorities charged Greenpeace *for/with/*of the bombing.

These examples show that government of particular prepositions is not semantically predictable, instead, different verbs which take PP complements require different values for the head feature PFORM on that complement.

#### 3.1.3.5 Equi and raising verbs

All equi and raising verbs take a predicative complement the subject of which is *identified* with a co-argument of the verb. They differ in that all subcategorised dependents of equi verbs are assigned semantic role, while one dependent of raising verbs is not asigned a role. In HPSG, this distinction is collapsed in lexical entries. An equi verb such as *try* subcategorises a subject NP and a complement VP. *Try* assigns the tryer role to the referential index corresponding to its subject and the CONTENT value of its VP complement to the psoa[5] argument. In addition, the index of the subject is in structure-sharing with the subject of the VP complement in the SUBCAT list:

$$\begin{bmatrix} \text{phon } \textit{try} \\ \text{synsem} \mid \text{loc} \mid \text{cat} \begin{bmatrix} \text{subcat } \langle \text{NP}_{\boxed{1}}, \text{VP}\big[\text{subcat}\langle \text{NP}_{\boxed{1}}\rangle\big]:\boxed{2}\rangle \\ \text{content} \begin{bmatrix} \text{reln } \textit{try} \\ \text{tryer } \boxed{1} \\ \text{soa-arg } \boxed{2} \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

Given that the subject of the infinitive VP complement is also assigned a role in the infinitive CONTENT description, a sentence such as *John tries to run* would be as follows:

$$\begin{bmatrix} \text{phon } \textit{try} \\ \text{synsem} \mid \text{loc} \mid \text{cat} \begin{bmatrix} \text{subcat } \langle \text{NP}_{\boxed{1}}, \text{VP}\big[\text{SUBCAT } \langle \text{ NP}_{\boxed{1}}\rangle\big]:\boxed{2}\ \rangle \\ \text{content} \begin{bmatrix} \text{reln } \textit{try} \\ \text{tryer } \boxed{1} \\ \text{soa-arg } \boxed{2} \begin{bmatrix} \text{reln } \textit{run} \\ \text{runner } \boxed{1} \end{bmatrix} \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

In the case of raising verbs such as *tend* below, subjects are not assigned a role in the matrix psoa. Here, subject *identification* is not made by structure-sharing of indexes but by structure-sharing of the whole synsem of the subjects. Thus, the SUBCAT list *tend* specifies that the synsem of its subject is structure-shared with the synsem of its subcategorised VP complement:

[5]Parameterised state-of-affairs.

$$\begin{bmatrix} \text{phon } \textit{tend} & \\ \text{synsem} \mid \text{loc} \mid \text{cat} & \begin{bmatrix} \text{subcat} & \langle \boxed{1}, \text{VP}\left[\text{SUBCAT}\langle\boxed{1}\rangle\right]:\boxed{2}\rangle \\ \text{content} & \begin{bmatrix} \text{reln } \textit{tend} & \\ \text{soa-arg } \boxed{2} & \begin{bmatrix} \text{reln } \textit{run} \\ \text{runner } \boxed{1} \end{bmatrix} \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

#### 3.1.3.6 Conclusions

The HPSG treatment of subcategorisation dependencies suggests a uniform treatment of functional, formal and semantic selections, for the values of the feature SUBCAT (or SUBJ and COMPLS) contain all three kinds of information (they are lists of synsems): functional information (e.g. the order of the elements on the COMPLS list); formal information (specifications for values of the attribute CATEG); and semantic information (specifications for values of the attribute CONTENT).

### 3.1.4 Subcategorisation in Categorial Grammar

The Categorial Grammar approach to subcategorisation will be exemplified with a description of the UCG framework used in the ACQUILEX project (Sanfilippo, 1993b). For ease of presentation, only a brief summary is given — see Sanfilippo (1993b) for a full account.

#### 3.1.4.1 Sign structure

Words and phrases are represented as (typed) feature structures where orthographic, syntactic and semantic information is simultaneously represented as a conjunction of attribute–value pairs forming a *sign*:

```
[ORTH: orth
 CAT: cat
 SEM: sem]
```

The category attribute of a sign is either basic or complex. Basic categories are binary feature structures consisting of a category type, and a series of attribute value pairs encoding morphosyntactic information:

```
[CAT-TYPE: cat-type
 M-FEATS: m-feats]
```

Three basic *cat-types* are used

1. n (noun);

2. np (noun phrase); and

3. sent (sentence).

For graphical convenience, basic category structures are abbreviated as follows:

```
cat-type[m-feats]
```

Morphosyntactic features are included only where needed.
Complex categories are recursively defined by letting the type ‘cat’ instantiate a feature structure with attributes RESult, DIRection and ACTive. RESult can take as value either a basic or complex category, ACTive is of type ‘sign’, and the direction attribute encodes order of combination relative to the active part of the sign (e.g. forward or backward):

```
[RES: cat
 DIR: dir
 ACT: sign]
```

In verbs, the active part of the category structure encodes the subcategorisation properties, e.g. subject and object for transitives:

```
[ORTH: < love >
 CAT:[RES:[RES:sent
           ACT:[np-sign
                CAT:nom]]
      ACT:[np-sign
           CAT:np[acc]]]]
```

The semantics of a sign is a *formula*. A formula consists of an index, a predicate and at least one argument which can be either an entity or a formula (which are both subsumed by `sem`)

```
[IND: entity
 PRED: pred
 ARG1: sem]
```

The index of a formula is an *entity* which provides partial information about the ontological type denoted by the formula, e.g. 'e' for eventualities and 'o,x,y,z' for individual objects. In addition a contentless entity, 'dummy', is employed in the semantic characterisation of pleonastic noun phrases, e.g. subject of extraposition verbs. The argument of a predicate can be either an entity or a formula. For ease of exposition formulas are linearised, e.g. the feature structure

```
[IND: [1] x
 PRED: book
 ARG1: [1]]
```

where `[1]` flags reentrant (e.g. identical) values is abbreviated as `<x1>book(x1)` where `x1` is a named variable.

#### 3.1.4.2 Subcategorisation

The classification of subcategorisation types involves defining

- Semantic predicate–argument structures
- Category structures
- Verb signs where links are made across the semantic and category structures

#### 3.1.4.3 Semantic predicate–argument structures

Verbs are characterised as properties of eventualities, and thematic roles are relations between eventualities and individuals, e.g.

```
<e1>and(<e1>sleep(e1), <e1>agent(e1,john))
```

Following Dowty (1989), the semantic content of thematic relations is expressed in terms of prototypical *cluster-concepts* — the proto-agent and proto-patient roles ('p-agt', 'p-pat') — determined for each choice of predicate through attribution of selected entailments which qualify the relative agentive strength and affectedness of event participants. Dowty's insights are augmented by introducing a third proto-role, 'prep' for prepositional arguments ('semantically restricted' in LFG terms) and the contentless predicate 'no-$\theta$' to characterise the relation between a pleonastic NP to its governing verb. In addition, proto-roles are formalised as

supersets of specific clusters of meaning components which are instrumental in the identification of semantic verb classes (Sanfilippo & Poznański, 1992; Sanfilippo, 1993b; Sanfilippo, 1993a) — see section 4.8 for examples.
A primary semantic classification of verb types is obtained in terms of argument arity. Further distinctions are made according to what kind of verbal arguments are encoded:

- proto-agent, e.g. 'John' in "John sleeps" and the sentences below
- proto-patient, e.g. "a book" in "John read a book"
- prepositional
  - oblique/indirect, e.g. "to Mary" in "John gave a book to Mary"
  - objective, e.g. 'Mary' in "John gave Mary a book"
- non-thematic, e.g. 'Bill' in "Bill seems to be sad"
- pleonastic, e.g. 'It' in "It bothers Bill that Mary left"
- predicative ('xcomp'), e.g. "to leave" in "John wishes to leave"
- sentential ('comp'), e.g. "that Mary left" in "John said that Mary left"

Here are some of the semantic structures distinguished:

- For strict intransitive verbs, e.g. "John sleeps":

```
STRICT-INTRANS-SEM
   <e1>and(<e1>pred(e1), <e1>p-agt(e1,x)
```

- The semantics of strict transitive verbs, e.g. "John drinks a beer":

```
STRICT-TRANS-SEM
   <e1>and(<e1>pred(e1), <e1>and(<e1>p-agt(e1,x),
                                 <e1>p-pat(e1,y)))
```

- For transitives taking an oblique complement and ditransitives, e.g. 'Mary gave a book to Bill" and "Mary gave Bill a book":

```
OBL-TRANS/DITRANS-SEM
   <e1>and(<e1>pred(e1), <e1>and(<e1>p-agt(e1,x),
                                 <e1>and(<e1>p-pat(e1,y),
                                         <e1>prep(e1,y))))
```

- Intransitive verbs with a thematic subject which take a clausal complement, e.g. "John intended to come' ' and "Bill thought that John would come":

```
P-AGT-SUBJ-INTRANS-XCOMP/COMP-SEM
   <e1>and(<e1>pred(e1), <e1>and(<e1>p-agt(e1,x), verb-sem))
```

#### 3.1.4.4 Category structures

Category structures are distinguished according to the values for the features RES and ACT. For example, the CAT of strict intransitives states that the result is a basic category of type 'sent' and the active part is a noun phrase (i.e. there is only subject selection):

```
STRICT-INTRANS-CAT
        [RES: sent
         ACT: np-sign]
```

More complex category types can be built using more basic category types, e.g.

- For strict transitives e.g. "Bill ate a sandwich"

```
STRICT-TRANS-CAT
        [RES: strict-intrans-cat
         ACT: [np-sign
               CAT: np[acc]]]
```

Morphosyntactic restrictions are encoded in selected (active) signs. For example, the outermost argument bears accusative case in the definition of the ditransitive category and prepositional case ('p-case') in the category definition for transitives which take a PP complement:

- For e.g. "John gives Mary a book"

```
DITRANS-CAT
        [RES: strict-trans-cat
         ACT: [np-sign
               CAT: np[acc]]]
```

- For transitives which take an prepositional complement e.g. 'John gave a book to Mary"

```
OBL-TRANS-CAT
        [RES: strict-trans-cat
         ACT: [np-sign
               CAT: np[p-case]]]
```

The remaining category types are organised into:

- 'comp-cat' for verbs which take a sentential complement;
- 'xcomp-cat' for verbs which take a predicative complement, further divided according to whether control is involved or not.

Control categories are used to describe the syntactic structure of both equi and raising verbs. All control categories follow (inherit from) the following pattern where the reentrancy tag `[1]` says that the complement active sign (e.g. the complement subject) is controlled by the immediately preceding active sign (control is expressed by equating entities which partially describe the semantics of active signs):

```
CONTROL-CAT
        [RES: [RES: cat
               ACT: [sign
                     SEM:ARG2: [1] entity]]
         ACT: [sign
               CAT:ACT: [sign
                         SEM:ARG2: [1]]]]
```

The controlling argument can be the subject or the object according to whether the verb is transitive or intransitive (transitivity is determined by the presence of an accusative active np-sign).

```
INTRANS-CONTROL-CAT
        [RES: [RES: sent
               ACT: [sign
                     SEM:ARG2: [1] entity]]
         ACT: [sign
               CAT:ACT: [sign
                         SEM:ARG2: [1]]]]

TRANS-CONTROL-CAT
        [RES: [RES: strict-intrans-cat
               ACT: [np-sign
                     CAT: np[acc]
                     SEM:ARG2: [1] entity]]
         ACT: [sign
               CAT:ACT: [sign
                         SEM:ARG2: [1]]]]
```

Actual control categories are built adding further specialisations to the control descriptions above. For example, the category structure for intransitive equi verbs is defined as follows:
For intransitive control verbs which take an infinitive VP complement e.g. "Jon wants/seems to leave"

```
INTRANS-VPINF-CONTROL-CAT
Inherits from INTRANS-CONTROL-CAT
        [RES: strict-intrans-cat
         ACT: [vp-sign
               CAT:RES: sent[fin]]]
```

#### 3.1.4.5 Verb signs

Verbs signs are defined by linking active signs in the category structure to argument slots in predicate argument structures. This is done by means of reentrancy links, as indicated by the tag `[1]` in the structure for strict intransitive verbs below.

```
[strict-intrans-sign
 CAT:ACT: [np-sign
           SEM: [1] <e1>p-agt(e1,x)]
 SEM: [strict-intrans-sem
       <e1>and(<e1>pred(e1), [1])]]
```

Since only templates for verbs which have a maximum of 3 arguments are given, only two additional general linking patterns are needed:

```
[two-arguments-verb-sign
 CAT: [RES: [RES: sent
             ACT: [sign
                   SEM: [1]]]
       ACT: [sign
             SEM: [2]]]
 SEM: <e1> and(and(pred(e1),[1]),[2])]

[three-arguments-verb-sign
 CAT: [RES: [RES: [RES: sent
```

```
                    ACT: [sign
                           SEM: [0]]]
             ACT: [sign
                    SEM: [1]]]
      ACT: [sign
             SEM: [2]]]
SEM: <e1> and(and(and(pred(e1),[0]),[1]),[2])]
```

To conclude, here are some sample two-arguments-verb-sign and three-arguments-verb-sign structures

#### 3.1.4.5.1 TWO-ARGUMENTS-VERB-SIGN

- The type of strict transitive e.g. "John reads a book":

```
STRICT-TRANS-SIGN
         [CAT: strict-trans-cat
          SEM: strict-trans-sem]
```

- For subject equi verbs which take an infinitive VP complement, e.g. "Jon wants to leave":

```
SUBJ-EQUI-INTRANS-VPINF-SIGN
         [CAT: intrans-vpinf-control-cat
          SEM: p-agt-subj-intrans-xcomp/comp-sem]
```

#### 3.1.4.5.2 THREE-ARGUMENTS-VERB-SIGN

- Ditransitives, e.g. "John gives Mary a book":

```
DITRANS-SIGN
         [CAT: ditrans-cat
           SEM: obl-trans/ditrans-sem ]
```

- Transitives which take an oblique object. According to Dowty's obliqueness hierarchy, the oblique complement ('np[p-case]') is the outermost subcategorised argument although it is the direct object NP which follows the verb , e.g. 'John gives a book to Mary":

```
OBL-TRANS-SIGN
         [CAT: [RES: strict-intrans-cat
                ACT: [np-sign
                       CAT: np[p-case]]]
          SEM: intrans-obl-sem]
```

Note that subcategorised arguments are positioned in the category structure of predicates according to the 'obliqueness hierarchy' as in HPSG, see section 3.1.3. For example, the 'goal' argument of ditransitive and transitives which subcategorise for a PP (`DITRANS-SIGN` and `OBL-TRANS-SIGN` above) is the outermost sign in the category structure, even though only in ditransitives does it precede the 'theme' object (the difference in word order is handled syntactically, see Sanfilippo (1993b) and references therein).

### 3.1.5 Subcategorisation in Dependency Grammar

Dependency Grammar resorts to the old idea that syntax is almost entirely a matter of the combination capabilities and completion requirements of words. Since the influential work of Tesnière (1959), the model for describing these phenomena is akin to the formation of molecules from atoms in chemistry. Like atoms, words have *valency*; they are apt to combine with a certain number and kind of other words forming larger chunks of linguistic material. Valency descriptions of words do not only correspond with the subcategorisation component in other frameworks but are the sole device of generating syntactic structures in the lexicalist versions of dependency grammar.

Dependency grammars, like phrase structure grammars, use trees (directed acyclic graphs) in order to depict the structure of a given phrase or sentence. While a phrase structure grammar associates the nodes in the tree with larger or smaller constituents and uses the arcs to represent the relationship between a part and the whole, all nodes in a dependency tree represent elementary constituents and the arcs denote the direct syntagmatic relationships between such elements.

Dependency grammar assumes that there is usually an asymmetry between immediate constituents of a phrase: one constituent is the *governor* or head, the other ones are the *dependents*. For example, one constituent may be a predicate, the other constituents are arguments; one constituent may be a modifier, the other one is modified. The governor gives rise to the whole construction and determines it. Dependents adjust to the demands imposed on the construction by the governor. The difference between heads and dependents is reflected by the hierarchy of nodes in the dependency tree.

Since each node represents an elementary segment (a terminal category), the nodes in a dependency tree are typically labeled by lexemes.

The sentence in (34) may be represented by the bracketed expression of (35), which is equivalent to a dependency tree:

(34) The big dog chases the cat.

```
(35)  [chase
          [dog
              [the]
              [big]]
          [cat
              [the]]]
```

It is no problem to add part-of-speech categories to the node labels, as in (36):

```
(36)  [V chased
          [N dog
              [Det the]
              [Adj big]]
          [N cat
              [Det the]]]
```

The dependency structure is especially suitable for functional categorisations, as in (37):

```
(37)  [PRED chased
          [SUBJ dog
              [DET the]
              [ATTR big]]
          [DIROBJ cat
              [DET the]]]
```

Both functional and morphosyntactic categories can be merged in the same tree, as in (38). Word order, which is not inherent in the tree structure, can be depicted by additional labeling, e.g. by the symbol ‘$<$’ for “left of the head” and ‘$>$’ for “right of the head”:

(38)
```
[PRED chased V
        [< SUBJ dog N
                [< DET the Det]
                [< ATTR big Adj]]
        [> DIROBJ cat N
                [< DET the Det]]]
```

A common misconception of the dependency relation must be mentioned here. Dependency is *not* a word-to-word relationship but a word-to-complement relationship. A complement can consist of many words. For example, the noun phrase *the big dog* is the subject complement of the verb *chase* in the above sentence rather than just the word *dog*; the phrase *the cat* is the object rather than just the word *cat*. Internally, any phrase is again structured according to word-to-complement principles and is represented as such. Therefore, all nodes in a dependency tree eventually correspond to terminal elements. Nevertheless, a dependency relationship only exists between a word at a dominating node and the whole phrases represented by the dependent subtree. As a consequence, the part-of-speech categories associated with the nodes are not really terminal. Similar to $\overline{X}$ theory, the categories of the lexical head in a phrase characterise the whole phrase.
Quite a few Dependency Grammars have recently been developed. See Fraser (1994) for a survey. In the following, we describe the role of subcategorisation in Dependency Unification Grammar (DUG). DUG has been implemented at the Institute for Computational Linguistics of the University of Heidelberg as a framework for parsing natural languages, see Hellwig (1983). DUGs for German, French and English have been drawn up for the ESPRIT and LRE projects “Translator’s Work Bench” (TWB) and “Selecting Information from Text” (SIFT).
Sentence (39)

(39) Arthur attends the EAGLES meeting.

is turned into the representation of (40) by the parser:

(40)
```
(ILLOC: statement': sign
   (< PROPOS: attend present': verb form[finite] person[he,U]
   s_type[statement] s_position[4,6,17] adjacent[left] margin[left]
      (< SUBJECT: Arthur: noun person[he,C] determined[+]
      s_position[4])
      (> DIR_OBJ1: meeting: noun person[it] determined[+,U]
      reference[definite,U] s_position[17] n_position[2,9,10]
         (< DETER: definit': determiner determined[+,C]
         (reference[definite,C] n_position[2])
         (< ATTR_NOM: EAGLES: noun determined[-] (n_position[9]))));
```

Without going into details, this much should be noted: (40) is a dependency tree with six nodes, one node for each elementary constituent from (39), including the full stop. Each node carries three types of information: a syntactic function (ILLOC, PROPOS, SUBJECT, DIR_OBJ1, DETER, ATTR_NOM), a lexeme (statement’, attend, Arthur, meeting, definit’, EAGLES) and a set of morphosyntactic features introduced by a part of speech category (sign, verb, noun, determiner).
What are the lexical prerequisites for the parser to construct such an output? As opposed to common generative grammars, a DUG does not consist of a set of production rules from which a formal representation is created in a separate step. DUG directly describes syntactic structures as fragments of the target representation. Basically three data sets are needed: a morphosyntactic lexicon, a set of valency templates and a valency lexicon.
The morphosyntactic lexicon relates each elementary string of the language (i.e. each word form) to a lexeme and a complex morphosyntactic category.
A valency template directly mirrors a fragment of a dependency tree. As such, it describes a specific syntagmatic relationship, e.g. the relationship between a verb and a subject or a verb and a direct object, between a noun and a determiner or two nouns in a compound, etc. Templates typically contain a dominating term which stands for the head node in the syntagmatic relation, and a dependent term functioning as a variable

for a dependent node. The latter is called a *slot*. Templates describe the combination capability of words by means of slots which are to be filled by appropriate material from the context. The set of templates in a DUG has a similar function as the set of rules in a generative grammar.
The following templates in (41) have been used in the tree in example (40):

(41)

```
(ILLOC: +statement: sign
    (< PROPOS: = : verb form[finite] s_type[statement]
    adjacent[left] margin[left]));

(*:+subject:verb form[finite,subjunctive]
s_type[statement,relative] s_position[6]
    (< SUBJECT:=:noun person[C] determined[+]
    s_position[4]));

(*:+dir_obj1:verb obj_number[singular] mood[active]
s_position[6]
    (> DIR_OBJ1:=:noun person[I,you,he,she,it]
    determined[+] s_position[17]));

(*: %dete_count_any: noun count[+] n_position[10]
    (< DETER: _ : determiner determined[C] reference[C]
    n_position[2]));

(*: %attr_nominal: noun n_position[10]
    (< ATTR_NOM: _ : noun determined[-] punct2[hyphen]
    n_position[9]));
```

Note that each template characterises the morphosyntactic form of the head, the syntactic function of the dependent and the morphosyntactic form of the dependent. If need be, lexical selections can also be specified in a slot.
The valency lexicon consists of references. A reference assigns a template or a set of templates to a lexical item, thus describing the combination capability of the item. There are three kinds of references according to the possible functions of templates: complements, adjuncts and conjuncts.
DUG makes a theoretical and technical distinction between complements and adjuncts. Theoretically, complements are dependents of a lexical item that are required by the word's inherent combinatorial semantics. Adjuncts, on the contrary, augment the dependency structure by their own virtue. While a term is incomplete until it has found its complements, adjuncts can be added to the set of dependents of a term in a relatively arbitrary fashion.
As a rule, complements are specified in the lexicon under the lemma of the governing term (i.e. in a top down fashion). Adjunct templates are specified in the lexical entry of the adjunct word, defining the potential of attachment of the lexical item as a dependent (i.e. in a bottom up fashion).
A conjunct is a phrase, usually introduced by a coordinator like *and* and *or*, which is attached to the rest of a sentence resulting in the coordination of some of its parts. The references of (42) link up the lexical items in sentence (39) with the templates of (41). (The full stop has the lexeme *statement'* and the determiner *the* is represented by the lexeme *definite'*.)

(42)

```
(:COMPLEMENTS (*:statement':sign) (: +statement));

(:COMPLEMENTS (*:attend:verb) (&(: +subject)(:+dir_obj1)));

(:ADJUNCT (*:definite':determiner) (: %dete_count_any));

(:ADJUNCT (*:EAGLES: noun) (: %attr_nominal));
```

These references are in fact the means how subcategorisation is implemented in a DUG. Remember each template which is mentioned in a reference is a combination of a syntactic function and a morphosyntactic realisation. There can be more than one template with the same name in order to characterise syntagmatic alternations of realisations. For example, among the subject templates are the (43) and (44) that describe the subjects in questions:

(43) (*:+subject:verb auxiliary[+] form[finite,subjunctive] s_type[question] s_position[6]
(> SUBJECT:=:noun person[C] determined[+] s_position[7]));

(44) (*:+subject:verb form[finite,subjunctive] s_type[question, relative] s_position[6]
(< SUBJECT:=:pronoun pro_form[interrogative, relative,C] person[C] gender[C] case[subjective] n_position[2]));

The first subject template (43) describes the subject of *Did Arthur attend the meeting* and the second template (44) accounts for *Who did attend the meeting?*. Both templates are already covered by the reference for *attend* in (42).
Alternations of templates can also be stated as an explicit disjunction in a reference, as in (45):

(45) (:COMPLEMENTS (:surprise) (& (, (:+subject_that) (:+subject_inf_to) (:+subject_ing)) (:+dir_obj1)));

This description covers *That he came surprises me*, *To see him here surprises me*, *Seeing him here surprises me*.
DUG describes the argument structure as a level of syntactic description. Usually, there is no ranking of participant roles. On the contrary, the subject is considered to be an argument of the verb as anything else (e.g. objects). Control and extraposition structures are handled by assignment of specific templates to the verbs that give rise to these structures. There is a mechanism of expansion rules (not covered in this survey) which creates additional templates on the basis of the ones in the lexicon, e.g. the template for the agent slot in the passive is created from the lexical coding of a direct transitive object. The same is true for the slot of *it* as a correlate with a subject or object clause.

### 3.1.6 A comparison

In all the approaches reviewed, subcategorisation involves an *argument structure* providing a specification of the participant structure of a predicate, and a *linking* specification which determines the functional/syntactic realisation of participant roles of the predicate.
In GB, LFG and DUG, argument structure is defined as a level of syntactic description; in HPSG and UCG, argument structure is part of the semantic representation of predicates. Thematic role annotations are used in GB, LFG and UCG to rank participants; this ranking determines the syntactic functionality of participants. In GB, the external/internal argument distinction is crucial in the determination of the structural realisation of participant roles. This distinction is not semantically motivated; it is simply assumed as a lexical specification. In LFG, a Universal Thematic Hierarchy is assumed. In the 1994 version of HPSG, there is no ranking of participant roles[6]. In UCG, $\theta$ ranking is semantically expressed in terms of Dowty's prototypical roles augmented with the notion of semantic restrictness used in LFG. No ranking of thematic roles is used in DUG.
In GB, LFG, HPSG and UCG, grammatical relations are viewed as links between (syntactically) selected constituents and $\theta$ roles. In GB, the thematic functionality of argument structure participants is directly projected in syntactic structure. Grammatical relations such as subject and objects are expressed as predication and government relations between nodes in tree structures. In LFG, HPSG and UCG, grammatical relations are specified in the lexicon. In LFG, this specification is given by the Lexical Mapping Principles which governs the linking of $\theta$ roles to primitive grammatical function features in lexical forms. In both HPSG and UCG, arguments are syntactically ranked according to the obliqueness hierarchy which reproposes the functions subject, object and indirect object in terms of relative position in the subcategorisation list. In DUG,

[6]Some proposals have been recently made to fill this gap, see Davis (1996) and Sanfilippo (1995a), Sanfilippo (1995b).

linking is expressed lexically as elementary form-function units; grammatical functions are viewed as primitives.

Further differences concern the treatment of control and extraposition structures with clausal complements. In GB, both equi and raising verbs subcategorise for a sentence. With equi and subject raising verbs, control is expressed syntactically. The complement subject of equi verbs is an empty category which is syntactically controlled according to the binding theory. With subject raising verbs, the complement subject moves into the matrix subject position to receive case leaving behind a coindexed empty category. Object raising verbs are treated as a case of straight sentential complementation, i.e. no control is involved, since the complement subject is allowed to receive case *in situ* by the matrix verb. In LFG, HPSG, UCG and DUG equi and raising verbs subcategorise for a verb phrase; control is lexically specified for all, including object raising verbs. A common trait to all is the need to specific the control propensity of a verb (e.g. subject or object control), which is done lexically. Finally, extraposition verbs include lexical specification of the pleonastic subject / object NP in LFG, HPSG, UCG and DUG, but not in GB where the pleonastic NP is inserted in syntax following movement of the argment which is amenable to extraposition.

Table 3.2 provides a schematic summary of the comparison drawn above. In addition, all the frameworks

| Information Types | GB | LFG | HPSG | UCG | DUG |
|---|---|---|---|---|---|
| Argt. structure<br>– Rep. level<br>– Θ ranking | syntax<br>ext/internal | syntax<br>$\theta$ Hierarchy | semantics<br>in prep. | semantics<br>proto-roles | syntax<br>no |
| Gram. Relations<br>– Gram. Functions<br><br>– Rep. level<br>– Linking | derived<br><br>syntax<br>predication & govt | decomposed into primitives<br>lexicon<br>Lexical Mapping Principles | derived<br><br>lexicon<br>obliqueness | derived<br><br>lexicon<br>obliqueness | primitive<br><br>lexicon<br>form-function units |
| Equi/Raising<br>– Clausal complement | sentence | verb phrase | verb phrase | verb phrase | verb phrase |
| Obj Raising<br>– Control expressed | no control | lexically | lexically | lexically | lexically |
| Equi/Subj Raising<br>– Control expressed<br>– Control propensity specified | syntactically<br>lexically | lexically<br>lexically | lexically<br>lexically | lexically<br>lexically | lexically<br>lexically |
| Extraposition<br>– Pleonastic NP | introduced in syntax | subcategorised | subcategorised | subcategorised | subcategorised |

Table 3.2: Comparison of subcategorisation in contemporary syntactic theories

reviewed make use of features such as case and verbal form (e.g. infinitive, gerundive) to express morphosyntactic properties of subcategorised phrases. The correspondences resulting from this comparison suggest the following minimal list of general core notions as the edited union of the various basic notions which characterise subcategorisation structure in the frameworks taken into consideration:

**Argument structure** — Predicate and arguments associated with ranked participant roles.

**Grammatical relations** — Linking of arguments in argument structure to grammatical functions governed by participant role functionality.

**Equi** — Controller & controllee are both thematic; control propensity specified lexically.

**Raising** — Only controlled position is thematic.

**Expletive Construction** — Pleonastic argument is found in either subject or object position.

**Morphosyntactic features** — Such as case and verbal form for subcategorised phrases.

We shall seein chapter 4 how these general common core notions can be used to develop subcategorisation standards.

## 3.2 Practical NLP lexicons

In this section an overview of the following practical lexicons for NLP is provided:

- ACQUILEX: a typed feature structure, multilingual lexicon developed within the Acquilex project for four different languages (Dutch, English, Italian and Spanish) following a sign-based approach to lexical specification (Sanfilippo, 1993b).
- COMLEX: the American English lexicon prepared for the Linguistic Data Consortium by the University of Pennsylvania, which uses a typed feature structure formalism (Rohen Wolff, Macleod & Meyers, 1994).
- EUROTRA: the lexicon for generation and analysis in the Eurotra Machine Translation system (ten Hacken, Maas & Maegaard, 1991).
- GENELEX: a full-scale application- and theory-independent lexicon developed within the Genelex project (GENELEX, 1993). It is assumed that from this general repository of lexical information theory-dependent lexical representations and/or application-specific dictionaries can easily be derived. The general architecture of the lexicon is conceptually based on the entity–relationship model.
- ILCLEX: an Italian lexicon, originally designed for integration with a robust wide-coverage corpus grammar, which collects information extracted from machine readable sources such as dictionaries and corpora (Vanocchi, Rosini, Carenini, Prodanof & Calzolari, 1994).
- LDOCE: the Longman Dictionary of Contemporary English (Procter, 1987), which was taken as a basis for the construction of several computational lexicons (see, for instance, Boguraev & Briscoe (1989).
- PLNLP: the lexicon used by the PLNLP Italian grammar, originally developed for a style-checking application, and later used for other different tasks, e.g. the acquisition of lexico-semantic information from machine readable dictionaries (Jensen, Heidorn & Richardson, 1993).

The lexicons differ largely as to the number of entries they contain and to the amount of linguistic information each entry is provided with: while some lexicons attach more importance to coverage, others appear to concentrate on granularity and depth of lexical representations often to the detriment of breadth. This is mainly due to the fact that most such lexicons were aimed at a wide variety of different applications (ranging from machine translation to style-checking). Moreover, some are strongly biased towards specific linguistic theories. For the sake of concreteness, a specific case study was chosen as a basis for comparison between these lexicons: the encoding of verb entries and, particularly, of their subcategorisation frame. Although this area has been the focus of intense research over the last fifteen years, it still rests on fairly shaky ground.
In what follows, we will list all aspects which appear to be relevant for the encoding of subcategorisation and consider them in more detail. It should be noted that not all of the following are pertinent to all lexicons, as will be indicated below:

- Number of arguments;
- Syntactic category of arguments;
- Functional role of arguments;
- Control and raising phenomena;

- Lexical selection;
- Morphosyntactic constraints;
- Frame alternation and argument optionality;
- $\Theta$ structure and deep structure;
- Other properties.

The division between the top of this list and the bottom is to be interpreted as indicative of a scale from the least to the most uniformly-distributed properties: in other words, the properties in the topmost part of the list are shared by all lexicons, the others are shared only by some and with considerable differences in the way these properties are represented therein. In what follows, we will first sketchily characterise the approach to verb subcategorisation that each lexicon takes, then move on to a more detailed consideration of the above listed aspects.

### 3.2.1 Approaches to verb subcategorisation

In this section, the different approaches to verb subcategorisation taken by the abovementioned lexicons are briefly described. We will mainly focus on the following general aspects:

- Theoretical background explicitly acknowledged in the lexicon (if any);
- Number of levels of linguistic description on which the lexical representation hinges;
- Lexical entry model;
- Governing criteria for entry identification and distinction (word sense, part of speech, subcategorisation pattern).

The treatment of subcategorisation for the **Acquilex** lexicon (Sanfilippo, 1993b) was developed using a Unification Categorial Grammar model (hereafter UCG) combined with a sign-based approach to lexical representation. While the description of verb types in Acquilex was meant to be compatible with several theoretical approaches, reference to a specific grammar framework was deemed to be necessary in order to test the fragment built (i.e. parse representative sentences). UCG was chosen because it provided a radically lexical theory of grammar; this ensured that lexical entries would contain as much information as can be appropriately captured in the lexicon. Consequently, modification for a different framework would be a process of (automatically) removing or transforming information, but not (manually) adding more. Word senses are criterial for identification/distinction of an independent lexical entry.

**ILCLEX** is intended to provide a theory-independent lexical representation. Three hierarchically ordered information levels were identified: pattern rules, patterns and subcategorisation frames. At the bottom level, subcategorisaton information corresponds to an explicit encoding of the various grammatical contexts the verb can occur in: the syntactic realisation of arguments is taken to be criterial for identification of a single subcategorisation frame. Patterns generalise over different sucategorisation frames abstracting away from the syntactic realisation of arguments. Pattern rules, at the top level, link different but related patterns of the same verb to one another. Note that the same pattern as well as the same subcategorisation frame can refer to different verb senses.

The **Comlex** subcategorisation model is based on the set of subcategorisation classes developed for Sager's Linguistic String Project Dictionary, with influences from the Brandeis Verb Lexicon. The dictionary consists of a list of detailed definitions of subcategorisation features and a set of word definitions which reference those features. The verb frame structure is articulated into two levels, the constituent structure (a list of surface structure constituents such as NP, PP, ADJP etc., immediately followed by a number used to coindex the constituent and grammatical structures), and the grammatical structure (a list of grammatical relations each followed by an index referring to an element from the constituent structure). The identity of a lexical entry is defined in syntactic terms only, regardless of sense variation.

The stratificational nature of the **Eurotra** unification-based grammar implies four different levels of linguistic description: the morphological level, the configurational or constituent level, the relational syntactic level and the interface or semantic level. Each level has its own grammar and its own lexicon. All of them, except the first one, could be considered relevant to the issue of verbal subcategorisation. In what follows, the **Eurotra** lexicon will be taken as a whole, and an indication of the linguistic level to which a specific aspect pertains is provided only when and if relevant. Note that at the interface structure level, different senses of the same verb give rise to different lexical entries. This does not hold at lower levels where syntactic criteria, either constituency-based or relational, are relevant to entry identification.
In the **Genelex** model, which is aimed at being theory-welcoming, an entry is a complex and structured object defined in terms of: a lexical unit, its syntactic behaviour and its complementation frame (subcategorisation frame in our terminology). For each complement within a complementation frame, its position and syntactic realisation is specified together with all sorts of restrictions when needed. Some notions are parameterisable: for instance, a complement can be defined either in purely syntactic terms (distribution paradigm, function) or in semantic ones ($\theta$ roles, semantic classes). Complementation patterns are not defined a priori, but can be tailored by the lexicographer to meet specific needs/applications. Both fine and coarse grained information can thus be couched in the model. At the moment, the Genelex dictionary contains, for each word, its complementation pattern specified for number of complements, their optionality (when relevant), their syntactic realisation, and lexical or morphosyntactic restrictions.
The **PLNLP** lexicon is a broad coverage lexicon, basically consisting of a list of word stems and fairly simple feature information associated with them, i.e. part of speech, morphological and subcategorisation features. It adopts a compact lexical representation strategy, since all different subcategorisation features for the same verb are collapsed within the same lexical entry regardless of sense distinctions.
In addition to part of speech information **LDOCE** specifies, for each entry, a subcategorisation description in terms of types and numbers of complements.When the same verb can take different types of complements without significant change in meaning, these are encoded as alternative subcategorisations (alternations) within the same entry. In the late 80s, LDOCE represented the most comprehensive description of grammatical properties of words to be found in any published dictionary available in machine readable form. This is why it is certainly worth considering in this overview of practical NLP lexicons mainly as a source of computational lexicons rather than a computational lexicon in its own right.

### 3.2.2 A comparative overview

In what follows, the different lexicons are characterised in more detail with respect to those aspects which are taken to be relevant to the present overview. These aspects define a data-driven, observational basis of comparison which will be beneficial to an overall assessment of the quantity and quality of linguistic information that is shared as a common core by all lexicons.

#### 3.2.2.1 Number of arguments

There appears to be substantive agreement on the presence or need of information on argument number, which is coded either directly, or indirectly through argument listing within the relevant subcategorisation pattern or through use of shorthand grammar codes. This issue is strictly related to the way droppable arguments are represented (argument optionality): a verb with two arguments, one of which is droppable, can either be represented as two separate entries (one with two arguments and the other with the obligatory argument only), or as one complex entry, where argument optionality is expressed as an extra property of a single frame. For more detailed information on these aspects, see the relevant subsection (3.2.2.7). Here, we concern ourselves only with whether and how argument number information is taken into account.
The only lexicon which explicitly encodes this information through an appropriate feature which takes a numeric value is **ILCLEX**. As to the others, this information is always to be somehow inferred.
In **Acquilex** a primary classification of verb types is obtained in terms of predicate arity, which is logically specified as a conjunction of formulae whose main predicates are (thematic) relations between eventualities and individuals. The number of arguments can be inferred from the number of such individuals.

In **Eurotra**, the number of obligatory complements/arguments of the verb is treated at the levels of both relational and interface structure. The maximum number of complements in the relational structure is determined by the maximum number of arguments in the interface structure and is equal to 4, a number judged appropriate for general language.
The number of complements in **Genelex** corresponds to the number of complement positions defined within the subcategorisation pattern of each lexical entry. There is no limit to the number of complements; the maximum number is determined by the lexicographer.
In **Comlex** the number of arguments can be inferred from the number of complements specified at the level of grammmatical structure. Note that Comlex, at the level of constituent structure, does not specify the subject argument in the complement list, unless the subject has to be morphosyntactically constrained somehow (as in verbs which take plural subjects only: see also section 3.2.2.6.
In both **LDOCE** and **PLNLP**, subcategorisation information is expressed through use of conventional grammatical codes rather than argument listing (e.g. TRANS for transitive verbs in **PLNLP**, or I for intransitive verbs with no object in **LDOCE**). Thus, the number of arguments is to be recomputed on demand on the basis of the code (although this is not possible in all cases).

#### 3.2.2.2 Argument syntactic category

The syntactic category of arguments is indicated in all lexicons, albeit indirectly in some cases. The set of categories used may vary considerably. Furthermore, differences in surface syntactic realisation of complements may have different consequences on the definition of what constitutes a unique lexical entry. In what follows, these two aspects are considered in more detail for each lexicon.
In **Eurotra**, the syntactic realisation of arguments is dealt with at the level of relational structure. The following phrasal categories may function as verbal complements: ADVP (adverbial phrase), AP (adjective phrase), NP (noun phrase), PP (prepositional phrase) and S (completive sentence — which may be finite or non finite). Different syntactic realisations of the same complement can be collapsed within the same lexical entry as long as their functional role does not vary.
In **Genelex** a complement may be realised either by a terminal or a non-terminal category, and this information is to be encoded obligatorily:

- The set of non terminal categories includes NP (noun phrase), PP (prepositional phrase), AP (adjective phrase), ADVP (adverbial phrase), DETP (determiner phrase), VP (verbal phrase) and S (sentence).
- The set of terminal categories includes noun, adjective, adverb, verb, preposition, conjunction, interjection, determiner, pronoun, particle.

Alternative syntactic realisations of the same complement are expressed through a distribution paradigm. To give a concrete example, the distribution paradigm of the direct object complement of the verb *expect* will specify the syntactic categories NP, that–clause and infinitival clause to account for the following examples:

- He expected some benefits.
- He expected to receive some benefits.
- He expected that they will receive some benefits.

Features of different types can be combined with these category specifications to allow for more fine grained restrictions, including lexical, morphological, morphosyntactic, syntactic, syntacticosemantic and semantic ones. For example, the syntactic category S can be further specified as to its type, e.g. completive, indirect–interrogative, or infinitive.
**ILCLEX** specifies the syntactic realisation of verb arguments at the level of subcategorisation frame. Each configuration of different syntactic realisations of arguments gives rise to a different subcategorisation frame. The list of syntactic categories follows: NP, PP, AP, ADVP, IND–THAT–CLAUSE (indicative that–clause), SUBJUNCT–THAT–CLAUSE (subjunctive that–clause), IND–WH–CLAUSE (indicative wh–clause), SUBJUNCT–WH–CLAUSE (subjunctive wh–clause), A–INF (infinitive clause introduced by the

preposition *a*), DI–INF (infinitive clause introduced by the preposition *di*), BARE–INF (infinitive clause introduced by no preposition) and WH–INF (infinitive clause introduced by a wh–element). The reader is reminded that different subcategorisation frames are subsumed under the same pattern, i.e. verb entry.
A different approach is presented in **Comlex**, where syntactic category information, which is encoded at the level of constituent structure, is criterial for the definition of independent lexical entries. For example, two different constructions of the verb *help* — as in *he helps to hang the laundry* and *he helps hang the laundry*, respectively with and without use of *to* before the verbal complement — are grouped in two different verb classes (see section 3.2.2.7 for a definition of frame group in Comlex), and, consequently, encoded as two separate entries.The list of syntactic categories used in Comlex follows: NP, PP, POSS (for possessive NP), PREP (for preposition), PART (for adverbial particle), ADJP, ADVP, VP and S.
In **Acquilex**, which follows the Unification Categorial Grammar model, each sign (subcategorised signs included) is specified for its syntactic category, which can be either basic or complex. Basic categories are binary feature structures consisting of a category type, and a series of attribute value pairs encoding morphosyntactic information:

```
[CAT-TYPE: cat-type
  M-FEATS: m-feats]
```

Three basic category types are used: ‘n’ (noun), ‘np’ (noun phrase) and ‘sent’ (sentence). Morphosyntactic features are included only where needed. Complex categories are recursively defined by letting the type ‘cat’ instantiate a feature structure with attributes RESult, DIRection and ACTive. RESult can take as value either a basic or complex category, ACTive is of type ‘sign’, and the direction attribute encodes order of combination relative to the active part of the sign (e.g. forward or backward):

```
[RES: cat
 DIR: dir
 ACT: sign]
```

In **LDOCE**, the grammar codes expressing the verb subcategorisation information implicitly include specifications on the syntactic realisation of its complements which can easily be derived by automatic procedures. See the contrast, for instance, between the code T1, used to mark transitive verbs with an NP object, and the code T3 used for transitive verbs with an infinitival clause as object.
**PLNLP** conveys information on the syntactic category of complements either through use of grammar codes which are assumed to be interpreted by default (e.g. transitive verbs are normally expected to take two NP arguments) or by explicit encoding of category information when the syntactic realisation of complements deviates from what is assumed by default (e.g. with transitive verbs taking an infinitival clause as object, the relevant subcategorisation features are TRANS and INFPCOMP).

#### 3.2.2.3 Argument functional role

Argument functional roles (or grammatical functions) are always indicated either explicitly (Genelex, Comlex, Eurotra, ILCLEX) or implicitly (PLNLP, LDOCE, Acquilex).
In **Genelex**, functional roles are assigned with reference to subcategorised positions (complements). Genelex suggests a list of basic functions, which may nonetheless be modified or integrated by the lexicographer.
In **Comlex**, functional roles are specified at the level of grammatical structure. The linking to the corresponding constituent structure is expressed by means of coindexation. Available grammatical relations are: Subject, Obj, Obj2, Comp, Prep (for bare prepositions) and Part (for bare particles). Note that the subject role is always indicated at the level of grammatical structure; this is in contrast with the fact (already mentioned above) that the subject constituent is given no explicit mention in the constituent structure unless it is morphosyntactically constrained. It is worth noting here that, unlike Genelex, Comlex grammatical functions may vary depending on the syntactic realisation of the complements (e.g. NP objects and clausal objects are not assigned the same functional label, as sentential objects are assigned a distinct role, i.e. Comp).
**Eurotra** treats information about the grammatical function of complements at the level of the relational structure. Nine grammatical functions are defined at this level: Subject, Direct Object, Indirect Object, By-agent

(in passive constructions), Attribute to subject (with attributive verbs), Attribute to object (with transitive-attributive verbs), Strongly bound PP, Weakly bound PP and Sentential complement.
In **ILCLEX**, functional roles are encoded at the pattern level. The following set of grammatical functions is provided: subject, object, predicative-subject, predicative-object, dative, concern (covering different instances of the genitive case), locative (locative complements), mod (adverbs or adverbial locutions) and lastly instrument (instrumental complement).
As for **LDOCE** and **PLNLP**, functional role information, like information about syntactic category, is to be inferred from verb grammar codes.
In **Acquilex**, functional roles are not encoded as primitive notions, but are nonetheless indirectly expressed in terms of position in category structure according to the 'obliqueness hierarchy'.

#### 3.2.2.4 Control and raising phenomena

This information is always expressed in all lexicons, although some of them make no explicit distinction between control and raising. **Genelex**, **Acquilex** and **PLNLP** use the term control with reference to both equi and raising verbs, e.g. *John wants to leave* and *John seems to be happy*. No special category types expressing the equi-raising distinction are deemed to be necessary, since the contrast is located only at the semantic level, where Acquilex makes provision for a fine grained set of verb semantic classes (i.e. the equi verb class is expressed as the class of verbs where the controller NP is assigned a thematic role, whereas the raising verb class is defined as the class of verbs in which the controller NP is non thematic). By contrast, **Comlex**, **Eurotra**, **ILCLEX** and **LDOCE** encode control and raising as two separate phenomena.
Regardless of whether the equi/raising distinction is made or not, control information is encoded in two different ways:

1. Through coindexation of the controller NP with (the subject of) the infinitival clause (**Genelex** and **Acquilex**);
2. By means of feature specification (**PLNLP**, **ILCLEX**, **LDOCE** and **Comlex**).

In **Eurotra**, information about the control properties of verbs appears encoded at the relational level dictionary by means of a feature specifying the syntactic function of the controller element: subject, direct object or indirect object. The coindexation between this element and the empty subject of the infinitival clause is then performed by grammar rules.

#### 3.2.2.5 Lexical selection

By lexical selection we mean, for present purposes, the specific lexical requirements that a verb imposes on its subcategorised context, e.g. the selection of a particular preposition introducing a given complement. All lexicons considered here express, in some way or another, this type of information, with varying degrees of granularity. Typical instances of lexical selection are: bound prepositions, particles, complementisers, impersonal subjects and clitics. Bound prepositions and particles are specified in all lexicons through listing and/or by indication (when possible) of a relevant class of prepositions or particles (as in the case of locative prepositions). For those lexicons whose information is articulated over different levels of linguistic description, lexical selection is specified at the syntactic level: the level of constituent structure for **Comlex** and **Eurotra**, in the CAT attribute of the verb sign in **Acquilex**, at the subcategorisation level within **ILCLEX.** Complementisers are explicitly listed in Comlex and Genelex, while in the others they are referred to indirectly through use of argument class (e.g. WH–complements and THAT–complements in **PLNLP** and ILCLEX). Impersonal subjects are registered through direct reference in Comlex (at the level of constituent structure, where, as already pointed out above, lexically constrained subjects, such as *it* subjects, are obligatorily specified) and Genelex. Again, other lexicons rather make use of verb class codes (e.g. the label IMPERSONAL used in PLNLP refers to verbs which can only occur in an impersonal construction). Genelex deals with clitics as a particular case of lexical selection. A language-specific case of lexical selection is auxiliary selection in those languages where this information is subject to lexically-governed variation (e.g. Italian and French).

#### 3.2.2.6 Morphosyntactic constraints

Under the heading of morphosyntactic constraints a wide variety of phenomena, both general and language specific, is gathered. In what follows, we provide a (partial) list of the types of constraints encoded, with an indication of the lexicon which explicitly deals with them. Note that, given their rather heterogeneous nature, it is difficult to state with certainty whether they are taken into account in each lexicon and by what formal means. It may well be that this information is presupposed by a specific verb class (e.g. the feature passivisability is in some cases implied by the class of transitive verbs), in which case it is not mentioned in this section.

**Passivisability** — This is treated as a morphosyntactic constraint on the verb entry within **Comlex** and **Eurotra** (where it is dealt with at the constituent structure level). In other lexicons, passivisability is rather classified as a case of frame alternation (see the relevant section below).

**Morphosyntactic restrictions on the arguments** — These restrictions refer to morphosyntactic properties (e.g. number or gender) that arguments have to exhibit when co-occurring with a given verb. For instance, plural or singular only subjects are specified in most of the lexicons taken into account. In this respect, the reader is reminded that in Comlex, where the subject constituent is usually omitted at the level of constituent structure, the subject is nonetheless overtly expressed therein when and only when it is morphosyntactically constrained. In this case, constraints are specified immediately to its right in the constituent list.

**Other kinds of morphosyntactic restriction** — Constraints are introduced for encoding verbs which occur with a given complement structure provided that certain morphosyntactic conditions are met. The argument structure of these verbs is qualified through specification of some additional parameters: e.g. the fact that one can say *she didn't realise whether…* but cannot say **she realised whether…* is captured in **Comlex** through the condition 'neg t' (i.e. 'negative = true').

**Language-specific morphosyntactic constraints on the verb only** — For Italian and Spanish there are verbs which in certain readings, most frequently intransitive ones but not only, take an obligatory clitic pronoun (marking the pronominal form). This is a lexically-governed property which must be specified in any Spanish or Italian lexicon; **ILCLEX** (at the pattern level), **PLNLP** and **Eurotra** specify it.

#### 3.2.2.7 Frame alternation and argument optionality

By frame alternation, we mean, in this context, the possibility for a given verb to appear in at least two different subcategorisation fames which can be related to each other through a limited set of mapping relations. Well-known instances of frame alternation are the causative/inchoative alternation (as between *Kim broke the glass* and *The glass broke*), the indefinite object alternation (as in *John ate a sandwich* and *John ate*) and the locative alternation (between — say — *John loaded hay on the cart* and *John loaded the cart with hay*). The lexicons taken into account for our purposes vary considerably in the way this information is encoded and in the range of phenomena treated under this rubric. To give an example, argument optionality is not always dealt with as a case of frame alternation. In **Eurotra**, different subcategorisation frames of the same verb give always rise to distinct lexical entries. On the contrary, many other lexicons express the relation between two (or more) frames explicitly. In **Acquilex**, only one member of the alternation is stored in the lexicon, while the others are derived from the stored entry through application of lexical rules. Constraints on the applicability of lexical rules are enforced through feature specification within the basic lexical entry. A similar approach is available in **Genelex**, where nonetheless a second alternative is also provided: all alternating frames are listed on a par and the links relating one frame to the other are marked. This is also done in **ILCLEX** through pattern rules. In **Comlex**, the notion of frame group is introduced to deal with frame alternation. A frame group is a family of alternating simple frames. For example, ditransitive verbs are assigned the same frame group, which encompasses both 'v np to np' (as in *she gave a kiss to her mother*) and 'v np np' (as in *she gave her mother a kiss*) frames. This leads to a multiplication of verb classes which could in principle be avoided. On the other hand, the Comlex approach does not force the lexicon-writer to choose either frame as

basic. **LDOCE** expresses frame alternation only implicitly through grammar code disjunction. The **PLNLP** approach is similar, with the main difference that verb senses are not distinguished.
Argument optionality is not always interpreted as a case of frame alternation. In fact, **Genelex** conveys information about optional arguments in one single frame, by enclosing them between brackets. The possibility of expressing constraints on the cooccurrence of optional and/or mandatory complements is also provided. In **Comlex** a different approach is taken; for instance an object deleting verb such as *eat* is assigned two separate subcategorisation frames (i.e. 'v np' and 'v') which are in their turn encoded in two distinct and unrelated lexical entries. In other lexicons (e.g. **ILCLEX**) strict obligatoriness of complements, rather than optionality, is marked.
Passivisability is treated as an instance of frame alternation in **Acquilex** and **Genelex**. Yet, this is not always the case: as already pointed out above, passivisability can be seen as a kind of morphosyntactic constraint or as an inherent property of transitive verbs.

#### 3.2.2.8 Θ structure and deep structure

Only two of the lexicons considered in the present survey encode information at this level of linguistic description at the boundary between syntax and semantics: Acquilex and Eurotra. In **Acquilex**, following Dowty, thematic relations are expressed as cluster concepts determined for each choice of predicate through attribution of selected entailments (e.g. volitional involvement, sentient, movement, etc.) which qualify the relative agentive strength and affectedness of event participants. Only three thematic role-like concepts are recognised: proto-agent, proto-patient and prep (the last one being added by Sanfilippo within the Acquilex framework). In **Eurotra**, this information is encoded at the level of interface structure. The argument roles used to describe the deep structure are: external argument, internal argument, second participant, second entity, attribute of the subject, attribute of the object, dative perceiver (with raising verbs) and adjuncts (measure, associated, place, origin, goal).

#### 3.2.2.9 Other properties

So far, we have presented aspects of lexical representation that were shared by more than one lexicon. Some other pieces of lexical information are provided by one lexicon only. To give but one example, **Eurotra** encodes aspectual information as well as selectional restrictions on verb arguments. They are intentionally neglected in this context, as they are not amenable to an inter-lexicon comparison.

### 3.2.3 Summary

Among the lexicons considered, there appears to be substantive agreement as to the range of linguistic phenomena that are taken into account in the lexical entry of a verb (as shown in table 3.3), much less so as to the way these phenomena are to be encoded. In some cases, the relevant information is only indirectly encoded but nonetheless inferrable (as indicated by the label 'infer' below) from other pieces of available information. It is interesting to note, here, that the main area of disagreement revolves around the boundary line between syntax and semantics, as shown in the last two columns of our table, which are the only ones where some minus signs appear (meaning that the phenomenon considered is not accounted for). The fact that the information stored in each lexicon is not uniformly distributed across the various levels of linguistic description has a number of consequences:

1. The lexicons differ greatly in the criterial properties they adopt for the definition of an individual lexical entry: for instance, if, for some lexicons (e.g. Comlex), different syntactic realisations of the same complement give rise to different lexical entries, other lexicons (e.g. Acquilex and LDOCE) collapse different argument structures of a verb in one entry provided that they share a common core meaning.

2. Some lexicons try to explain at one single level what other lexicons tend to distribute over more than one level; a paradigmatic example is the distinction between control and raising verbs, which is located at the semantic level by those lexicons which avail themselves of this level (e.g. Acquilex), while being captured in purely syntactic terms in those lexicons, such as Comlex, which do not resort to semantics;

yet some other lexicons (e.g. PLNLP) neutralise the distinction itself by using a unique category to encompass both cases.

3. As a consequence of both 1 and 2, although there exists a common set of observational linguistic phenomena which all lexicons agree in taking into account, the way these phenomena are related to one another in each lexicon varies considerably.

| Lexicon | arg. arity | syn. cat. | funct. role | control | lexical select. | morphsyn. constr. | frame altern. | deep struct. |
|---|---|---|---|---|---|---|---|---|
| Acquilex | infer | infer | infer | + | + | + | + | + |
| Comlex | infer | + | + | + | + | + | + | – |
| Eurotra | infer | + | + | + | + | + | – | + |
| Genelex | infer | + | + | + | + | + | + | – |
| ILC | + | + | + | + | + | + | + | – |
| LDOCE | infer | infer | infer | + | + | + | infer | – |
| PLNLP | infer | infer | infer | + | + | + | infer | – |

Table 3.3: Lexical entry information for verbs

A final point worth stressing in this context is that some lexicons, due to their theoretical biases and/or their application-driven needs, appear to be rather prescriptive as to the sort of information which should be put in the lexicon: e.g. Eurotra allows a maximum of four arguments in a verb entry, each of which can only be assigned a non-terminal syntactic category. The Genelex attitude in this respect is far more liberal.

## 3.3 Syntactically annotated corpora

In this section, a brief overview will be given of the main annotation schemes used in the syntactic annotation of corpora to date. There are six main schemes in use at the moment, the majority of which are used on English only, although most are now being extended to include the annotation of other European languages. The schemes are the following:

1. Constraint Grammar
2. Lancaster/IBM Scheme
3. Paris/IBM Scheme
4. TOSCA
5. UPenn
6. SUSANNE

All of the above schemes use a traditional phrase structure approach to syntactic analysis, apart from Constraint Grammar, which uses a partial dependency approach. The Constraint Grammar scheme is also the only fully automated system; the other schemes are applied either manually, or in an interactive manner, involving human intervention/correction at different points in the analysis.

### 3.3.1 Constraint Grammar

#### 3.3.1.1 Introduction

The Constraint Grammar (Karlsson, Voutilainen, Heikkilä & Antilla, 1995) Framework has been implemented with Two-level Morphology, to produce a system for syntactic analysis of unrestricted text. The most comprehensive system is ENGCG (for the analysis of written English) but the system is currently being developed for Finnish, Swedish, Danish, German, Basque and French.

The Constraint Grammar Framework differs from the syntactically annotated corpora under study in this section for three reasons:

1. The aim of this work was the production of a system that may be used to annotate corpora, rather than the annotation of a particular corpus.
2. The system produces a shallow surface syntactic analysis, based on dependency-oriented syntax, while other syntactically annotated corpora tend to use a constituent structure analysis.
3. Ambiguities (occurring in 3–7% of all words) are left unresolved.

The analysis is carried out at word level, and all text words receive one or more morphosyntactic analyses, consisting of:

- A base form;
- The morphological analysis (including a part of speech tag, inflection, derivation and subcategorisation; and
- Dependency-oriented syntactic word tags (marked by the '@' sign)

#### 3.3.1.2 Method of annotation

Both the morphological and syntactic analysers use rule-based linguistic descriptions. The system works in the following way:

1. Tokenisation;
2. Lookup of morphological tags;
   (a) Lexical component;
   (b) Guesser;
3. Resolution of morphological ambiguities;
4. Lookup of syntactic tags;
5. Resolution of syntactic ambiguities.

#### 3.3.1.3 Tokenisation

The tokeniser identifies punctuation and multiword units, and splits enclitic forms into grammatical words.

#### 3.3.1.4 Morphological lookup

This process begins with a lexical analysis based on a large lexicon including all inflected and central derived word forms. The lexical analyser assigns all possible morphological analyses to each word that is in the lexicon, and the remaining words are assigned an analysis by means of the *guesser* (a heuristic rule-based module). These rules are mainly governed by word shape, and if none of them apply, then a nominal analysis is given.

#### 3.3.1.5 Resolution of morphological ambiguities

The rule-based Constraint Grammar parser is used to resolve some of the ambiguities at this stage. The constraints are partial paraphrases of form definitions of syntactic constructs such as the noun phrase. The English grammar for example, contains about 1,200 grammar-based constraints, plus 200 heuristic constraints.

#### 3.3.1.6 Syntactic lookup

All possible syntactic tags are introduced for each word. This could, in some cases, mean that more than ten alternatives are given for one morphological reading.

#### 3.3.1.7 Resolution of syntactic ambiguities

The parser finally consults a syntactic disambiguation grammar. The English version of the Constraint Grammar contains 800 syntactic constraints, of a similar form to the rules at the morphological resolution stage.

#### 3.3.1.8 Syntactic tags

The English version of the Constraint Grammar marks the syntactic functions shown in table 3.4:

| | |
|---|---|
| @+FAUXV | finite auxiliary verb |
| @−FAUXV | nonfinite auxiliary verb |
| @+FMAINV | finite main verb |
| @−FMAINV | nonfinite main verb |
| @SUBJ | subject |
| @F−SUBJ | formal subject |
| @OBJ | object |
| @I-OBJ | indirect object |
| @PCOMPL-S | subject complement |
| @PCOMPL-O | object complement |
| @APP | apposition |
| @NPHR | stray nominal |
| @N | title |
| @O-ADVL | object adverbial |
| @ADVL | adverbial |
| @DN> | determiner |
| @NN> | premodifying noun |
| @AN> | premodifying adjective |
| @QN> | premodifying quantifier |
| @GN> | premodifying genitive |
| @AD-A> | premodifying ad-adjective |
| @<AD-A | postmodifying ad-adjective |
| @<NOM-FMAINV | postmodifying nonfinite verb |
| @<NOM | other postmodifier |
| @<P-FMAINV | nonfinite verb as complement of preposition |
| @<P | other complement of preposition |
| @CC | coordinator |
| @CS | subordinator |
| @INFMARK | infinitive marker |

Table 3.4: ENGCG tags

#### 3.3.1.9 Example

As mentioned above, the syntactic tags are distinguished by the use of the '@' sign. The analysis is dependency based, but only partially. As can be seen in table 3.5, dependency relations are shown by the use of the left and right angle brackets, showing that a word is dependent on another to either the right of the left. In the example below, *Karlsson* is marked as '@<P' meaning that it is the complement of a preposition to be found previous to *Karlsson* (in this case the preposition *by*).

```
”<*i>”
      ”i” <*> <NonMod> PRON PERS NOM SG1 SUBJ @SUBJ
”<started>”
      ”start” <SV> <SVO> <P/on> V PAST VFIN @+FMAINV
”<work>”
      ”work” N NOM SG @OBJ
”<on>”
      ”on” PREP @ADVL
”<an>”
      ”an” <Indef> DET CENTRAL ART SG @DN>
”<*english>”
      ”english” <*> <Nominal> A ABS @AN>
”<description>”
      ”description” N NOM SG @<P
”<within>”
      ”within” PREP @<NOM @ADVL
”<the>”
      ”the” <Def> DET CENTRAL ART SG/PL @DN>
”<*constraint>”
      ”constraint” <*> N NOM SG @NN>
”<*grammar>”
      ”grammar” <*> N NOM SG @NN>
”<framework>”
      ”framework” N NOM SG @<P
”<proposed>”
      ”propose” <Vcog> <SVO> <SV> PCP2 @<NOM-FMAINV
”<by>”
      ”by” PREP @ADVL
”<*karlsson>”
      ”karlsson” <*> <Proper> N NOM SG @<P
”<$[>”
”<1990>”
      ”1990” <1900> NUM CARD @ADVL
”<$;>”
”<1994a>”
      ”1994a” <1994a> NUM CARD @ADVL
```

Table 3.5: ENCG output

### 3.3.2 Lancaster/IBM Treebank and the IBM Paris Treebank

This section will discuss the IBM Paris Treebank, and the various Lancaster schemes. Although many changes have been made throughout the various incarnations of the Lancaster syntactic annotation schemes, for our purposes they may be considered together. The Paris scheme (for French) was intended as a parallel scheme to the Lancaster English annotation scheme, and can therefore also be discussed with the various Lancaster schemes.

#### 3.3.2.1 Introduction

Both these schemes use a constituent structure analysis called *skeleton parsing*, which is a limited form of parsing, that indicates the structure of a sentence in terms of major constituents, i.e. sentence types, predicates, clauses and major phrase types. The constituents are marked using square brackets, which correspond to the phrase structure markers of tree diagrams. The schemes were developed with two main considerations in mind:

1. The scheme should be as uncontroversial as possible from the linguistic point of view, and unlikely to be affected by theoretical difference, thus producing a consistent annotation.

2. The scheme should be simple and easy to learn.

#### 3.3.2.2 Method of annotation

The corpora were annotated manually by a team of grammarians, using a screen editor program, which validates the parse only to the extent of checking for unbalanced brackets.

#### 3.3.2.3 Syntactic tags

The labelled constituents of the UCREL Skeleton Parsing System are shown in table 3.6:

| | |
|---|---|
| Fa | Adverbial clause |
| Fc | Comparative clause |
| Fn | Noun clause |
| Fr | Relative clause |
| G | Genitive |
| J | Adjective phrase |
| N | Noun phrase |
| Nn | Metalinguistic constituent |
| Nr | Temporal adverbial noun phrase |
| Nv | Non-temporal adverbial noun phrase |
| P | Prepositional phrase |
| S | Direct speech |
| Si | Interpolated or appended sentence |
| Tg | *-ing* clause |
| Ti | *to* + infinitive clause |
| Tn | Past participle clause |
| V | Verb phrase |
| & | First conjunct |
| + | Second, (etc.) conjunct |
| @ | Discontinuity marker |

Table 3.6: UCREL Skeletal Parsing System tags

Unlabelled brackets may also be used in cases where a constituent is not one of the types for which a label is authorised, and the grammarian is convinced that the sequence of words is a constituent.
Most of the abovementioned labels represent categories that are easily identifiable and non-controversial. The tag 'Nn' for a metalinguistic constituent was introduced in order to deal with the problem of computerese verbal-name constructions, often found in the IBM manuals being processed, e.g.

```
the [Nn enter Nn] key
```

Similar expressions will occur in other genres, and this constituent was therefore included in the general scheme, for example:

```
a [Nn rent a bike Nn] scheme
```

The French scheme developed at IBM Paris is almost identical to the above shown scheme, but the following labels were added:

- Attributive adjective phrase
- Adverbial prepositional phrase
- Infinitive verb phrase

B01000 1 v
[N The_AT world_NN1 N][V owes_VVZ [N a_AT1
considerable_JJ debt_NN1 [P of_IO [N gratitude_NN1
N]P]N][P[P&g to_II [N Mr._NNSB1 Byles_NP1 [N
the_AT Butcher_NN1 N]N]P&g] ,_, or_CC [P+ to_II [N
the_AT [ less_RGR generic_JJ ] alias_NN1 [Fr[P
under_II [N which_DDQ N]P][N this_DD1 indispensable_JJ
tradesman_NN1 N][V supplied_VVD [N the_AT needs_NN2
[P of_IO [N the_AT Stevenson_NP1 family_NN1 N]P]N]
[P in_II [N Bournemouth_NP1 N]P][P in_II [N 1885_MC
N]P]V]Fr]N]P+]P] ,_, [Fa since_ICS [N Stevenson_NP1 N]
[V explained_VVD [Fn that_CST [[N it_PPH1 N][V was_VBDZ
[N the_AT bills_NN2 [P of_IO [N Mr._NNSB1 Byles_NP1 N]
P]N]V]][[N which_DDQ N][V drove_VVD [N him_PPHO1 N]
[P to_II [N the_AT writing_NN1 [P of_IO [N The_AT
Strange_JJ Case_NN1 [P of_IO [N[N&g Dr._NNSB1 Jekyll_NP1
N&g] and_CC [N+ Mr._NNSB1 Hyde_NP1 ._.
N+]N]P]N]P]N]P]V]]Fn]V]Fa]V]

Table 3.7: APHB output

#### 3.3.2.4 Examples

A partially parsed sentence from the American Printing House for the Blind corpus is shown in table 3.7: In this example, there is a cleft sentence within an adverbial clause, and this is marked by unlabelled brackets. The scheme does not make any allowances for marking any more detail that that which is shown in the surface structure. This is only to be expected with the theory-neutral approach, and intention to be uncontroversial, of the scheme designers.
The following example shows a parsed sentence from the IBM Computer Manual Corpus:

```
M1154602 v
[N Files_NN2 N][V[V& come_VV0 [P into_II
[N the_AT print_NN1 queue_NN1 N]P]V&]
and_CC [V+ either_LE [V&[V& match_VV0
[N[G a_AT1 printer_NN1 's_$ G] setup_NN1
N]V&] (_( [V+ get_VV0 [Tn printed_VVN
Tn]V+] )_) V&] or_CC [V+[V& do_VD0
not_XX match_VVI V&] (_( [V+ wait_VV0
V+] )_) V+] V+]V] ._.
```

In this short sentence, complex verbal coordination is shown, exhibiting
the use of the '&' and '+' signs for coordination.
Below are two example parsed sentences from the French corpus, the second of which illustrates the labelling of coordinated sentences, which follows the Lancaster approach for English:

```
[N Vous_PPSA5MS N] [V accedez_VINIP5 [P a_PREPA [N
cette_DDEMFS session_NCOFS N] P] [Pv a_PREP31 partir_PREP32
de_PREP33 [N la_DARDFS fenetre_NCOFS [A Gestionnaire_AJQFS [P
de_PREPD [N taches_NCOFP N] P] A] N] Pv] V] ._.
```

```
[Z [Z& [N L' industrie N] [V n'EG a pas reussi [P [Vi etablir [N des
prix [A satisfaisants A] N] [Pv pour [N ces grains N] Pv] Vi] P] V] Z&]
 , mais [Z+ [N le ministre N] [N lui N] , [V a reussi V] Z+] Z] .
```

### 3.3.3 TOSCA annotation scheme

TOSCA (Tools for Syntactic Corpus Analysis) is an annotation project developed at the Katholieke Universiteit at Nijmegen, the Netherlands. The main aim of the project is the production of resources for linguistic research in the areas of syntax and language use.

#### 3.3.3.1 Introduction

The TOSCA annotation scheme has been used in the analysis of the Nijmegen corppus, and of the TOSCA corpus, both of which consist of mainly written language. It is also being used for some parts of the ICE corpus, in which spoken language is also included.

#### 3.3.3.2 Method of annotation

The TOSCA annotation scheme is applied by an interactive system between linguist and computer. The computer is used to produce all possible analyses, from which the linguist selects the correct choice. The part of speech tagging and the addition of syntactic labels take place as part of one process, in which the chosen sequence of word class tags is used as input to the parser. The parser is automatically generated from a formal grammar in the AGFL formalism (Affix Grammar over Finite Lattices).

#### 3.3.3.3 Syntactic tags

In the TOSCA annotation scheme, constituents are labelled for their function and category, while additional syntacticosemantic information is contained in various attributes.
The three major units of description are the word, the phrase and the clause/sentence. The non-lexical category labels are shown in table 3.8:

| | |
|---|---|
| AJP | Adjective phrase |
| AVP | Adverb phrase |
| CL | Clause |
| CLOID | Clausoid |
| CONJ | Conjunct |
| COORD | Coordination |
| DAJP | Discontinuous adjective phrase |
| DTP | Determiner phrase |
| NP | Noun phrase |
| NPAP | Appositive noun phrase |
| PP | Prepositional phrase |
| QTAG | Question tag |
| S | Sentence |
| SUBP | Subordinator phrase |
| TXTU | Textual unit |
| VP | Verb phrase |

Table 3.8: TOSCA tags

In addition to these labels, there are over 90 function labels which are added to the annotation, and over 100 attribute labels. The function labels identify such phenomena as:

- Heads and pre/post modifiers within phrases;
- Subject/object/indirect object;
- Notional subject/notional object;
- Raised subject complement/raised direct object;

- Transitive complement/focus complement;
- Subject/object complements;
- Auxiliary verbs, operators, interrogative operators and main verbs;
- Adverbials.

The attribute labels represent the field in which some of the following are included:

- Active/passive voice;
- Transitive/ditransitive/monotransitive/dimonotransitive;
- Ellipsis/enclitic/proclitic;
- Attributive/predicative;
- Indicative/subjunctive/progressive/present/past;
- Comparative/superlative/intensifying;
- Declarative/interrogative/exclamatory/intensive/negative;
- Raised subject complement/raised direct object.

In addition to the above mentioned syntacticosemantic labels, the annotation scheme contains a number of labels used for extra textual material such as speaker changes, headings and pauses.

#### 3.3.3.4 Examples

The following example shows how the simple sentence *He walked in the garden* is analysed with the TOSCA scheme. The lower case tags in brackets are the attribute labels, while the first label in capitals represents the non-lexical category, and the second shows the function label:

```
NOFU, TXTU ()
    UTT, S (act, decl, indic, intr, past, unm)
      SU, NP ()
         NPHD, PN (pers) {He}
      V, VP (act, indic, intr, past)
        MVB, LV (indic, intr, past) {walked}
      A, PP ()
         P, PREP () {in}
         PC, NP ()
            DT, DTP ()
               DTCE, ART (def) {the}
            NPHD, N (com, sing) {garden}
    PUNC, PM (per) {.}
```

A more complex example is shown below:

```
NOFU,TXTU()
 UTT,S(decl,indic,intr,pass,pres,unm)
  SU,NP()
   DT,DTP()
    DTCE,ART(indef) {An}
   NPPR,AJP(attru)
```

```
       AJHD,ADJ(attru) {alternative}
      NPHD,N(com,sing) {pathway}
     NOFU,COORD(decl,indic,intr,pass,pres)
      CJ,CONJ(decl,indic,intr,pass,pres)
       V,VP(indic,intr,mod,pass,pres)
        OP,AUX(indic,mod,pres) {may}
        AVB,AUX(indic,infin,pass) {be}
        MVB,LV(indic,motr,pastp) {deprived}
       A,PP()
        P,PREP() {of}
        PC,NP()
         NPHD,PN(pers,sing) {it}
      COOR,CONJN(coord) {and}
      CJ,CONJ(decl,indic,intr,pass,pres)
       V,VP(indic,intr,mod,pass,pres)
        A,CON() {hence}
        AVB,AUX(indic,infin,pass) {be}
        MVB,LV(indic,motr,pastp) {controlled}
       A,AVP(excl)
        AVHD,ADV(excl) {simply}
       A,PP()
        P,PREP() {by}
        PC,NP()
         NPHD,N(com,sing) {limitation}
         NPPO,PP()
          P,PREP() {of}
          PC,NP()
           NPHD,N(com,sing) {substrate}
     A,CL(act,indic,intens,pres,sub,unm)
      SUB,SUBP()
       SBHD,CIBJN(subord) {when}
      SU,NP()
       NPHD,N(com,plu) {demands}
       NPPO,PP()
        P,PREP() {on}
        PC,NP()
         DT,DTP()
          DTCE,ART(def) {the}
         NPPR,AJP(attru)
          AJHD,ADJ(attru) {main}
         NPDH,N(com,sing) {pathway}
      V,VP(act,indic,intens,pres)
       MVB,LV(indic,intens,pres) {are}
      CS,AJP(prd)
       AJHD,ADJ(prd) {heavy}
    PUNC,PM(per) {.}
```

### 3.3.4 SUSANNE annotation scheme

The SUSANNE Corpus (Surface And Underlying Structural ANalyses of Natural English) is a 130,000 word corpus marked with grammatical tags, surface and logical grammatical annotation. The corpus consists of 64 of the 500 texts in the Brown corpus, and the annotated version is available from the Oxford Text Archive.

#### 3.3.4.1 Introduction

The aim of the project was to provide a publicly available standard for grammatical analysis. The scheme is fully explicit and is described in detail in Sampson (1995).

#### 3.3.4.2 Method of annotation

The corpus was manually annotated by a team of linguists and computer scientists.

#### 3.3.4.3 Syntactic tags

The syntactic labelling is carried out on three levels:

1. Surface grammar;
2. Syntactic function tags;
3. logical (or deep) grammar.

#### 3.3.4.4 Surface grammar

The surface grammar identifies constituents in a clause and assigns labels to these constituents. The scheme used at this level is very similar to that of the Lancaster scheme, although more detail is included. This detail consists mainly of further subcategorisation of phrases, for example a noun phrase may be marked as singular or plural, common or proper. Grammatical function is also included at this level — different tags are used for a noun phrase marked as subject or non-subject.

#### 3.3.4.5 Syntactic function tags

The function tags used in the SUSANNE annotation are shown in table 3.9.

| | |
|---|---|
| *Complements* | |
| :s | logical subject |
| :o | logical direct object |
| :i | logical indirect object |
| :u | prepositional object |
| :e | predicate complement of subject |
| :j | predicate complement of object |
| :a | agent of passive |
| :S | surface but not logical subject |
| :O | surface but not logical object |
| :G | guest constituent |
| *Adjuncts* | |
| :p | place |
| :q | direction |
| :t | time |
| :h | manner or degree |
| :m | modality |
| :c | contingency |
| :r | respect |
| :w | comitative |
| :k | benefactive |
| :b | absolute |
| *Others* | |
| :n | particle of phrasal verb |
| :x | propositional relative clause |
| :z | complement of catenative |

Table 3.9: SUSANNE tags

#### 3.3.4.6 Logical grammar

Ghost nodes (extra nodes dominating no wording) are added to parsetrees to show the logical position of elements that have been moved or deleted in the surface structure. Function tags are added to node labels to mark the logical structure, and indices are also added to mark the relationship between nodes marked grammatically as counterparts, such as a ghost and the corresponding full surface constituent. Ghost nodes compare with traces, as used by the UPenn Treebank. Other grammatical constructions marked include the following:

- Extraposition from VP;
- Equivalent NP–deletion;
- Raising;
- Relative clauses;
- Wh–fronting;
- Whiz–deletion;
- VP–fronting;
- Negative transport.

#### 3.3.4.7 Examples

Table 3.10 provides an example from the SUSANNE corpus (Sampson, 1995) which gives an impression of the various aligned information types that can be given. The columns (i.e. fields) contain the following information:

**Field 1** — Text references;

**Field 2** — Part of speech tags;

**Field 3** — The text words;

**Field 4** — Base form (lemmatised forms of Field 3; e.g. *said* is lemmatised as 'say');

**Field 5** — Syntactic annotation (brackets and labels).

The following example illustrates the use of a ghost node:

```
[Nns:s123 John ] wanted [Ti:o s123 to go ]
```

In this example, ':s' is the Subject function tag, ':o' the Object tag. 'Ti' stands for 'infinitival clause'. The ':o' tag on 'Ti' indicates that the infinitival clause *to go* is the Object of *wanted*. The 's123' ghost node indicates the logical position of the surface Subject *John*. The number '123' is an index to establish the relation between the ghost and its surface realisation.

### 3.3.5 UPenn Treebank

This section gives a short survey of the second phase of syntactic annotation of the UPenn Treebank Project, as a summary of the publications Marcus, Marcinkiewicz & Santorini (1993) and Marcus, Kim, Marcinkiewicz, MacIntyre, Bies, Ferguson, Katz & Schasberger (1994).

| | | | | |
|---|---|---|---|---|
| A01:0010a | YB | <minbrk> | | [Oh.Oh] |
| A01:0010b | AT | The | the | [O[S[Nns:s. |
| A01:0010c | NP1s | Fulton | Fulton | [Nns. |
| A01:0010d | NNL1cb | County | county | .Nns] |
| A01:0010e | JJ | Grand | grand | . |
| A01:0010f | NN1c | Jury | jury | .Nns:s] |
| A01:0010g | VVDv | said | say | [Vd.Vd] |
| A01:0010h | NPD1 | Friday | Friday | [Nns:t.Nns:t] |
| A01:0010i | AT1 | an | an | [Fn:o[Ns:s. |
| A01:0010j | NN1n | investigation | investigation | . |
| A01:0020a | IO | of | of | [Po. |
| A01:0020b | NP1t | Atlanta | Atlanta | [Ns[G[Nns.Nns] |
| A01:0020c | GG | +<apos>s | – | .G] |
| A01:0020d | JJ | recent | recent | . |
| A01:0020e | JJ | primary | primary | . |
| A01:0020f | NN1n | election | election | .Ns]Po]Ns:s] |
| A01:0020g | VVDv | produced | produce | [Vd.Vd] |
| A01:0020h | YIL | <ldquo> | – | . |
| A01:0020i | ATn | +no | no | [Ns:o. |
| A01:0020j | NN1u | evidence | evidence | . |
| A01:0020k | YIR | +<rdquo> | – | . |
| A01:0020m | CST | that | that | [Fn. |
| A01:0030a | DDy | any | any | [Np:s. |
| A01:0030b | NN2 | irregularities | irregularity | .Np:s] |
| A01:0030c | VVDv | took | take | [Vd.Vd] |
| A01:0030d | NNL1c | place | place | [Ns:o.Ns:o]Fn] Ns:o]Fn:o]S] |
| A01:0030e | YF | +. | - | .O] |

Table 3.10: Aligned information from SUSANNE

#### 3.3.5.1 Annotation in phase I

The first phase of the UPenn Treebank project (from November 1989 to December 1992) produced 4.5 million words of text, all tagged for part of speech and $\frac{2}{3}$ skeletally bracketted. The text was automatically tagged and parsed (using the Fidditch partial parser), and then corrected by hand. The syntactic analysis used a modified form of the Lancaster Treebank approach, using a context-free structure.
The main goals of the second phase, which began in 1993, were the following:

**Mark predicate–argument structure** — Most importantly, the notation should allow for *automatic* extraction of predicate argument structure.

**Provide a richer syntactic annotation** — A broader coverage of syntactic phenomena described is aimed at. (See table 3.11 of old and new syntactic labels.

**Describe non-continuous structures/dependencies** — This was formerly made difficult due to the restrictions of a context-free mechanism.

**Mark grammatical functions and semantic relations** — How are subconstituents semantically related to their predicates? At a minimal level, logical subjects and objects are to be marked.

**Provide a *standard objective methodology for parsers*** — Comparison of parsers that rely on deeper (i.e. closer to predicate argument structure) syntactic representation with surface parsers can be facilitated by marking of predicate argument structure.

**Provide a better consistency of analyses** — Correction of some previous inconsistencies in syntactic tagging.

#### 3.3.5.2 Annotation in phase II

The following issues were to be addressed in phase II of the project:

- *Consistent* treatment of related grammatical phenomena.
- *Null elements* and indexing encode underlying structure.
- *Discontinuous elements* are marked by a non-context free mechanism.
- Clearly distinguishable *semantic roles*, i.e. semantic labels identifying role with respect to predicate, are marked.
- Distinction between verb *arguments* and *adjuncts* where distinctions are clear (where they are not, the notational device `-CLR` is used). The difference between argument and adjunct is *not* structurally marked.
- Gapping: recovery of predicate argument structure for whatever structure is *complete* within the sentence.
- Each syntactic bracket has at least one label and up to *four labels* (including numerical indices).

#### 3.3.5.3 Consistency

Example: A predicate is always either

- The lowest VP; *or*
- The phrasal structure immediately under copular *be* .

```
(S (NP-SBJ your safety belt)
   (VP is-PRD your friend))
```

- in 'small clauses' the constituent tagged `-PRD`

```
(S (NP-SBJ I)
   (VP consider
      (S (NP-SBJ Kris)
         (NP-PRD a fool))))
```

#### 3.3.5.4 Null elements

- Two kinds of null elements (cf. 4 types in UPenn 1[7]):
  - '*T*' for WH–movement and topicalisation; and
  - '*' for all other null elements.
- Coindexing provides underlying position .
- WH–prefixed labels (WHNP, WHADVP, WHPP) only when there is WH–movement, not in WH–questions (SBARQ), not in auxiliary inverted structures (SQ).

[7] The four types are: '*' for understood subject of infinitive or imperative; '0' for zero variant of *that* in subordinate clauses; 'T' for Trace, which marks position where a moved wh–constituent is interpreted; and 'NIL' which marks position where a preposition is interpreted in pied-piping contexts.

- Predicate argument structure can be recovered by replacing a null element with coindexed lexical material: `eat(Tim, what)` from
- Phenomena: passives, zero subjects in infinitival constructions, object and subject control, topicalisation.

```
(SBARQ (WHNP-1 What)
       (SQ is
           (NP-SBJ Tim)
           (VP eating
               (NP *T*-1)))
        ?)
```

Some examples are shown below of the use of null elements:

- Infinitive constructions

```
(S (NP-SBJ-1 Chris
   (VP wants
       (S (NP-SBJ *-1)
          (VP to
              (VP throw
                  (NP the ball)))))))
```

- Object control

```
(S (NP-SBJ Ford
   (VP persuaded
       (NP-1 Zaphod
       (S (NP-SBJ *-1)
          (VP to
              (VP run
                  (PP-CLR for
                         (NP president))))))))
```

- Subject control

```
(S (NP-SBJ-1 Zaphod
   (VP promised
       (NP Ford
       (S (NP-SBJ *-1)
          (VP to
              (VP run
                  (PP-CLR for
                          (NP president))))))))
```

- Raising

```
(S (NP-SBJ-3 Everyone)
   (VP seems
      (S (NP-SBJ *-3)
         (VP to
            (VP dislike
                (NP Drew Barrymore))))))
```

#### 3.3.5.5 Discontinous elements

The context free mechanism of Phase I led to the trapping problem when a sentential level adverb is followed by verb complements.
With context free notation, one can do one of the following, with their respective consequences:

- Treat complement right → Adverb trapped in VP

```
(S (NP-SBJ Chris)
   (VP knew
       (SBAR yesterday
          that
            (S (NP-SBJ Terry)
              (VP would
                 (VP catch
                    (NP the ball)))))))
```

- Treat adverb right → Complement must (falsely) attach to S

```
(S (NP-SBJ Chris)
   (VP knew)
    yesterday
   (SBAR that
        (S (NP-SBJ Terry)
           (VP would
              (VP catch
                 (NP the ball))))))
```

In UPenn II, discontinuous elements are called *pseudoattached*:

- *ICH* Interpret constituent here (simple extraposition)
- *PPA* Permanent predictable ambiguity
- *RNR* Right node raising
- *EXP* Expletive

#### 3.3.5.6 Simple extraposition: *ICH*

```
(S (NP-SBJ Chris)
   (VP knew
       (SBAR *ICH*-1)
       (NP-TMP yesterday)
       (SBAR-1 that
              (S (NP-SBJ Terry)
                 (VP would
                    (VP catch
                       (NP the ball)))))))
```

#### 3.3.5.7 True ambiguity: *PPA*

Even given context, the ambiguity cannot be resolved for human annotators.

- Attach at most likely site (or at higher site);
- Mark other possible sites with *PPA* element.

```
(S (NP-SBJ I
      (VP saw
          (NP (NP the man)
              (PP *PPA*-1)
          (PP-CLR-1 with
                    (NP the
                        telescope))))))
```

#### 3.3.5.8 Right node raising: *RNR*

The same constituent appears to have been shifted out of both conjuncts.

```
(S But
   (NP-SBJ-2 our outlook)
   (VP  (VP has
             (VP been
                 (ADJP *RNR*-1)))
         ,
         and
         (VP continues
              (S (NP-SBJ *-2)
                 (VP to
                     (VP be
                         (ADJP *RNR*-1)))))
         ,
         (ADJP-1 defensive)))
```

#### 3.3.5.9 Extraposed sentences with 'it'

- *It* is recognised as surface subject;
- Extraposed clause is attached at S level with *EXP* attach;
- Extraposed clause is interpreted as a subject of *a pleasure* (*it* ignored for argument interpretation).

```
(S (NP-SBJ (NP It)
           (S *EXP*-1))
   (VP is
       (NP a pleasure))
   (S-1 (NP-SBJ *)
        (VP to
            (VP teach
                (NP her)))))

pleasure(teach(*someone*, her))
```

#### 3.3.5.10 Semantic roles/grammatical functions

As it is very difficult to determine a set of underlying semantic roles, the UPenn II Project restricted itself to the *clearly distinguishable semantic roles* listed below. The given list mostly relates to *adjuncts*.

#### 3.3.5.10.1 Semantic roles

- **-VOC** vocatives
- **-DIR** direction, trajectory
- **-LOC** location
- **-MNR** manner
- **-PRP** purpose
- **-TMP** temporal phrases
- **-BNF** benefactive
- **-EXT** extent

#### 3.3.5.10.2 Grammatical Functions

- **-SBJ** surface subject
- **-LGS** logical subjects in passives
- **—** obvious NP and S objects
- **-ADV** clausal and NP adverbials
- **-DTV** dative object (unshifted form)
- **-NOM** non-NPs acting as NPs
- **-PRD** non VP predicates
- **-CLF** true clefts
- **-TPC** topicalised and fronted constituents
- **-PUT** locative complement of *put*
- **-CLR** special class 'closely related'

The special class 'closely related' is used for constructions occupying middle ground between arguments and adjuncts. These constructions correspond to Quirk *et al.*'s class of *predication adjuncts*, some *phrasal verbs* (Quirk, Greenbaum, Leech & Svartvik, 1985). Usage:

- On S or SBAR: semicomplement clauses with *use*, *design*, *hire*, *build*, *cost*, *take* and *it*–extraposition.

```
(S (NP-SBJ the process)
   (VP will
       (VP take
           (NP (QP as many as six)
                months)
           (S-CLR (NP-SUBJ *)
                  (VP to
                      (VP complete))))))
```

- On PP, ADVP, SBAR-PRP: categories normally interpreted as adverbials (adjuncts), -CLR indicates a closer relationship.

```
(VP associate
    (NP snow)
    (PP-CLR with
          (NP winter)))
```

- On NPs: fixed phrases like *take place*.

```
(VP taking
    (NP-CLR care)
    (PP-CLR of
        (NP the problem)))
```

Examples are given below for the tags **-DTV**, **-PRD**, **-TPC**, **-CLF** and **-PRP**:

#### 3.3.5.10.2.1 -DTV

```
(S (NP-SBJ Aristotle)
   (VP gave
       (NP the book)
       (PP-DTV to
               (NP Plato))))
```

```
(S (NP-SBJ Aristotle)
   (VP gave
       (NP Plato)
       (NP the book)))
```

#### 3.3.5.10.2.2 -PRD

Non-VP predicates

```
(SQ Was
    (NP-SBJ he)
    (ADVP-TMP ever)
    (ADJP-PRD successful)
    ?)
```

```
(SINV and
     (ADVP-PRD-TPC-1 so)
     (VP did
         (ADVP-PRD *T*-1))
     (NP-SBJ the hippopotamuses))
```

#### 3.3.5.10.2.3 -TPC

```
(S (PP-TPC-12 Of
        (NP (NP the 500 barbers)
                (PP-LOC in
                    (NP Philadelphia))))
   ,
   (NP-SBJ (NP (QP only 10))
```

```
           (PP *T*-12))
    (VP know
        (SBAR (WHNP-13 what)
              (S (NP-SBJ they)
                 (VP are
                     (VP doing
                         (NP *T*-13)))))))
```

#### 3.3.5.10.2.4 -CLF

```
(S-CLF (PP-TMP In
               (NP the past))
       ,
       (NP-SBJ it)
       (VP has
           (VP been
               (NP-PRD-2 the
                         husband)
                   (SBAR (WHNP-1 who)
                         (S (NP-SBJ *T*-1)
                            (VP has
                               (VP been
                                  ADJP-PRD-3
                                    dominant)))))))
```

#### 3.3.5.10.2.5 -PRP

```
(S (NP-SBJ-1 (NP activity)
             (PP-LOC at
                   (NP (NP a number)
                       (PP of
                           (NP brokerage
                               houses)))))
    (VP was
        (VP curtailed
            (NP *-1)
            (PP-PRP as
               (NP (NP a result)
                   (PP of
                     (NP the earthquake))))))
```

### 3.3.5.11 Gapping

- The extraction of predicate-argument structure for gapped sentences is possible if the full *parallel* structure is within the same clause.
- The parallel structure is then used as a template: arguments (e.g. NP-SBJ=1) are *mapped* from gapped clause onto the template (e.g. NP-SBJ-1).
- Constituent NP=1 should be mapped over NP-1 in the *largest conjoined structure* that NP-1 and NP=1 both occur in.

```
(S (S (NP-SBJ-1 Mary)
      (VP likes
```

```
          (NP-2 Bach)))
   and
   (S (NP-SBJ=1 Susan)
      ,
      (NP=2 Beethoven)))

like(Mary, Bach)
like(Susan, Beethoven)

(S (S (NP-SBJ I)
      (VP eat
          (NP-1 breakfast
          (PP-TMP-2 in
               (NP the morning))))
   and
   (S (NP=1 lunch)
      (PP-TMP=2 in
            (NP the afternoon)))))
```

However, there is no recovery of structure *outside* the single sentence concerned.

```
Who threw the ball?

(FRAG (NP Chris)
      ,
      (NP-TMP yesterday))

What is Tim eating?

(FRAG (NP-SBJ Mary Ann)
      (VP thinks
          (SBAR 0
                (FRAG (NP chocolate)))))
```

### 3.3.6 Phrase and clause level labels

New labels (in comparison to Phase I labels) are shown in $\boxed{\text{boxes}}$.

- **ADJP** adjective phrase
- **ADVP** adverb phrase
- $\boxed{\textbf{CONJP}}$ conjunction phrase (*as well as…*)
- $\boxed{\textbf{FRAG}}$ fragment
- $\boxed{\textbf{INTJ}}$ interjection
- $\boxed{\textbf{LST}}$ List marker
- $\boxed{\textbf{NAC}}$ not a constituent
- **NP** noun phrase
- $\boxed{\textbf{NX}}$ in complex NPs: head of NP
- **PP** prepositional phrase

- **PRN** parathetical
- **PRT** particle
- **QP** quantifier phrase
- **RRC** reduced relative clause
- **S** simple declarative clause
- **SBAR** clause introduced by (empty) subordinating conjunction, indirect questions, relative clauses
- **SBARQ** direct question (wh–word or wh–phrase)
- **SINV** inverted declarative sentence
- **SQ** inverted yes/no question
- **UCP** unlike coordinated phrase
- **VP** verb phrase
- **WHADJP** wh–adjective phrase (*how hot*)
- **WHADVP** wh–adverb phrase
- **WHNP** wh–noun phrase
- **WHPP** wh–prepositional phrase
- **X** unknown, uncertain, unbracketable

### 3.3.7 Summary

In table 3.11, the various corpus syntactic annotation schemes are summarised with respect to the type of linguistic information which may be included in the lexical representation of verb entries, and their subcategorisation frame. The table is provided to clearly correlate the markings in an annotated corpus with the information that may be provided in a lexicon, and in doing so may seem to blur the distinction between types of information that are relevant to either a lexicon, or an annotation scheme, which in most cases will be very different. However, the feeding of information between the lexicon and the annotated corpus can be seen as bidirectional — much of the information marked in a corpus will necessarily be included in the corresponding lexicon, but this information is then derivable from the corpus, for enrichment of the same, or another lexicon.

| | ENGCG | IBM | TOSCA | SUSANNE | UPENN |
|---|---|---|---|---|---|
| arg-no | $+$ | infer | infer | infer | infer |
| syn-cat | infer | $+$ | $+$ | $+$ | $+$ |
| funct-role | $+$ | $-$ | $+$ | $+$ | $+$ |
| control | † | $-$ | $+$ | $+$ | $+$ |
| lex-select‡ | $+$ | $+$ | $+$ | $+$ | $+$ |
| deep-str | $-$ | $-$ | $-$ | $+$ | $+$ |
| morphsyn-constr | By the nature of an annotated corpus, this information will be inferable | | | | |
| frame-alt | By the nature of an annotated corpus, this information will be inferable | | | | |

Table 3.11: Subcategorisation information in corpus annotations

†Although grammatical function information is included, control information is not necessarily deriveable since ambiguity may remain in the final analysis.

‡In an annotated corpus such phenomena will be marked either as a (bound) particle, or if not discontinous, then possibly as a multi-word verb form.

# Chapter 4

# Standardising Subcategorisation

## 4.1 Introduction to the concept of frame

In this chapter, we present our results regarding the syntactic description of the subcategorisation of verbs. We note, however, that our results could easily be extended to the syntactic description of other parts of speech. It has to be emphasised that the purpose of our work in dealing with standardisation is the following:

- Identify a core of linguistic substance;
- Propose a (very small) set of primitives;
- Propose a neutral descriptive formalism based on this small set;
- Make it easy in this formalism to encode the basic notions (mandatory);
- Give the possibility in this formalism to coherently encode more refined information.

The different approaches to subcategorisation that have been examined exhibit many common notions, linked together in different combinations but reflecting a common core. The main points where divergences occur concern:

- The definition of the boundary between what has to be described in the syntactic part of the lexicon and what in the semantic part;
- Whether some semantic information has to be part of the description of the syntactic properties of the lexical entries; and, as a consequence,
- Whether the *interface* between syntax and semantics is partly included in the syntactic description of the lexicon.

In order to avoid committing ourselves to a specific approach, we opted for a representation model where no particular criteria are adopted to fix the boundaries and the interface between syntax and semantics. The main descriptive core is purely syntactic, but the possibility of encoding semantic information which has a direct bearing on syntactic behaviour is also given. Coherence with existing standardisation work for other aspects of lexical description has been insured by taking into account the results of the EAGLES morphosyntactic group.

### 4.1.1 Frames

Syntactic frames, as descriptions of the subcategorisation structure of a verb, are present in all the approaches to lexical specification which have been examined. Abstracting away from theory- and system-specific coding protocols, we may characterise a frame as a generalisation over different syntactic contexts required by a verb and that are associated with the same syntactic behaviour. In our recommendations, and also to fulfil the requirement of being open, the precise criteria to decide when to generalise into one frame and when to

split information over different frames are not fully given and some choices must still therefore be made by the user when filling out the proposed Frame description.
Frames can be defined for each verb type by providing:

- A list of slots (complements) having phrasal realisations;
- Categorial and morphosyntactic information for the lexical unit;
- Control information for equi and raising verbs;
- Surface order information

A frame represents synthetically a set of possible syntactic structures associated with the verb. In a frame, the combinatorial realisations of slots are supposed to yield possible surface instantiations of the frame. In other words, for some syntactic structure associated with a verb, each slot of the verb frame will either have some realisation, or it will have none if the slot can be optionally left unrealised. Constraints on realisation give criteria to split frames in some cases. Slots are described as independent of each other from the point of view of their realisation, but they are linked with constraints on linear precedence, as well as with control relations.
The synthesis of these different basic notions and their relations is made through the notion of *syntactic frame*, which can be represented as in figure 4.1.
The different descriptive elements that can be shared by other descriptive elements have an attribute identifier (*id*), that is, a unique key, that allows other elements to point to them by means of this id. Some descriptive elements (e.g. *slot*) can be present "0, 1 or *n* times" (conventionally expressed by '*'). Some descriptive elements (e.g. *realisation*) can be present "1 or n times" (conventionally expressed by '+').

### 4.1.2 Slots

Slots provide information about subcategorised elements. Slots are defined by:

- An *index*, giving their canonical order in the frame. A canonical order can be given over slots based, for instance, on a canonical order given over the *function*s. This criterion for ordering is not formally constrained by the frame, but each slot necessarily has an index that gives its range in the frame. This canonical order is not necessarily the same as the linear order, which is constrained by other descriptive elements.
- A Boolean attribute *optionality* that specifies if a slot has to be realised necessarily or not (required information).
- The list of possible realisations described syntactically (required information) and semantically (optional information).

#### 4.1.2.1 Slot realisations

Slot realisations are characterised syntactically and semantically by:

- Category label (required information);
- Morphosyntactic information when necessary (optional information) ;
- Function (required information);
- Semantic class restrictions (optional information);
- Thematic role (optional information).

A slot realisation carries also some information concerning the relation of the realisation with the other slots of the frame or to the lexical unit itself:

FRAME:

```
id        unique_key

Slot*     id              unique_key
          index           number
          optionality     Yes|No
          Realisation+
                  id                  unique_key
                  controlled_by       number*
                  obviates            number*
                  in_order            pre/post

                  syntax              category
                                          label  np | vp | pp | ...
                                          features [feature_i value_i]*
                                      function
                                          name   name_i
                                          subject   yes | no
                                          predicative   yes | no

                  semantics           themat_rol
                                          rol_name   name_i
                                          role_features [(feat_i:val_i)]*
                                      semant_class
                                          class_name   name_j
                                          is_a class_name_l*

self      [features  [feature_i value_i]*  ]
control   control-verb | subject-raising-verb | object-raising-verb

rel_order*    before_slot           number
              after_slot            number
              before_realisation    id
              after_realisation     id
```

Figure 4.1: Representation of FRAME

- *controlled_by* to point to the slot(s) that control(s) the realisation (required information when relevant);
- *obviates* to point to the slot(s) that cannot be coreferent with the realisation (optional information);

- *in_order* to specify if the usual linear order of the realisation is before or after the verb that subcategorises it.

#### 4.1.2.2 Alternative realisations

Alternative slot realisations are represented as a disjunction of slot realisations associated with one slot. This makes it possible to represent with a single frame uses of the same verb which differ with respect to the realisation of a single complement. The next two paragraphs illustrate this for objective and predicative complements.

##### 4.1.2.2.1 Objective complements and their different realisations: NP, that–clause, infinitival

(46) He *expected* some profits — NP

(47) He *expected* to make some profits — Infinitival

(48) He *expected* that they will make some profits — That–clause

##### 4.1.2.2.2 Predicative complements and their different realisations: AP, PP, NP, VP

(49) He *seemed* happy — AP

(50) He *seemed* the happiest person in the world — NP

(51) He *seemed* in good shape — PP

(52) He *seemed* to be happy — VP

#### 4.1.2.3 Optionality

Some verbs do not allow any of their complements to be omitted, whereas others do, e.g.

(53) Eve *ate* (an apple)

(54) Marie *arrive* à partir
Marie succeeds to leave
'Marie managed to leave'

(55) Marie *arrive* (à Paris)
Marie arrives in Paris
'Marie is arriving (in Paris)'

In those cases where one or more complements can be omitted, the possibility of indicating optionality at the slot level allows us to represent, with one frame, uses of the same verb varying minimally in syntactic valency. Optionality is a property of a slot in a frame: there is no restriction on its meaning, and, in particular, if some optionality could be constrained by some realisation of some other slot of the frame, then this is a criterion for splitting into two (or more) different frames.

#### 4.1.2.4 Categorial and morphosyntactic information

The syntagmatic label information is the main information that characterises the syntactic property of a slot realisation. A list of labels for constituents is offered (as part of the primitive set the description is based on), and more refined information on distribution can be expressed by adding constraints expressed through valued features.

#### 4.1.2.5 Grammatical relations or functions

Function is one of the mandatory syntactic characteristics of slot realisation. Function expresses the syntactic relation linking the realisation to the verb that subcategorises for it. Function is characterised by

- *name* (e.g. subject, object, etc.): possible names are not given as primitives.
- *subject*: Boolean
- *predicative*: Boolean

#### 4.1.2.6 Semantic class

Semantic class is one of the non mandatory semantic characteristics of slot realisation. The semantic class expresses a restriction on the semantic interpretation of the slot realisation as belonging to a certain semantic class and semantically constrains the set of possible realisations. Semantic class is characterised by

- *name* (e.g. volitive animate, human): possible names are not given as primitives.
- *is_a_class*: hierarchical link to another semantic class; simple or multiple hierarchies can be described through this link.

#### 4.1.2.7 Thematic role

Thematic role is one of the non mandatory semantic characteristics of slot realisation. The thematic role expresses the role of the slot realisation in the set of semantic participants of the situation or event associated with the frame. Thematic role is characterised by

- *name*: this can be any string or an integer.
- *role_features*: the set of elementary characteristics of the thematic role, e.g. *cause*, *change*, etc.(see section 4.8).

### 4.1.3 Self

Here we deal with morphosyntactic constraints concerning the lexical unit being described (e.g. auxiliary selection).

### 4.1.4 Control

Control information deals with relations between two slots, one being the controller and the other the controlled. There are different types of control: subject or object control and raising. This is a property associated with the frame; more information on control is associated with controlled slot realisation.

### 4.1.5 Rel_order

The linear order information is supposed to be dealt with by external knowledge of the language (grammar) insofar as non lexically-driven order is concerned. However, some lexical items specially constrain the order, and this information has to be present within the lexical description. The constraints on surface linearity are handled at the frame level. As in GPSG, the inventory of slots (*rewriting elements* in GPSG) is kept separated from the linear precedence of slots or slot realisations. A frame provides a list of slots whose rank in the list is *conventional* (canonical or arbitrary). The numbering will not be affected by phenomena such as:

- Pronominalisation;
- Optional slots actually missing;
- Stylistic inversions.

(56) Adam *donna* une pomme à Eve
Adam gave an apple to Eve
'Adam gave an apple to Eve'

(57) Adam lui *donna* une pomme
Adam to her gave an apple
'Adam gave her an apple' (Cliticisation)

(58) A Eve Adam *donna* une pomme
To Eve Adam gave an apple
'It was to Eve that Adam gave an apple' (Stylistic inversion)

(59) Adam *donna* une pomme
Adam gave an apple
'Adam gave an apple' (Missing optional complement)

Constraints on linear precedence are formalised at the level of the frame.

### 4.1.6 Frame set

The same lexical morphological unit may exhibit different subcategorisation patterns. Consider the verb 'expect' for instance; it has at least four subcategorisation patterns corresponding to two different meanings:

- In the meaning of 'wait for'

  (60) Mary *expected* [a friend/to welcome a friend/that a friend came]

  (61) A friend *is expected* by Mary

- In the meaning of 'to assume' or 'to suppose'

  (62) Friendship/to welcome friends/being a welcoming person *is expected* of Mary

  (63) It *is expected* of Mary [to welcome her friends/that Mary welcomes her friends]

The information we need to provide at the syntactic level is the subcategorisation for four kinds of constructions:

- Example (60) active form, auxiliary *have*, one obligatory complement, direct object NP, infinitival, that–clause
- Example (61) passive form, auxiliary *be*, one optional complement, oblique *by*
- Example (62) active form, auxiliary *be*, one optional complement, oblique *of*
- Example (63) passive impersonal form, auxiliary *be*, two optional complements, oblique *of*, object: infinitival and *that*–clause

To relate different syntactic uses associated with the same deep structure or with the same meaning, frames which describe subcategorisation options for the same deep predicate may be grouped by inclusion in a frame set which explicitly relates them and relates slots or slots realisations of one with slots or slots realisations of another related frame. The result of frame alternation rules can be explicitely described this way, and thus the set of relevant rules for a given verb. Using this descriptive possibility, in examples (60)–(63), we could have two sets of frames:

- The first for the two 'wait for' frames;

- The second for the two ‘suppose’ frames.

This can be represented as follows: A *frame set* is a set of related frames, i.e. different surface alternations explicitly linked. There are no constraints on the number of slots of the frames belonging to the same frame set. All frames do not necessarily have the same number of slots — some slots are related to others but not necessarily all of them (impersonal subject for instance); some realisations of slots can be selected and related to another slot or to other slot realisations. This can be represented as in figure 4.2.

```
Frame_set        [FRAME]*

related_set      first
                     frame_pointer            id
                     slot_index               number
                     slot_realisation_index   id

                 second
                     frame_pointer            id
                     slot_index               number
                     slot_realisation_index   id
```

Figure 4.2: Representation of Frame_set

### 4.1.7 Possible extensions

The notion of frame and frame set has been designed to describe verb subcategorisation. However, these notions as represented are not specific to verbs, and could easily be extended to deal with the subcategorisation of other categories. Some place could be added in the descriptive elements to express the probabilities of some information that cannot be fully described by simple Boolean values (e.g. optionality — if we consider that, in context, most of the subcategorised elements are optional, but some are more optional than others). Preference attributes for some information could also be added.

## 4.2 (Morpho)Syntactic features

### 4.2.1 Categorial realisation of a slot

A slot can be occupied by a terminal or a non-terminal category.

#### 4.2.1.1 Non-terminal categories

##### 4.2.1.1.1 Justification

The easiest way to define subcategorisation is to refer to traditional complements or to logical arguments of a predicate. In both cases, these subcategorised elements have phrasal instantiations, which must be specified in syntactic terms.

##### 4.2.1.1.2 List of non-terminal categories

We set up a list of absolutely necessary and uncontroversial non-terminal categories.
Our list is the following: S, VP, NP, PP, AP, ADVP, DETP and XP.

**S: Sentence** — Example: *We made it*

**VP: Verb Phrase** — No subject given: *to become a president*

**NP: Noun Phrase** — *a very pleasant day for fishing*

**PP: Preposition Phrase** — *for the sake of charity*

**AP: Adjective Phrase** — *very easy to please*

**ADVP: Adverb Phrase** — *quite familiarly*

**DETP: Determiner Phrase** — *almost every*

**XP: Under-specified phrase** — any possible phrase.

We are aware that other categories are used in the linguistic community, but they seem to be specialisations of existing categories, compensating, in some formalisms, for the lack of restricting features.
*Restricting features* are at the user's disposal in EAGLES. The variety of features, feature values and feature combinations provides (in association with a category selected from our list of categories) for a very powerful means of expression. In this way, we ensure that any phrasal restriction can be expressed.
As a consequence, our list of categories remains very short. Sentences, for instance, do not have to be split into that–clause, infinitival, interrogative–clause or whatever. In this way, we avoid explosion in the number of categories, which is otherwise mainly due to the mixing of different kinds of information in the category label.
Another consequence is that our list of categories is completely language independent, whereas morphosyntactic features can be very language dependent.

#### 4.2.1.2 Terminal categories

##### 4.2.1.2.1 Justification

The need for terminal categories is not obvious at first sight. We consider the following cases.

###### 4.2.1.2.1.1 Verb complementation and other complementation

Verb complementation, which is best known, is not concerned with such a specification, except for impersonal subjects. However, the concept of complementation and the need for subcategorisation is not restricted to verbs.
Subcategorisation applies virtually to any entry. As such, major categories like adjective, noun and adverb definitely need subcategorisation specifications.

(64) *A loaf* <u>of</u> bread — N

(65) *Curious* <u>about</u> these things — ADJ

(66) *Plenty* <u>of</u> bread — ADJ

In the long run, minor categories may also be candidates for subcategorisation specifications. For example, some prepositions call for infinitivals, some do not.
We may see 'subcategorisation' as nothing else but 'lexical selection'. However, this latter notion is much larger than 'complementation'. What comes under lexical selection may be:

- An oriented selection:
  - From a governor towards its governed part;
  - From a governed part towards its governor.
- A mutual selection between a governor and its governed part:
  - Verb/subject dependency;
  - Light-verb/noun dependency and any frozen or semi-frozen expression.

#### 4.2.1.2.1.2 Fine grained subcategorisation

Subcategorisation may be more or less fine grained. This requirement shows up with the encoding of large scale lexicons only. Some specifications require one to go down to the terminal level, close to phrase rewriting.
For example, some predicative adjectives take an impersonal subject:

(67) It is *obvious* that this subject must be taken into account

(68) Il est *évident* que nous devons traiter ce sujet

As the needs for such specifications cannot be settled once and for all (too many lexical units to be considered, too many languages to be handled, too many views on lexicon and subcategorisation), allowing for terminal categories in slot definitions covers such specifications.

#### 4.2.1.2.1.3 Lexical unit specification

When describing complementation patterns, restrictions have to be placed not only on subcategorised elements, but also on the subcategorising element. Some restrictions must be added in a syntactic frame on the lexical unit itself: auxiliary, tense, number and so on. These restrictions are treated in a similar way to restrictions on complements, that is to say by restricting features on a category. Therefore, the lexical unit is pointed to by its terminal category:

(69) to *go* down to the cellar (to have gone down, be down) V[Aux:BE]

(70) *Descendre* à la cave (être descendu)

(71) To *go* down the steps (to have gone down) [Aux:HAVE]

(72) *Descendre* les escaliers (avoir descendu)

#### 4.2.1.2.2 List of terminal categories

The set of terminal categories which may be entered in a slot derives directly from the list of categories provided by the Morphosyntax group. This guarantees coherence between the different layers of the lexicon in terms of linguistic approach and formal descriptive language and sets up an interface between layers.
For more information on these categorial labels (definition and usage) please refer to the EAGLES CLWG Morphosyntax document. The list of shared categories is the following: noun, adjective, pronoun, verb, adverb, conjunction, adposition, determiner, article and interjection.

### 4.2.2 Restrictive features on slot realisation

#### 4.2.2.1 Lexicographic needs

The description of the complementation pattern requires specification of the categorial selection for each slot. However, in most cases, bare categories do not supply enough information and categories must thus be completed with restricting features.
The most acknowledged and general needs concern verbal complementation:

- Infinitivals: VP plus mood selection
- That clauses: S plus ‘that’ introducer selection
- PP introduced by an empty bound preposition: PP plus prepositional selection
- Impersonal subjects: PRO plus subcategorisation selection

Other needs arise with the encoding of full scale lexicons and the requirement to deal with forms of complementation other than verb complementation.
To provide for all these needs, we allow categories (terminal or non-terminal) to be completed and specialised with *restricting features* stemming from unification-type formal grammars (HPSG, LFG) and lexicons (GENELEX).

- Infinitivals: VP[Mood:INFINITIVE]
- that clauses: S[Introd:THAT]
- PP introduced by an empty bound preposition: PP[Introd:TO]
- Impersonal subjects: PRO[Type:IMPERSONAL]

Typed feature formalisms were mostly developed for grammars. Our task is to define a typed feature model for a lexicon. The main part of the job consists in listing the relevant features. What kind of restrictions do we need to encode in a lexicon? Which features will be made available to lexicographers?
Three kinds of features must be clearly distinguished:

1. Morphosyntactic features;
2. Lexical features; and
3. ‘True’ subcategorisation features.

The first two come from the terminal level. The third concerns the phrasal level only.

#### 4.2.2.2 Morphosyntactic features

There is clear evidence for morphosyntactic restrictions at the phrasal level: infinitivals, impersonal subjects, third person, instrumental case, adverbs of degree, etc.
Morphosyntactic features may be split into:

- Pure morphological information: Finite, Mood, Tense, Person, Number Gender, Possessor, Case, ….
- Terminal syntactic information: Type, Wh-Type, Reflexive, MainFunc, AuxForm, Aux, Degree, Aspect, Voice, ….

The EAGLES CLWG Morphosyntax group has already defined, the set of relevant attributes (features) for each category. We decided to rely on these features and to take them as a starting point, an input and a checklist for our phrase restrictions for different reasons:

- The coherence requirement was to be satisfied among the differents layers of the lexicon.

- The establishing of morphosyntactic features is a long and difficult task. It requires the definition of different types of features — all types do not apply to all categories and must be restricted to some of them; some values (and even types) do not apply to all languages but only to some of them, etc. Since this job was already done in the morphosyntax work, we did not want to lose the benefit of it nor do the job twice.

  One of the main advantages of the representation chosen by the morphosyntax group is to take language specificity into account without providing an over-generating union of all possible and 'blind-to-language' valued features.

  The features are articulated on different levels, corresponding to different degrees of obligatoriness:

  **Level 1** — Presents the grammatical features, such as Gender, Number, Person, etc., which are *recommended features* constituting the minimal core set of features.

  **Level 2** — Contains information which is pertinent to all or many languages (more than 3) and is considered to be useful and easy to standardise but not yet usually encoded by current practices or not purely morphosyntactic: they are to be considered as *optional features*.

  **Level 3** — Presents language specific features, not belonging to the set of recognised common features. The values presented in this row are labelled with the initials of the language they are pertinent to.

#### 4.2.2.2.1 Features on terminal categories

Morphosyntactic features to be added on a terminal category realising a slot are directly inherited from the morphosyntax specifications.

**Subject slot** — PRO[Type:IMPERSONAL] *it* **rains**

**Direct object slot** — PRO[Case:dat] *I* **told him** *some story*

#### 4.2.2.2.2 Features on non-terminal categories

A phrase inherits (in linguistic terms, not in unification terms) from the characteristics of its head and bounded elements. In other terms, some of the characteristics of a phrase are linked to the characteristics of its components.
Before entering into details, the reader should be warned about our goal. It is not the point here to decide which category *actually* instantiates such and such a head (or specifier and so on). This issue is highly controversial and remains open. Take nominal heads, for instance. May a nominal head be a N, Pro, Adj, Adv and even a Det in some cases?

(73) Let's call our neighbour

(74) Let's call John

(75) They pleased him

(76) She bought the blue one

(77) She always gives to the poor

(78) I have both

(79) He wants many of them

In this section, our goal is to decide which features defined at terminal level remain relevant at phrasal level for phrasal specification and subcategorisation. Thus, we are more interested in virtual and prototypical heads (cf. $\overline{X}$ theory) than in actual heads for our specifications. For each phrase, some terminal components must be examined in detail to check whether their morphosyntactic attributes would be relevant for phrasal restrictions.
It is well known that morphosyntactic attributes are language dependent. Consequently, phrasal morphosyntactic restrictions that are linked to terminal categories are also language dependent. The list of phrasal morphosyntactic features cannot be specified in absolute terms.
The method for specifying morphosyntactic restrictions on phrases is the following:

1. Prototypical components relevant for phrasal specification must be listed for each language;

2. Relevant attributes of these components must be extracted from the general morphosyntax framework to constitute the subset of one's language.

To illustrate these points, we examine what we derive for English verb complementation. Note that different tables would have to be established for noun, adjective and any other subcategorising category. Our first example covers *restricted S* (table 4.1) as illustrated in the example fragments (80)–(84):

| **Restricted S** | | | |
|---|---|---|---|
| **Component** | **Feature** | **Useful** | **Example** |
| verb | Type | no | |
| | Finite | yes | S[Fin:yes] |
| | Mood | no | |
| | Tense | maybe | S[Tense:past] |
| | Person | no | |
| | Numb | no | |
| | VForm | no | |
| | Aspect | maybe | S[Aspect:imperf] |
| | Voice | no | |
| | Reflexive | no | |
| | MainFunc | no | |
| | Aux | maybe | S[Aux:BE] |
| | AuxFunct | no | |
| pro | Type | no | |
| | Wh–type | yes | S[WType:inter] |
| | Pers | no | |
| | Numb | no | |
| | Gend | no | |
| | Case | no | |
| | Poss | no | |
| conj | Type | no | |
| | Coord–Type | no | |
| | Sub–Type | no | |

Table 4.1: Restricted S

(80) To expect that something will happen S[Fin:yes]

(81) She lets Mary cook on sundays

(82) I wonder why we should come S[WType:inter]

(83) I wonder whether we should come S[WType:inter]

| **restricted VP** | | | |
|---|---|---|---|
| **Component** | **Feature** | **Useful** | **Example** |
| verb | Type | no | |
| | Finite | yes | VP[Fin:no] |
| | Mood | maybe | VP[Mood:gerund] |
| | Tense | maybe | VP[Tense:past] |
| | Person | no | |
| | Numb | no | |
| | Gend | no | |
| | VForm | yes | VP[VForm:ING] |
| | Aspect | maybe | VP[Aspect:imperf] |
| | Voice | no | |
| | Reflexive | no | |
| | MainFunc | no | |
| | Aux | no | |
| | AuxFunct | no | |
| conj | Type | no | |
| | Coord-Type | no | |
| | Sub–Type | no | |

Table 4.2: Restricted VP

(84) I wonder who should come S[WType:inter]

Our next example covers *restricted VP* (table 4.2) as illustrated in the example fragments (85)–(86):

(85) You can lose your wallet VP[Fin:no]

(86) They started screaming VP[VForm:ING]

Our third example covers *restricted NP/PP* (table 4.3) as illustrated in the example fragments (87)–(89): Verb is introduced as a component of PP only for infinitivals and participials. Morphological number cannot be expected to represent a semantic contraint such as collective subject — collective subjects may occur as singular or plural. Impersonal pronoun will be directly specified on the PRONOUN terminal category, not on the non-terminal NP category. This is why the Type attribute associated with pronouns is not relevant for NP specifications.

(87) They pleased him NP[Case:acc]

(88) They asked him to come PP[Fin:no][Mood:inf]

(89) He excels in entertaining people PP[Fin:no][Mood:gerund]

Our final example concerns *restricted AP*, (table 4.4) as illustrated in the example fragment (90)

(90) He looks confidentx AP[Use:pred]

**REMINDER** — The tables given here only hold for English verb complementation.

#### 4.2.2.3 Lexical features

The selection of a specific lexical item is another important requirement for syntactic subcategorisation.

| Restricted NP/PP | | | |
|---|---|---|---|
| **Component** | **Feature** | **Useful** | **Example** |
| noun & det | Numb | no | |
| | Gend | no | |
| noun | Type | no | |
| pro | Type | no | |
| | Wh–type | maybe | NP[WhType:inter] |
| | Pers | yes | NP[Pers:3] |
| | Numb | no | |
| | Gend | no | |
| | Case | yes | NP[Case:acc] |
| | Poss | no | |
| det | Type | no | |
| | Wh–Type | no | |
| | Pers | no | |
| | Poss | no | |
| verb | Type | no | |
| | Finite | yes | PP[Fin:no] |
| | Mood | yes | PP[Mood:inf], PP[Mood:gerund] |
| | Tense | maybe | PP[Mood:part][Tense:past] |
| | Person | no | |
| | Numb | no | |
| | VForm | no | |
| | Aspect | no | |
| | Voice | no | |
| | Reflexive | no | |
| | MainFunc | no | |
| | Aux | no | |
| | AuxFunct | no | |

Table 4.3: Restricted NP/PP

| Restricted AP | | | |
|---|---|---|---|
| **Component** | **Feature** | **Useful** | **Example** |
| adj | Type | yes | AP[Type:qual] |
| | Degree | no | |
| | Numb | no | |
| | Gend | no | |
| | Use | yes | AP[Use:pred] |
| | Mod-Type | no | |

Table 4.4: Restricted AP

#### 4.2.2.3.1 Introducers

Bound prepositions such as *to* or *for* introducing a NP or a VP, complementisers such as *that* starting that–clauses or *whether* starting embedded interrogative clauses — all these elements are the best-known subcategorised lexicalised items. They can be gathered together under the class of *introducers*.
An *introducer* is:

- Placed at the leftmost end of the phrase it introduces;
- Most of the time semantically empty (bound prepositions can be viewed as case markers) or introduces a semantic relation (weakly bound prepositions) realised via a grammatical function word;

- A connector between its own phrase and a governor of higher range (this axiom excludes determiners from introducers).

The notion of introducer prevents us from making any statement about the link between the category of the introducer and the category of the introduced phrase. Introducers may equally be considered as true introducers or as phrase headings and heads. This depends on the theory and on the linguistic phenomena analysed. This feature provides for the means of specification of introducers which play a very special role in the left-to-right expansions of languages (*centrifugal* in Tesnièrian terminology).

(91) He wants *to* work — PP[Introd:TO] *vs.* VP[Introd:TO]

(92) He looked forward *to* the meeting — PP[Introd:TO] *vs.* NP[Introd:TO]

(93) He went *to* the movie — PP[Introd:TO] *vs.* NP[Introd:TO]

#### 4.2.2.3.2 Terminal elements

A terminal category realising a slot may also need to be lexicalised for the purpose of subcategorisation. In this case, since we are not at phrasal level but at terminal level, we will use another feature called *Lex*:

(94) There is — PRO[Lex:THERE]

(95) It rains — PRO[Lex:IT]

This feature applies to light–verb/noun dependency and any frozen or semi-frozen expression. However, we will not develop this issue here.

### 4.2.2.4 Syntactic features

#### 4.2.2.4.1 Passivisable

(96) Mary *expected* [a friend/to welcome a friend/that a friend came]

(97) A friend *is expected* by Mary

Direct objects which may be passivised will bear this property. This feature must be considered as a short cut for a frame alternation. It does not prevent one from describing the passive frame and the alternation between active and passive form.

#### 4.2.2.4.2 Other subcategorisation

Some rewriting specificity may be expected for subcategorised phrases. In such a case, and to avoid a complete rewriting, a *SubCat* feature will be associated with a phrase.

(98) Some salt

(99) *Plenty* <u>of salt</u> — PP[Introd:OF][SubCat:NODET]

## 4.2.3 Appendix

There follows a list of examples to illustrate how EAGLES proposes to encode well-known restrictions.

#### 4.2.3.1 Self restrictions

##### 4.2.3.1.1 Defective verbs

Impersonal verbs are usually taken to be defective (*pleuvoir* 'to rain'). Even so, they may be not defective in other constructions.

(100) Il *pleut* des cailloux sur la place

(101) Des cailloux *pleuvent* sur la place

In such cases, it is possible to mark the morphosyntactic restriction of 3rd person singular on the lexical unit only in association with the impersonal construction:

Self=[Pers:3][Numb:sg]

##### 4.2.3.1.2 Auxiliary

Different auxiliaries can be associated with the main verb for different constructions.

(102) To *be gone* down to the cellar Self=[Aux:BE]

(103) *Être descendu* à la cave Self=[Aux:ETRE]

(104) To *have gone* down the steps Self=[Aux:HAVE]

(105) *Avoir descendu* les escaliers Self=[Aux:AVOIR]

##### 4.2.3.1.3 Particles

English and German verbs have the particularity to be sometimes expanded with an adverbial particle. This particle is isomorphormic to a preposition, it may adhere to the verb or be separated from it by complements. Though it looks like a preposition, an adverbial particle does not introduce a complement. The particle is definitely not a slot realisation but is a component of the same morphological unit in different uses. The verb *set*, for example, may occur with or without a particle:

(106) He *set* the table

(107) You are all *set up* now Self=[Part:up]

(108) They *set up* an inquiry Self=[Part:up]

(109) I have *set* it all *up* for you Self=[Part:up]

##### 4.2.3.1.4 Pronouns

If an object is accidently coreferent with the subject, the reflexive or reciprocal pronoun is nothing else but one of the many realisations of an object slot.

###### 4.2.3.1.4.1 Reflexive

(110) She *trusted* <u>him</u>

(111) He *asked* <u>me</u> whether it was true

(112) He *asked* <u>himself</u> whether it was true

##### 4.2.3.1.4.2 Reciprocal

(113) They always *fight* their superiors

(114) They always *fight* each other

##### 4.2.3.1.4.3 Pronominal

If a pronoun has no reference at all, the verb is considered to be a pronominal verb. The pronoun is not a slot realisation but it is a component of the lexical unit.

(115) S'*exclama*-t-il Self=[Pro:SE]

(116) La tasse s'*est cassée* Self=[Pro:SE]

Some cases are controversial because the lack of referentiality is not so obvious.

(117) Il **s'***est lavé*

(118) Il s'*est lavé* la figure Self=[Pro:SE]

#### 4.2.3.2 S restrictions

##### 4.2.3.2.1 *that*–clause

(119) He *expected* that they would make some profits S[Finite:+][Introd:THAT]

##### 4.2.3.2.2 Subordinate interrogative clauses with total or partial questioning

(120) He *wondered* whether he should come S[Finite:+][WhType:Inter]

(121) He *wondered* who should come S[Finite:+][WhType:Inter]

#### 4.2.3.3 VP restrictions

##### 4.2.3.3.1 Infinitive clause

(122) He *wanted* to leave VP[Mood:infinitive]

##### 4.2.3.3.2 Participial clause

##### 4.2.3.3.2.1 *ing*–form

(123) They *started* screaming all around VP[VerbF:ing]

(124) He *talked* of leaving VP[VerbF:ing]

##### 4.2.3.3.2.2 *ed*–form

(125) He *looked* preoccupied by this meeting VP[VerbF:pastpart]

#### 4.2.3.4 PP restrictions

##### 4.2.3.4.1 Strongly bound prepositions

(126) He *counted* on the manager PP[Introd:on]

#### 4.2.3.4.2 Weakly bound prepositions

(127) He *jumped* into/under/… the train PP[Introd:into]/PP[Introd:under]/…

### 4.2.3.5 NP restrictions

#### 4.2.3.5.1 Impersonal subjects

(128) It *is raining* cats and dogs PRO[Lex:IT]

(129) There *is* a lot of rain PRO[Lex:THERE]

#### 4.2.3.5.2 Case

It is allowed to attach a case indication on a NP or a PP, even for languages with no case marking. This facility prevents the systematic recording of pronouns as slot realisations, which is possible if wished however.

(130) He/John *went* back home NP[Case:nom]

(131) I *missed* him/John so much NP[Case:acc]

#### 4.2.3.5.3 Passivisable direct objects

(132) Mary *expected* [a friend/to welcome a friend/that a friend would come]

(133) A friend *is expected* by Mary

## 4.3 Grammatical function

The general goal of current approaches to grammatical relations and relation changing (henceforth GRs and GR–changing) has been to establish a systematic link between the syntactic realisation of arguments and their encoding in lexical structure (see 3.1). In most cases, this enterprise has relied on the assumption that there is a strong correlation between the lexical properties of predicates (e.g. selectional restrictions) and the ways in which argument selection and selection changing proceed. This correlation has been captured by generalising argument types across predicates in terms of thematic relations. From a theoretical perspective, a considerable amount of effort has been devoted to relating generalisations about argument selection and selection changing to specific grammar frameworks. Three main orientations have emerged according to whether grammatical functions are:

1. Reduced to constituency relations between phrase markers;
2. Defined as primitive elements of the grammar; or
3. Derived from the semantic constituency of predicates.

### 4.3.1 Phrase structure based approaches

The practice of reducing GFs to constituency relations between phrase markers dates back to Chomsky's dissertation (Chomsky, 1955), and was described in some detail in Chomsky (1965). Chomsky's account consists in identifying structural relations of dominance such as '[NP,S]' and '[NP,VP]' which define thematic GRs at D–structure, and secondary (i.e. non-thematic) GRs at S–structure. Secondary GRs result from movement operations at S–structure induced by the *case filter* (see 3.1.1). Thematic GRs are derived from the categorial rules which generate D–structure phrase markers and specify the structural environment in which an argument is assigned a particular role by a predicate, according to the predicate's inherent lexical properties. The realisation of lexical properties is secured through the *Projection Principle* (see 3.1.1).

### 4.3.2 GFs as primitives

This is the approach taken in DUG and earlier versions of LFG (Bresnan, 1982); in later versions of LFG, grammatical functions are decomposed into primitive features (see 3.1.2 for details).

### 4.3.3 GFs derived from the semantic constituency of predicates

Working within a Montague Grammar (MG) framework, Dowty (Dowty, 1982a; Dowty, 1982b) proposes that GRs be derived from the hierarchical order in which a verb combines with its arguments. Following Montague and Schoenfinkel, Dowty assumes that verbs are *n*–place functions from argument denotations into $n - 1$–place functions:

> *Montague–Schoenfinkel Principle*
> A multi-place verb of *n* arguments is always represented as a functor of one argument that yields a *n-1* place verb (phrase) as its value. (Dowty, 1982b, 116)

It is the order inherent in functor–argument structures which gives rise to GRs. Although the operation which assembles the phonological form of a sentence may vary across languages, the semantic association of a verb with its arguments is carried out uniformly (i.e. through functional application) and therefore the hierarchical order in which a verb joins its arguments is invariant. In the light of this approach, GRs can be seen as deriving from this universal characterisation of the functor–argument association according to the following scheme:

1. A SUBJECT is any NP combined with an IV to produce a S.
2. A DIRECT OBJECT is any NP combined with a TV to produce a VP.
3. An INDIRECT OBJECT is any NP combined with a DTV to produce a TV.

This approach is followed in UCG and HPSG (see 3.1.3–3.1.4 ).

### 4.3.4 Conclusions

Since we are only concerned with subcategorisation, only subcategorisable functions are taken into consideration.

**Necessary specifications** — The subject/complement and ±predicative distinctions. All the theories we have reviewed make use of these differences.

**Recommended specification** — A further partition of complements into direct (DCOMP) and indirect (ICOMP). LFG, UCG and HPSG use this difference.

**Useful** — A more fine-grained distinction *à la* LFG, introducing further labels for clausal complements (COMP, XCOMP) and second objects (OBJ2).

## 4.4 Relative and independent order

### 4.4.1 Introduction

Usually, it can be considered that *grammar* is the proper place to describe how different elements of the subcategorisation frame of the lexical elements can be realised or instantiated in a real sentence, together with other non subcategorised elements. In particular, the different regularities concerning the possible linear order of different elements are part of the grammatical knowledge which is not encoded in the lexicon, as well as possible perturbations of the usual order. Grammar, in particular, describes the different *ordering constraints* (if there are any), which constrain the possible relative positions of different constituents or different functions within the sentence. Grammatical knowledge of general properties of a particular language (expressed outside the lexicon description) usually is enough to deal with most linear ordering limitations. Furthermore,

in most cases, syntactic frames associated with the lexicon have no need to have more information about order between slots or slot realisations in the same frame or about order relative to *self*, mainly because order constraints are usually general and independent of lexical entries.
However, there are great variations concerning ordering constraints depending on the language involved. *For some lexical units and for some languages*, particularly when aiming for a fine-grained lexical description, *some verbs may constrain the possible order of their slots or slot realisations more than others* (even if the same slots or slot realisations are involved). This then means that grammar (which cannot deal with such specific ordering) and lexicon together are incomplete unless the specific ordering constraints are explicitly present within the lexical description. The level of the frame is the proper place to encode such constraints when necessary, as the order of different syntactic objects linked to the lexical unit is at issue.
The slots are ordered in a conventional or canonical order. This order does not concern the linear ordering of realisations in real sentences. Surface order is supposed to be described outside the lexicon, unless special constraints are lexically dependent.

### 4.4.2 Order of realisations relative to *self* (the verb)

The standard (or most usual) linear position of slot realisation in relation to the verb can be specified at the slot realisation level, by using a special attribute, *in_order*.
This attribute defines the *preferred surface position of a slot realisation relative to the verb*. It takes three possible values: *pre, post*, meaning that the slot realisation preferentially occurs in pre- or post-verbal position, and *non-relevant* when there is no preference. This attribute expresses preference depending on the verb or the language in question. Importantly, the order relative to *self* can also be *fully constrained*: it is then expressed at frame level using relative order constraints. This will be explained in the following.

### 4.4.3 Ordering relation constraints between elements of a frame

Linear ordering constraints are potentially very complicated between realisations of different slots, and can be conditional, in particular, on optional positions when these are involved in the frame. We suggest here a simple way to deal with ordering constraints. We decided not to give the possibility of expressing a conditional ordering relation: the apparent need for such constraints is rather a criterion to split into two different frames in our approach. Thus, two complicated conditional ordering relations should be solved by splitting into two (or more) frames.
We propose to give the possibility of describing ordering constraints only when linking one element of one frame (slots or slot realisations) to another one or to the verb (*self*), different linear ordering constraints attached to the same frame being independent of each other.

### 4.4.4 Expression of linear ordering constraints

Linear ordering constraints are *attached to the frame* at its top level. A linear ordering constraint can link *two slots*, *two slot realisations*, *one slot and one slot realisation* or *one slot or slot realisation* to the verb (*self*). It is expressed through two attributes, always present, whose value is the index of the slot within the frame:

* ***before_slot*** – This is the slot that is first in linear ordering in real sentences instantiating the behaviour of a lexical entry associated with the frame concerned.

* ***after_slot*** — This is the slot that is the last (of the two) in linear ordering.

Slot numbering in a frame starts with 1. *Conventionally*, referring to slot '0' means we are referring to the verb (*self*) itself. Thus, constraints linking two slots or a slot and *self* are expressed in a uniform way. Two more attributes can be present or not:

* ***before_realisation*** — and respectively

* ***after_realisation*** — These make it possible to select a particular realisation for the selected slots (*before_slot* and *after_slot*, respectively). In this case, only the selected realisations are concerned by the

linear ordering constraints. The realisations are selected by pointing to one element in the list of realisations. When no realisations are selected for a slot (*before_slot* or *after_slot* or both), this means that all the realisations of this slot are concerned with the linear ordering constraint. When slot 0 (*self*) is selected, selection on realisation is not relevant, and thus not allowed.

As the above concern linear ordering, which is mainly a grammar problem, and as many (and maybe all, for some languages) regularities are captured by grammar, this descriptive element is not a fundamental one within the recommendations we provide. However, it provides good expressive power *for fine-grained descriptions* and *for some languages*, for which these constraints might result in being completely necessary.

## 4.5 Diathesis alternation and argument optionality

A crucial problem for lexicon design and construction is how to represent *diathesis alternation* phenomena, which occur when the same verb can appear in a variety of subcategorisation frames related to one another through valency alternations. Typical examples are:

(134) John ate the cake

(135) John ate

(136) John broke the glass

(137) The glass broke

Examples (134) and (135) exemplify the indefinite object alternation, according to which a given slot filler — the object in the case at hand — is optionally expressed. Examples (136) and (137) illustrate the causative–inchoative alternation, where the same slot filler can be associated with different syntactic functions, either the direct object of the transitive reading or the subject of the intransitive reading of the same verb.

### 4.5.1 Optionality

Earlier (section 4.1), we saw that each slot is specified for two different attributes: slot realisation and optionality. Accordingly, given examples (134) and (135) above, the relevant slot can be specified for optionality by means of the Boolean feature *Optionality*, as exemplified in table 4.5.

| | | |
|---|---|---|
| frame | ([slot | (Index: 1, [slot_real(Index: 1, CatLabel: np, Subject: subj)]), |
| | slot | (Index: 2, Optionality: opt, [slot_real(Index: 1, CatLabel: np, Subject: comp)])]) |

Table 4.5: Use of *Optionality*

In what follows, for the sake of readability, slot(Index: *i*) and slot_real(Index: *i*) will be shortened as slot_*i* and slot_real_*i* respectively.
It is not always the case that argument optionality can be stated in the simple terms of the example above. More difficult cases can occur as witnessed by the following examples:

(138) She paid

(139) She paid the bill

(140) She paid the bill to her mother

(141) *She paid to her mother

(142) She wrote

(143) She wrote the letter

(144) She wrote the letter to her mother

(145) She wrote to her mother

The difference in syntactic behaviour between *write* and *pay* cannot be captured conveniently through specification of the attribute Optionality. With *write*, omission of any slot filler is admissible, and can thus be encoded within the same frame as shown in table 4.6.

| | | |
|---|---|---|
| frame | ([slot_1 | ([slot_real_1(CatLabel: np, Subject: subj)]), |
| | slot_2 | (Optionality: opt, [slot_real_1(CatLabel: np, Subject: comp)]), |
| | slot_3 | (Optionality: opt, [slot_real_1(CatLabel: pp, Subject: comp)])]) |

Table 4.6: Encoding for *write*

On the contrary, with *pay*, omission of the direct object (e.g. *bill* in the examples at hand) additionally requires omission of the indirect object. The only way to deal with this case in the present framework is to encode two distinct but possibly related frames, only one of which contains an optional slot, as shown in tables 4.7 and 4.8, where frame_1 (short for frame(Index: 1)) represents the constructions in (138) and (139), and frame_2 (short for frame(Index: 2)) the construction in (140).

| | | |
|---|---|---|
| frame | (Index: 1, | |
| | [slot_1 | ([slot_real_1(CatLabel: np, Subject: subj)]), |
| | slot_2 | (Optionality: opt, [slot_real_1(CatLabel: np, Subject: comp)])]) |

Table 4.7: Encoding for *pay_1*

| | | |
|---|---|---|
| frame | (Index: 2, | |
| | [slot_1 | ([slot_real_1(CatLabel: np, Subject: subj)]), |
| | slot_2 | ([slot_real_1(CatLabel: np, Subject: comp)]), |
| | slot_3 | ([slot_real_1(CatLabel: pp, Subject: comp)])]) |

Table 4.8: Encoding for *pay_2*

As the reader will notice, this is not the only possible way of accounting for the constructions above: e.g. one could represent (139) and (140) as a single frame with an optional PP and an obligatory direct object, with (138) being assigned a distinct but possibly related frame.
To sum up, argument optionality is usually expressed by resorting to the attribute *Optionality*, whose possible values are { opt | no_opt }.
However, in particularly difficult cases such as *pay*, different but possibly related frames need to be specified. This encoding can be seen as a particular case of frame alternation to whose treatment we turn in the following section. In principle, the representation of optionality in terms of frame alternation can also be extended to the treatment of simpler cases such as *write* and *eat*. The lexicographer is left free to choose between the two options.

### 4.5.2 Frame alternations

Unlike argument optionality, *frame alternations* always require the independent description of more than one frame to be related at the frame_list level. Let us consider first the case of the causative–inchoative alternation exemplified in (138) and (139). Below, frame_1 (table 4.9) and frame_2 (table 4.10) describe the constructions *John broke the vase* and *the vase broke* respectively.

| | | |
|---|---|---|
| frame_1 | ([slot_1 | ([slot_real_1(CatLabel: np, Subject: subj)]), |
| | slot_2 | ([slot_real_1(CatLabel: np, Subject: comp)])]) |

Table 4.9: *John broke the vase*

frame_2 ([slot_1 ([slot_real_1(CatLabel: np, Subject: subj)])])

Table 4.10: *the vase broke*

In order to express the fact that, in these examples, *vase* is associated with two different slots in the two frames, we avail ourselves of the notation in table 4.11, specified at the frame_list level, which encodes the correspondence between slot_2 in frame_1 and slot_1 in frame_2.

frame_list ([related_1(FrameIndex: 1, SlotIndex: 2, FrameIndex: 2, SlotIndex: 1)])

Table 4.11: Notation at the frame_list level

There is a number of points which are worth stressing here. This notation departs from the traditional assumption that, given two alternating frames, one is taken to be basic and the other derivative, i.e. derived from the latter by means of lexical rules. EAGLES chose to represent all possible alternating frames of the same verb on a par by explicitly listing them within the lexicon; frame alternation is stated here as a relation between slots belonging to fully specified individual frames. The EAGLES approach is not as descriptively economic as the traditional one. On the other hand, it does not force the lexicographer to decide about the direction of the derivation, i.e., ultimately, about which frame is to be encoded explicitly within the lexicon as the base. Moreover, the EAGLES approach is compatible with the different treatments of frame alternation phenomena put forward in the literature and used in practical NLP lexicons, where alternation can be expressed (Comlex) or not (Eurotra), and in which slot relatedness can be encoded in a number of different ways (e.g. by enforcing identity of syntactic function, as in Comlex, or identity of thematic role, as in Acquilex).
More complex cases of frame alternation, which involve more than one slot pair, can also be found, as shown in examples (146) and (147).

(146) To give a letter to John

(147) To give John a letter

whose corresponding frames are given in tables (4.12) and (4.13).

| | | |
|---|---|---|
| frame_1 | ([slot_1 | ([slot_real_1(CatLabel: np, Subject: subj)]), |
| | slot_2 | ([slot_real_1(CatLabel: np, Subject: comp)]), |
| | slot_3 | ([slot_real_1(CatLabel: pp, Subject: comp)])]) |

Table 4.12: *to give a letter to John*

| | | |
|---|---|---|
| frame_2 | ([slot_1 | ([slot_real_1(CatLabel: np, Subject: subj)]), |
| | slot_2 | ([slot_real_1(CatLabel: np, Subject: comp)]), |
| | slot_3 | ([slot_real_1(CatLabel: np, Subject: comp)])]) |

Table 4.13: *to give John a letter*

In this case, relatedness involves two different slot pairs, namely slots_2 and slots_3 in both frames, as shown in table (4.14).

| | | |
|---|---|---|
| frame_list | ([related_1(FrameIndex: 1, SlotIndex: 2, | FrameIndex: 2, SlotIndex: 3), |
| | related_2(FrameIndex: 1, SlotIndex: 3, | FrameIndex: 2, SlotIndex: 2)]) |

Table 4.14: Relatedness of slot pairs

As pointed out above, argument optionality can also be treated as an instance of frame alternation: accordingly, the frames corresponding to the constructions of *eat* exemplified in (134) and (135) can be listed independently in the lexicon as in tables 4.15 and 4.16. Their relationship can be stated in terms of the correspondence between slots_1, i.e. the subjects, of both frames, as shown in table 4.17.

frame_1 ([slot_1 ([slot_real_1(CatLabel: np, Subject: subj)]),
slot_2 ([slot_real_1(CatLabel: np, Subject: comp)])])

Table 4.15: Frame for eat_1

frame_2 ([slot_1 ([slot_real_1(CatLabel: np, Subject: subj)])])

Table 4.16: Frame for eat_2

frame_list ([related_1(FrameIndex: 1, SlotIndex: 1, FrameIndex: 2, SlotIndex: 1)])

Table 4.17: Relationship between eat_1 and eat_2

### 4.5.3 Conclusions

EAGLES *recommends* specification of diathesis alternation and argument optionality in the lexicon, although this is not a universally adopted practice, as witnessed by those lexicons which list alternating frames as completely separate and unrelated lexical entries (e.g. Eurotra).
In EAGLES, diathesis alternation phenomena are captured at the frame_list level, which can be interpreted as a paradigmatic structure, where all alternants are encoded on a par. The main advantage of this encoding scheme is that all possible alternations are entered in the lexicon on a par, meaning that no single alternant is given a privileged status, say as a base from which all other alternants are derived through some lexical rule.
As for argument optionality, two alternative ways of encoding it are made available: either as a particular case of frame alternation (which requires listing of two distinct frames), or as a single frame where the optional argument is marked for optionality through feature specification.

## 4.6 Control

The following phenomena fall under the rubric of *control*:

1. How to represent the difference between raising and control (equi) verbs;
2. How to represent different kinds of raising verbs;
3. How to represent different kinds of control (equi) verbs.

We will deal with each of these in turn.

### 4.6.1 The distinction between raising and control verbs

To our knowledge, all theories agree that raising and control verbs differ at least *semantically*. Though these verb types may perhaps have the same syntactic properties, a raising verb takes one complement or subject which is not semantically an argument of this verb.
This can be illustrated with the verbs *try* and *seem*: they may both take a subject and an infinitival complement with *to*; however, with *try* the subject phrase is semantically an argument of the verb, but with *seem* it is not.
This can also be illustrated with the verbs *persuade* and *believe*: they may both take a subject, a direct object and an infinitival complement with *to*; however, with *persuade* the direct object is semantically an argument of *persuade*, but with *believe* the direct object is not semantically an argument of this verb.
If this semantic difference were the only difference between control and raising verbs, there would be no need to distinguish them syntactically. Only their semantic specifications would differ.
However, many theories claim that the distinction between control and raising verbs is not only semantic in nature, but also syntactic. This is true for GB, for HPSG, for GPSG and perhaps even for LFG.
There is interesting evidence from Icelandic that the distinction between control and raising verbs must indeed be represented syntactically. The evidence is basically the following. Consider verbs which combine with a subject and an infinitival complement, and take as the infinitive a verb which requires *quirky* case on its subject. Here, the subject of the infinitive cannot occur. We then have structures of the form: *NP V Vinf* ....

**The question is now** — What is the case of the subject NP?

**The answer is** — The case of this NP is determined by V if V is a control verb, but it is determined by Vinf (the case its subject would get) if V is a raising verb.

A similar example can be constructed for object-control and object-raising verbs. Consider verbs which take a subject, a direct object and an infinitival complement, and again take an infinitival verb which requires 'quirky' case on its subject. We then have structures of the form: *NPsubj V NPobj Vinf*....

**Now the question arises** — What case does *NPobj* bear?

**The answer is** — Its case is determined by V if V is a control verb, but it is determined by *Vinf* if V is a raising verb.

Since many theories require a syntactic distinction between control and raising verbs, it is proposed that the distinction between control and raising verbs is obligatorily indicated in the syntactic representation in the EAGLES standard. Theories which do not require this distinction can simply ignore this information.
The next question is: how do we represent the difference between control and raising verbs *syntactically*. Each theory has different mechanisms for this, but EAGLES should restrict itself to a simple distinction between control verb and raising verb by means of an attribute which can take two values. Let us call the attribute *CONTROL*, and let it have two possible values (for the moment), viz: *control-verb*, and *raising-verb*. It will be slightly refined below.

### 4.6.2 How to distinguish different kinds of raising verbs

Two different kinds of raising verbs must be distinguished: *subject-raising* verbs and *object-raising* verbs. It is proposed that this distinction is specified explicitly, by replacing the value *raising-verb* by two values: *subject-raising-verb* and *object-raising-verb*. No more is necessary, and perhaps even this is already redundant (since a raising verb which takes a direct object is necessarily an object-raising verb, and a raising verb which takes no direct object is a subject-raising verb).

### 4.6.3 How to distinguish different kinds of control verbs

Control verbs differ as to which argument controls the understood subject of an infinitive. For a verb like *promise* it is the subject, for a verb like *persuade* it is the (in)direct object, for a verb like *signal to* it is the prepositional object, etc.
It is proposed that this is represented in the slot realisation part of the syntax frame. The slot realisation part of an infinitival clause (or perhaps also, for certain languages, for finite clauses) can contain the attribute *controlled-by*. This attribute takes disjunctions of slot-identifiers as its possible values.
In the following examples a single frame is given. The slots in this frame are identified by non-negative integers. After each slot identifier, each slot realisation is contained in square brackets. Not all attributes are represented.

**promise:**

0 — [cat=NP,gf=subj,...],...

1 — [cat=NP,gf=obl,...],...

2 — [cat=Sinf,gf=obj,controlledby:0,...],[cat=NP,gf=obj,...],...

**persuade**

0 — [cat=NP,gf=subj,…],…

1 — [cat=NP,gf=obl,…],…

2 — [cat=Sinf,gf=obj,controlledby:1],[cat=PP,pform=of, gf=obl],…

Certain verbs allow either the subject or the complement to control an infinitival complement without a difference in meaning for the verb. Some examples from Dutch are *Ik stelde hem voor/ bood hem aan/ suggereerde hem weg te gaan* 'I proposed/ offered /suggested to him to leave'. For such cases we need disjunctive values such as *controlledby = 0 | 1*, etc. Thus *propose* might be represented as follows:

**propose**

0: — [cat=NP,gf=subj,…],…

1: — [cat=NP,gf=obl,…],…

2: — [cat=Sinf,gf=obj,controlledby:0 | 1,…], [cat=NP,gf=obj,…],…

If a difference in controller goes together with a different meaning of the verb, two different lexical items must be postulated. This is the case e.g. for French *dire*:

Jean a dit à Paul être très fatigueé
Jean has said to Paul be very tired
'Jean told Paul that he(=Jean) was very tired'
Jean a dit à Paul de partir
Jean has said to Paul to leave
'Jean told Paul to leave' (i.e that Paul should leave)

#### 4.6.3.1 Two types of control

An additional distinction must be made to distinguish a verb like *try* from a verb such as *decide*. For a verb such as *try* the controller must be the subject in all cases, but for a verb such as *decide* it can be the subject or the *by-phrase* (even when it is absent) or the understood subject can be interpreted as involving a group containing the referent of the subject or the by-phrase, or completely arbitrary. The difference can be illustrated in passivised structures, cf.
**It was tried (by the committee) to leave*.
versus
*It was decided (by the committee) to leave*.
This distinction is well-known from various frameworks. In GB it is known as the distinction between *obligatory* v. *nonobligatory* control, in LFG as the distinction between *functional* and *anaphoric* control, in HPSG a similar distinction is made as well (cf. Pollard and Sag, 'Head Driven Phrase Structure Grammar', chapter 7 and chapter 6, p. 290).
It is proposed that within EAGLES such theory-dependent notions are not introduced. Instead, the exact complementation properties and control properties of each individual lexical item are enumerated in full, at the cost of redundancy. However, if two frames which specify such complementation and control properties are related, they are put together in the same set of frames. In this manner, it is indicated that these frames are indeed related and not just two arbitrary variants.
The same approach is taken to *Bach's Generalisation* (i.e. with certain verbs, absence of the object when it functions as a controller is impossible), and *Visser's Generalisation* (certain verbs cannot be passivised if their subject functions as a controller). These generalisations are not incorporated themselves, but different frames resulting from them are put together in one set of frames to express their relatedness. Thus, though there is a need to stipulate different frames under these assumptions in certain cases, their relatedness is indicated.

It is proposed that if the understood subject can be interpreted as involving a group containing the referent of some argument, then this is specified by adding an asterisk after the relevant slot-identifier in the value of the *controlled-by* attribute.
It is also assumed that, if the controller is optional, it is specified as being an optional argument in the same frame. In this situation, it is not possible to describe the behaviour of the verb by means of two frames, since the controller cannot be specified correctly then.
If the choice of controller is not determined by properties of the lexical item itself, nothing is specified for the attribute *controlled-by*. French *falloir* (e.g. *il faut faire ça*) might be an example of such a case, since adding an overt controller yields ill-formed or marginal results: *?? il me faut faire ça*.

#### 4.6.3.2 Switch of controller

It is well-known that a characterisation in terms of slot identifiers or grammatical functions in itself is insufficient to fully characterise all control possibilities. In particular, cases of 'switch of controller' cannot be dealt with adequately. However, it appears to us that a lexical specification as given above is sufficient. It is possible to formulate general rules which account for 'controller switch' provided that a specification of the controller in terms slot identifiers is available. Such general rules will determine whether the controller switches and where it switches to by taking into account the lexical specifications plus properties of the embedded infinitival. We have in mind examples such as *Kim promised Sandy to be allowed to attend the party* (control by Sandy instead of by Kim), *Dana asked Pat to be allowed to attend the party* (control by Dana instead of by Pat), *Sandy was promised to be allowed to attend the party* (control by the subject of passivised *promise*, instead of ill-formedness[1] as in **Sandy was promised by John to take care of himself*). We thus suggest that nothing has to be specified lexically for such cases.

### 4.6.4 Summary

We will briefly recapitulate the attributes and their possible values proposed:

**CONTROL** — control-verb | subject-raising-verb | object-raising-verb

**CONTROLLEDBY** — disjunction of 0 .. N. This is an attribute–value pair for slot realisations of infinitival, and, in certain languages, finite clauses. *N* is here the maximum number of allowed slots in a frame. Each slot-identier *i* in the disjunction may be complemented by an asterisk (*), to indicate that a group containing the reference of the argument realised in accordance with slot *i* can serve as the controller.

## 4.7 Semantic class

Information concerning semantic class is considered in the present context only insofar as it enters into the account of lexically-governed syntactic processes (such as frame alternation). The kind of semantic class information that is of use mainly for disambiguation purposes (e.g. PP attachment) is not relevant at the level of linguistic description we are concerned with here, on the grounds that it involves processing issues which pertain to the role of semantics in NLP and which are to be dealt with accordingly by a prospective EAGLES Lexicon/Semantics group.
For each slot filler, semantic class information is optional, although it is highly recommended whenever it has a bearing on a proper account of phenomena which are already represented in the lexicon. A case in point is *frame alternation* (see section 4.5 on diathesis alternation above). By way of exemplification, consider the case of causative–inchoative alternation illustrated in (148)–(153):

(148) John began his career/the lesson/the concert/the show/the book

(149) His career/the lesson/the concert/the show/*the book began

(150) John gathered his friends/the animals/his papers/his maps/berries

[1] It must be noted that judgements on these sentences by native speakers are not uniform.

(151) His friends/the animals/*his papers/*his maps/*berries gathered

(152) Mary rang the bell/the door bell/*the telephone/*the alarm clock

(153) The bell/the door bell/the telephone/the alarm clock rang

These examples show that a given verb does not always undergo the alternation. For each example, possible and impossible alternations are contrasted. Two different cases can be distinguished: (148)–(149) and (150)–(151) show that not all possible objects of the transitive reading can play the role of subject of the intransitive reading of the same verb; conversely, (152)–(153) illustrate the case in which not all possible subjects of the intransitive reading can be the object of the transitive one.
From this it follows that the representation of the causative–inchoative alternation of different verbs in the lexicon requires that semantic class information concerning the argument alternating between two possible syntactic realisations (i.e. as object of the causative reading and as subject of the inchoative one) be also considered (Montemagni, Pirrelli & Ruimy, 1995; Montemagni & Pirrelli, 1995). For instance, with the verb *begin*, the alternation is possible only with event-denoting objects (e.g. concert, lesson, show) and with *gather*, only with inherently volitional ones (e.g. animals, friends). With *ring*, more subtle lexico-semantic distinctions have to be considered: for instance, *bell* and *door bell* can both be subject and object of the causative and inchoative readings respectively, but *telephone* and *alarm clock* can only play the subject role within the inchoative reading.
Starting from the simplest case, consider the possible representation of *gather* in an EAGLES-conformant lexicon, as shown in tables 4.18–4.20:

| | | | |
|---|---|---|---|
| frame_1 | ([slot_1 | ([slot_real_1(CatLabel: np, Subject: subj)]), |
| | slot_2 | ([slot_real_1(CatLabel: np, Subject: comp)])]) |

Table 4.18: gather_1

| | | |
|---|---|---|
| frame_2 | ([slot_1 | ([slot_real_1(CatLabel: np, Subject: subj)])]) |

Table 4.19: gather_2

| | | |
|---|---|---|
| frame_list | ([related_1(FrameIndex: 1, SlotIndex: 2, | FrameIndex: 2, SlotIndex: 1)]) |

Table 4.20: Frame relatedness for *gather*

Here, the condition is tacitly enforced that relatedness, as expressed at the frame_list level, holds if and only if the 'semclass' values specified at the slot level are met by both related slots. To be more concrete, in the case of *gather* the alternation holds only with volitional_animate entities, but fails to apply with entities that are not inherently volitional. In terms of extensional semantics, this means that an alternating argument must belong to the intersection between the sets of entities and the set of volitional animates.
A more complex case is represented by the verb *ring* where the causative–inchoative alternation appears to be sensitive to rather specific conceptual nuances like those distinguishing bells from telephones or alarm clocks. This example, when compared with the previous ones, shows that the semantic class information needed to account for frame alternation ranges from general restrictions, corresponding to broad semantic classes (such as 'animate', 'human' and the like) on the one hand, to highly specific restrictions on the other hand. It is not always easy to isolate linguistically relevant distinctions from cognitively possible ones as, in some cases, the distinction is not so clear-cut. This difficulty relates to the more general issue of how semantic class information should be represented in the lexicon. Although this problem is not immediately relevant here (arguably, a final decision on the matter should be taken at the level of semantic description), we raise this point at this juncture as an issue worthy of consideration, with a view to the development of standards for the representation of semantic information in the lexicon.

## 4.8 Thematic Roles

Thematic relations were introduced in generative grammar during the mid-1960s and early 1970s (Gruber, 1976; Fillmore, 1968; Jackendoff, 1972) as a way of classifying the arguments of natural language predicates into a closed set of participant types which were thought to have a special status in grammar. A list of the most popular roles and the properties usually associated with them is given below.

**Agent** — A participant which the meaning of the verb specifies as doing or causing something, possibly intentionally. Examples: subjects of *kill*, *eat*, *hit*, *smash*, *kick* and em watch.

**Patient** — A participant which the verb characterises as having something happen to it, and as being affected by what happens to it. Examples: objects of *kill*, *eat* and *smash* but not those of *watch*, *hear* and *love*.

**Experiencer** — A participant who is characterised as aware of something. Examples: subject of *love* or object of *annoy*.

**Theme** — A participant which is characterised as changing its position or condition, or as being in a state or position. Examples: objects of *give* and *hand*, subjects of *walk* and *die*.

**Location** — The thematic role associated with the NP expressing the location in a sentence with a verb of location. Examples: subjects of *keep*, *own*, *retain* and *know* and locative PPs.

**Source** — Object from which motion proceeds. Examples: subjects of *buy* and *promise*, objects of *deprive*, *free* and *cure*.

**Goal** — Object to which motion proceeds. Examples: subject of *receive* and *buy*, dative objects of *tell* and *give*. (Adapted from Dowty (1989))

Since its inception, the classification of argument positions into role types was meant to be carried out in terms of primitive semantic properties of predicates. Jackendoff (1972) suggested that thematic relations should be defined in terms of the three semantic subfunctions CAUSE, CHANGE and BE which constitute some of the primitive building blocks of lexical meanings. For example, the semantic representation of a transitive verb like *open* would be that of (154) where, according to Jackendoff's characterisation of roles in terms of semantic subfunctions, $\mathrm{NP}^1$ is agent and $\mathrm{NP}^2$ theme.

(154) $$\mathrm{CAUSE}\left(\mathrm{NP}^1, \begin{bmatrix} \mathrm{CHANGE} \\ physical \end{bmatrix} (\mathrm{NP}^2, \mathrm{NOT\ OPEN}, \mathrm{OPEN})\right)$$

An analogous proposal was developed by Dowty (1979) within a Montague Grammar framework and later adopted and extended by Foley & van Valin (1984).
Dowty (1989) assumes that there are only two 'thematic-role-like concepts' for verbal predicates: the *proto-agent* and *proto-patient* role. Proto-roles are conceived of as 'cluster-concepts' which are determined for each choice of predicate with respect to a given set of semantic properties. The properties which contribute to the definition of the proto-agent and proto-patient roles are listed below.

**Contributing properties for the proto-agent role** — Volition; sentience (and/or perception); causes event; movement.

**Contributing properties for the proto-patient role** — Change of state (including coming-to-being, going-out-of-being); incremental theme (i.e. determinant of aspect); causally affected by event; stationary (relative to movement of proto-agent).

According to Dowty, proto-roles are essentially meant for argument selection, e.g. lexical assignment of grammatical functions to subcategorised arguments. Proto-roles are related to argument selection through the *Argument Selection Principle* and two corollaries which govern the lexicalisation of GRs.

> *Argument Selection Principle*: The argument of a predicate having the greatest number of proto-agent properties entailed by the meaning of the predicate will, all else being equal, be lexicalised as the subject of the predicate; the argument having the greatest number of proto-patient properties will, all else being equal, be lexicalised as the direct object of the predicate.
>
> *Corollary 1*: If two arguments of a relation have (approximately) equal numbers of entailed proto-agent and proto-patient properties, then either may be lexicalised as the subject (and similarly for objects).
>
> *Corollary 2*: With a three-place predicate, the non-subject argument having the greater number of entailed proto-patient properties will be lexicalised as the direct object, the non-subject argument having fewer entailed proto-patient properties will be lexicalised as an oblique or prepositional object (and if two non-subject arguments have approximately equal entailed proto-patient properties, either may be lexicalised as direct object). (Dowty, 1987, 20)

The idea underlying this approach to argument selection is that the clustering of semantic properties such as those above provide a ranking according to which the arguments of a verb compete with one another for subjecthood and objecthood. For example, the subjects of a ditransitive verb such as *write* correspond to the arguments for which the properties volition, sentience, causation and motion are entailed, while the direct object argument is generally understood as being an incremental theme[2], causally affected and stationary as well as undergoing change. The indirect object in turn has fewer entailed proto-agent properties than the subject argument (e.g. it lacks volition and causation) and fewer proto-patient properties than the direct object arguments (it does not undergo change and is not causally affected). At parity of ranking, alternative lexicalisation patterns may arise. According to Dowty, this is what happens with lexical 'doublets' such as *buy* and *sell*:

> Consider first the case of *buy* vs. *sell*, *lend* vs. *borrow*. A sale transaction requires both a buyer and a seller to be sentient, to act volitionally, causally and — normally — with some movement (so that the buyer gets the sold object and the seller gets the buyer's money). Both these participants qualify well for subjecthood according to the selection principle, but moreover they qualify *equally* well, so [the selection principle] licences both lexicalisations.

In some cases, the determination of grammatical relations is more subtle. Consider the case of psychological verbs such as *like* and *please* where the syntactic realisation of the *experiencer* and *stimulus* arguments differ in spite of meaning similarities. Dowty observes that with respect to properties which promote agentivity (e.g. volition, sentience, causation, motion) either the stimulus or experiencer role can be realised as a subject.

> (i) the predicate entails that the experiencer has some perception of the stimulus — thus the experiencer is entailed to be sentient/perceiving though the stimulus is not — and (ii) the stimulus causes some emotional reaction or cognitive judgement in the experiencer. The first of these is a property that counts licensing the experiencer as subject, while the second is one that counts as licensing the stimulus as subject.

What tips the scale in favour of the stimulus argument with verbs such as *please* is the possibility of an inchoative interpretation which implies a change of state in the Experiencer as shown in (155). Argument roles which have the property of undergoing change of state are canonically more suitable to be realised as objects.

(155) The birthday party is pleasing Mary (right now)

All else being equal, psychological verbs which may express change of state (e.g. *amuse*, *please*, *frighten* and *irritate*) will thus realise the stimulus argument as subject and the experiencer as object. Interestingly enough, verbs such as *like*, as in (156)where the experiencer surfaces as the subject, do not seem to be able to give rise to an inchoative interpretation, i.e. may not be construed as expressing change of state (cf. Dowty (1987)):

[2] I.e. "Being an incremental theme" is the property ascribed to the argument of a verb whose reference properties are involved in the determination of telic aspect for the event denoted by the resulting verb phrase or sentence, e.g. *drink NP* describes either a process or an accomplishment according to whether the NP has cumulative (*beer*) or quantised (*a beer*) reference).

(156) *Mary is liking the birthday Party (right now)

Sanfilippo & Poznański (1992), Sanfilippo (1993b) and Sanfilippo (1993a) propose to extend the functionality of Dowty's prototype roles by including, in the defining clusters, properties which are instrumental for the identification of semantic verb (sub)classes. For example, it is well known that at least six subtypes of psychological verbs can be distinguished according to semantic properties of the stimulus and experiencer arguments (see Jackendoff (1972) and references therein), as in table 4.21.

| STIMULUS | | EXPERIENCER | | EXAMPLE |
|---|---|---|---|---|
| non-causative source | | neutral | reactive emotive | *experience* |
| ” | | positive | ” | *admire* |
| ” | | negative | ” | *fear* |
| neutral | caus. source | neutral | affected emotive | *interest* |
| positive | ” | positive | ” | *delight* |
| negative | ” | negative | ” | *scare* |

Table 4.21: Six subtypes of psychological verbs

This characterisation of psychological verbs can be rendered by defining a lattice of thematic sorts relative to the stimulus and experiencer arguments which extend prototype agent and patient roles, as in figure 4.3.

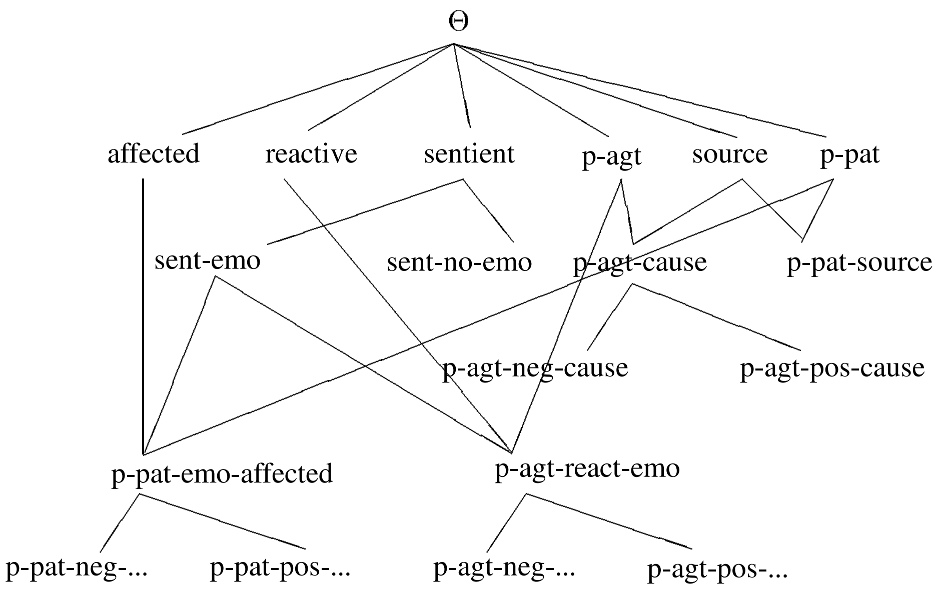



Figure 4.3: Lattice of thematic sorts

This approach can be further extended taking into account properties pertaining to other semantic classes of verbs (Austin, 1962; Searle, 1969; Talmy, 1985; Pustejovsky, 1991; Pustejovsky, in press; Asher & Lascarides, 1993; Sanfilippo, 1995a). For example, Sanfilippo (1995b) proposes to treat thematic roles as complex predicates which specify

- The proto-role involved (proto-agent/patient/oblique role);
- Basic properties such as *cause*, *change*, *be*, *source*, *path* and *goal* (the last four described in terms of event onset/mid/coda).

This is shown in figure 4.4.
The specification of basic properties is done in terms of features which are instrumental in identifying semantic verb subclasses, as in figure 4.5.

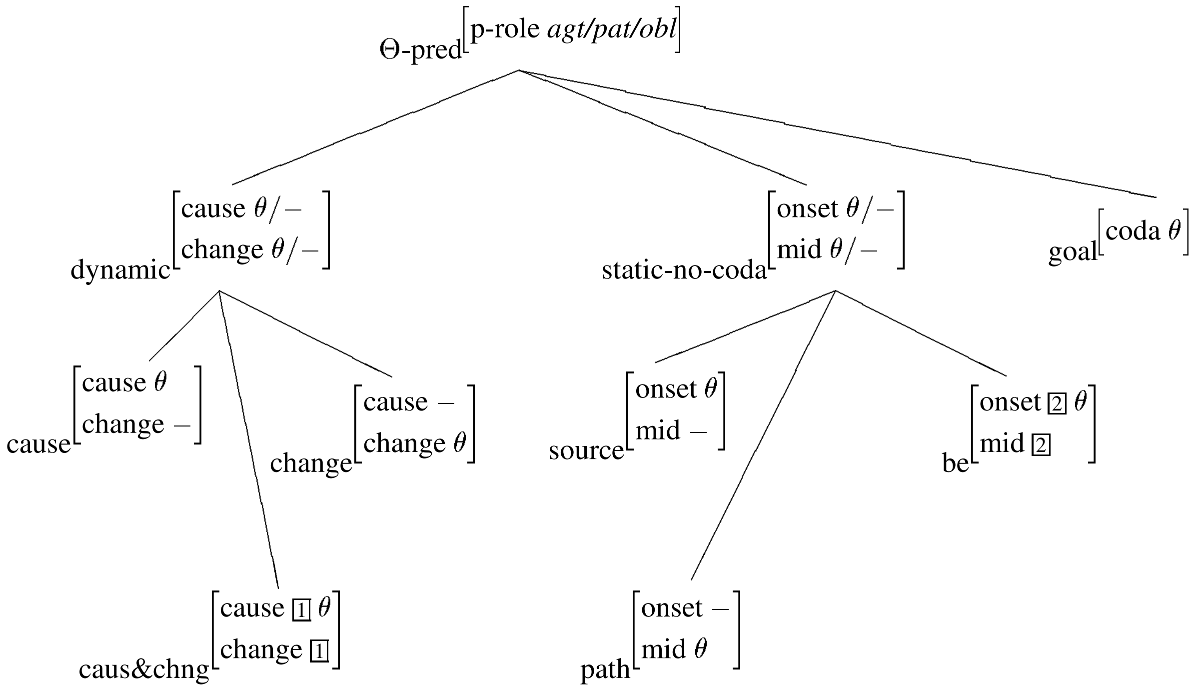



Figure 4.4: Thematic roles as complex predicates

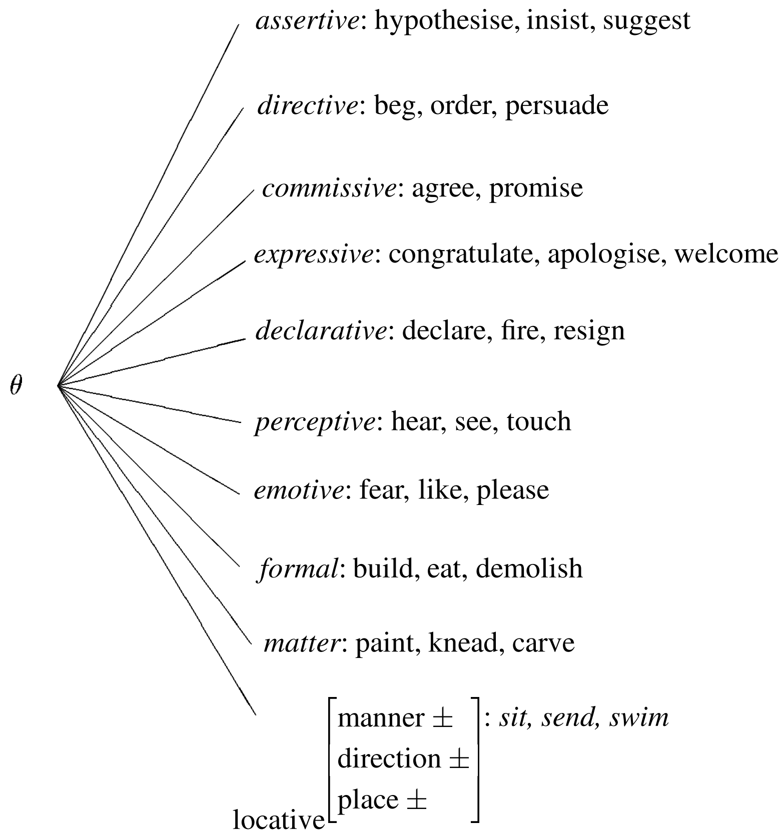



Figure 4.5: Specification of basic properties

The examples in figures 4.6–4.9 illustrate a few specific applications of the ensuing classification.

$$
\begin{array}{lll}
\text{John's grandmother} & \text{baked} & \text{a delicious cake} \\[2ex]
{}_{\text{cause}}\left[\text{cause } \boxed{1}\ \textit{formal}\right] & & {}_{\text{change}}\left[\text{change } \boxed{1}\right]
\end{array}
$$

Figure 4.6: Example 1

$$
\begin{array}{lll}
\text{Every true Oxonian} & \text{likes} & \text{to go fishing} \\[2ex]
{}_{\text{be}}\left[\begin{array}{l}\text{onset } \boxed{1}\ \textit{emot} \\ \text{mid } \boxed{1}\end{array}\right] & & {}_{\text{source}}\left[\text{onset } \boxed{1}\right]
\end{array}
$$

Figure 4.7: Example 2

$$
\begin{array}{lll}
\text{She} & \text{promised} & \text{Bill} \\[2ex]
{}_{\text{caus\&chng}}\left[\begin{array}{l}\text{cause } \boxed{1}\ \textit{commit} \\ \text{change } \boxed{1}\end{array}\right] & & {}_{\text{goal}}\left[\text{coda } \boxed{1}\right]
\end{array}
$$

Figure 4.8: Example 3

$$
\begin{array}{lll}
\text{A few participants in this competition} & \text{swim} & \text{for their country} \\[2ex]
{}_{\text{caus\&chng}}\left[\begin{array}{l}\text{cause } \boxed{1}\ \ {}_{\text{loc}}\left[\begin{array}{l}\text{man } + \\ \text{dir } -\end{array}\right] \\ \text{change } \boxed{1}\end{array}\right] & &
\end{array}
$$

Figure 4.9: Example 4

### 4.8.1 Conclusions

#### 4.8.1.1 Recommended specification

Use generic indexing (e.g. via integers) to distinguish event participants with possible inclusion of basic thematic properties such as *cause*, *change*, *be*, *source*, *path* or *goal*.

#### 4.8.1.2 For future reference

Enrich thematic specifications with

- Role names such as (proto)-agent/patient; and
- Features which are instrumental in identifying semantic verb classes (see figure 4.5).

# Chapter 5

# Formalisation

## 5.1 Conceptual representation of the specifications via E–R diagrams

The results of our work can be synthesised in an Entity–Relationship diagram (see figure 5.1) which is to be understood as follows. There are some conceptual elements, called entities, that are described by means of their:

- Attributes and possible values of the attributes;
- Associations (or relationships) with other entities.

Relationships link two entities and may carry attributes. Relationships importantly carry cardinality information that specifies whether an entity is necessarily or not involved in a relationship and whether it can be related to more than one entity with the same relationhip. Entities are here represented by boxes, relationships by a circle. In figure 5.1, just one relationship has attributes: the relationship between frame and slot. Cardinalities are expressed by pairs of numbers at each end of a relationship: each pair of numbers expresses the minimum and maximum number of relationships concerning all instances of the entity at that end of the relationship. So, when the minimum is 0, the relationship information is optional, and when it is 1 or more, it is obligatory. The cardinality indication *0,n* means that zero, 1, or several relationships of the same type can relate with an instance of the entity (i.e. optionality is specified); *1,n* means that 1 or several relationships can relate with an instance of the entity; *0,1* means zero or 1; *1,1* means exactly 1, and so on. For instance, a FRAME_SET entity can be related with 0, 1, or several RELATED entities and a RELATED entity is related with at least 1 but possibly many FRAME_SET entities.

## 5.2 SGML DTD for a formal description

A SGML DTD (Document Type Definition) is a physical model of the grammar that describes the mark-up of data. When translating our conceptual E–R model to a DTD we have adopted a systematic method, designed in particular to encode most of the constraints, cardinality constraints at least. The principles of transposition from the E–R formalism into SGML are as follows:

- E–R entities become SGML elements.
- Attributes of E–R entities become attributes of SGML elements. When the values of an attribute are mutually exclusive and when they constitute a closed vocabulary, then these values are represented in the form of listed SGML attributes.
- Relationships carrying no attributes, pointing at a shared E–R entity, are expressed by reference links between SGML elements (IDREF or IDREFS depending on the cardinality).
- Relationships that carry attributes are expressed in the form of SGML attribute-carrying elements, connected (by reference link) with the SGML elements translating the E–R entities.

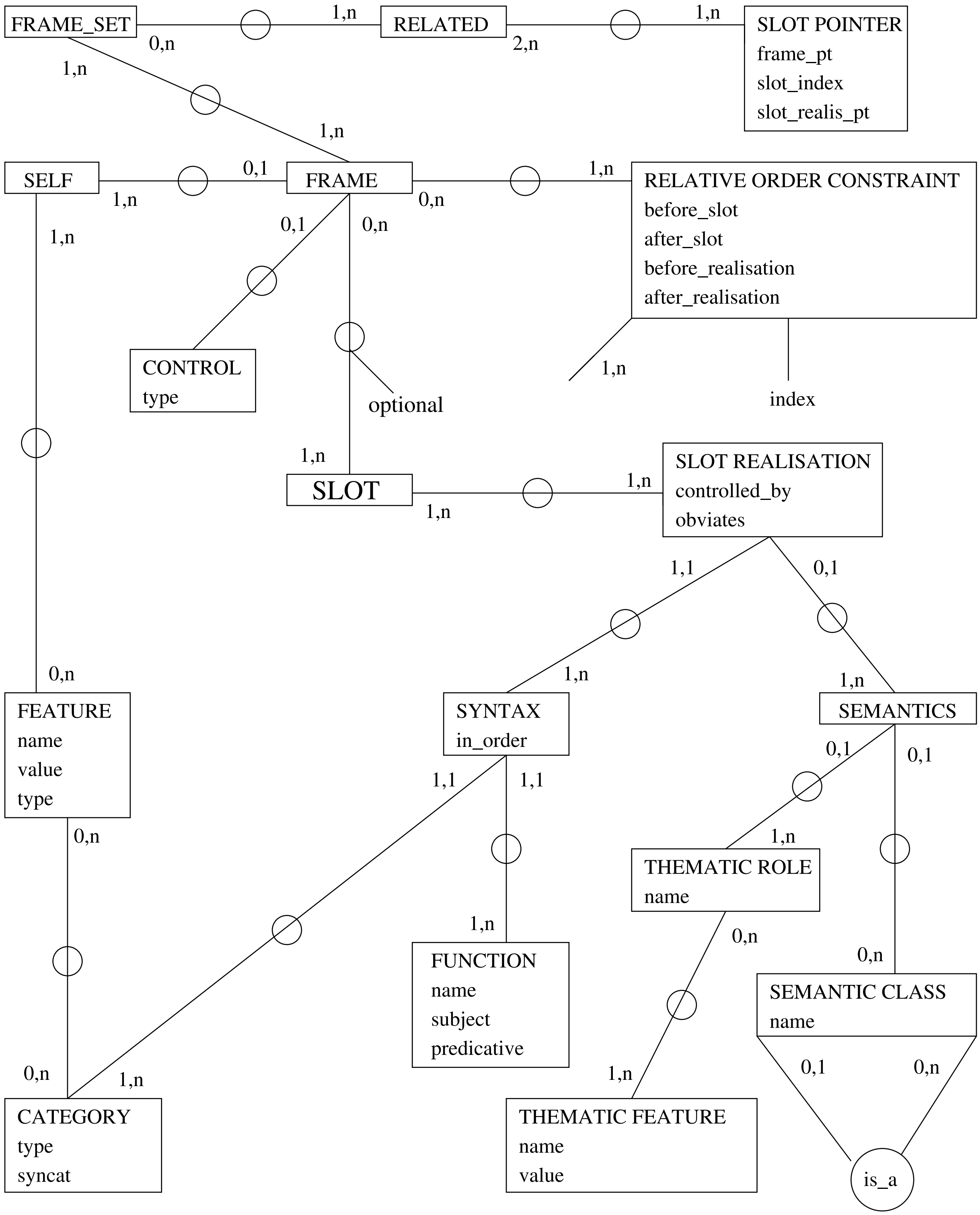



Figure 5.1: Entity–Relationship diagram for subcategorisation information

The model expressed at conceptual level by means of the Entity–Relationship formalism is now translated to a SGML DTD that describes a possible physical format of a file containing a lexicon conformant with the model, following the transposition principles we adopted. Thus, although we do not propose a complete E–R model of the lexicon, we here propose a subset of a complete DTD that could encode the lexicon and that could include this subpart to deal with syntactic information. On its own, it is not usable, because it has to be inserted in a global representation of the lexicon. As EAGLES is producing recommendations rather than a

complete and finished model, our work can be seen as an example of a possible SGML formalisation, not as a strong constraint.
To make it easier to understand a SGML DTD, we note:

- A SGML ELEMENT is here the expression of something like a structured object
- A SGML ELEMENT has attributes, and in particular the special attribute *id*, which is a unique key that makes it possible to refer to ELEMENTS by pointing to them from other ELEMENTS through their reference, using attributes that are of type IDREF (a kind of pointer).

### 5.2.1 Partial SGML DTD

```
<!ELEMENT Frame_set - O EMPTY>
<!ATTLIST Frame_set
      id              ID           #REQUIRED
      name            CDATA        #IMPLIED
      comments        CDATA        #IMPLIED
      related_l       IDREFS       #IMPLIED
      frame_l         IDREFS       #REQUIRED>
```

<!-- The Frame_set ELEMENT is the syntactic characterisation associated with the lexical unit to describe one of its syntactic behaviours. A lexical morphological unit will be associated with as many Frame_set ELEMENTs as it has different syntactic behaviours. A Frame_set has at least **one** FRAME, and possibly several associated FRAMEs, that can be related or not through the Related ELEMENTs that are pointed to by their ID (related_l attribute) -->

```
<!ELEMENT Related - O EMPTY>
<!ATTLIST Related
      id              ID           #REQUIRED
      name            CDATA        #REQUIRED
      comments        CDATA        #IMPLIED
      first_slot_pt   IDREF        #REQUIRED
      second_slot_pt  IDREF        #REQUIRED>
```

<!-- The Related ELEMENT associates one slot or slot realisation of one frame with another slot or slot realisation of another frame. It describes explicitly how two frames in a Frame set are related. first_slot_pt and second_slot_pt refer to Slot_Pointer ELEMENTs that are linked -->

```
<!ELEMENT Slot_pointer - O EMPTY>
<!ATTLIST Slot_pointer
      id              ID           #REQUIRED
      frame_pt        IDREF        #REQUIRED
      slot_index      NUMBER       #REQUIRED
      slot_real_pt    IDREF        #IMPLIED>
```

<!-- The Slot_pointer ELEMENT has different attributes:

- frame_pt points to a Frame via its ID,
- slot_index points to a slot of the pointed Frame via the index it has in the list of Ordered_Slot
- slot_real_pt refers to a Slot_Realisation via its ID, the realisation of the pointed-to Slot

Slot_Pointer gives access to a Slot of a Frame (when slot_real_pt is not present, a Slot_Realisation in the other case) -->

```
<!ELEMENT Frame - O EMPTY>
<!ATTLIST Frame
      id              ID           #REQUIRED
      name            CDATA        #IMPLIED
      comments        CDATA        #IMPLIED
      rel_order_cst_l IDREFS       #IMPLIED
      control         IDREF        #IMPLIED
      self            IDREF        #IMPLIED
      ordered_slot_l  IDREFS       #IMPLIED>
```

<!-- The Frame ELEMENT is the main descriptive ELEMENT of this model. It gives access to different types of information and to their syntactic and surface relations.

- rel_order_cst_l refers to a list of Relative_order_constraint that can be empty
- control refers to the Control information via its ID. It can be present or not.
- self refers to Self characteristics when in the Frame. It can be absent if no special characteristics are to be specified.
- ordered_slot_l: the list of subcategorised elements, called Slot, referred to by their ID. This list is not empty most of the time, however in limited cases it can be empty if the bare lexical unit is its own context (verbs with no subcategorised complements nor subject in certain languages, for instance weather verbs in Spanish).

-->

```
<!ELEMENT Relative_order_constraint - O EMPTY>
<!ATTLIST Relative_order_constraint
      id              ID           #REQUIRED
      before_slot     NUMBER       #REQUIRED
      after_slot      NUMBER       #REQUIRED
      before_reali    IDREF        #IMPLIED
      after_reali     IDREF        #IMPLIED>
```

<!-- Relative_order_constraint expresses one order constraint over all the realisations of one slot or just one, and all the realisations of one slot or just one or to self (lexical unit associated with the syntactic description)

- before_slot points via its_index to a slot (or to self) constrained to be the first in the surface instantiation of the frame
- after_slot points via its_index to a slot (or to self) after the before Slot on the surface
- before_reali selects just one realisation by reference to its ID to be affected by the constraint on the before slot
- after_reali selects just one realisation by reference to its ID to be affected by the constraint on the after slot

The slots are numbered counting from 1 (index count). Conventionally, 0 represents self. Thus, before_slot and after_slot can represent any slot or the lexical unit (verb) which is being described syntactically. This permits one to constrain the order between slots and the order between a slot and the self. -->

```
<!ELEMENT Ordered_slot - O EMPTY>
<!ATTLIST Ordered_slot
      id              ID           #REQUIRED
      comments        CDATA        #IMPLIED
      index           NUMBER       #REQUIRED
      slot            IDREF        #REQUIRED
      optional       (YES|NO)      NO>
```

<!-- Ordered_slot ELEMENT associates a Slot with a Frame expressing some attributes of this association:

- index gives its conventional or canonical rank in the list of slots of the frame
- slot refers to the Slot via its id
- optional expresses the optional or obligatory character of the Slot when associated with the Frame.

-->

```
<!ELEMENT Slot - O EMPTY>
<!ATTLIST Slot
      id              ID            #REQUIRED
      name            CDATA         #IMPLIED
      comments        CDATA         #IMPLIED
      slot_reali_l    IDREFS        #REQUIRED>
```

<!-- Slot ELEMENT is a generalised subcategorised element of the Frame, characterised by the fact that it has at least one realisation and sometimes several. Slot can be related to other slots of the same frame via relative order constraints and also to other slots of other frames in the same frameset through Related ELEMENT. slot_reali_l refers to the Slot_realisation of the Slot, at least one. -->

```
<!ELEMENT Slot_Realisation - O EMPTY>
<!ATTLIST Slot_Realisation
      id              ID            #REQUIRED
      name            CDATA         #IMPLIED
      comments        CDATA         #IMPLIED
      controlled_by   NUMBERS       #IMPLIED
      obviates        NUMBERS       #IMPLIED
      in_order        (PRE|POST|NOT_RELEVANT)    POST
      syntax          IDREF         #REQUIRED
      semantics       IDREFS        #IMPLIED>
```

<!-- Slot_Realisation ELEMENT is one element of the distribution of a Slot

- controlled_by points, via indices, to the possible slots that are the controllers (usually one, but sometimes more than one possible controller, the list has to be understood as alternative elements) of a realisation when it is, for instance, an infinitival,
- obviates points, via indices, to impossibly coreferring slots (usually none or one, but sometimes more than one, to be understood as AND)
- in_order specifies the usual/most probable/canonical place of the realisation in relation to the verb that subcategorises for it
- syntax points to the syntax properties (Syntax ELEMENT) of the realisation through its ID. It is the only information necessarily present for Slot_Realisation
- semantics points to the semantic properties (Semantics ELEMENT) of the realisation through its ID. This element can be absent, and its presennce depends on the lexical entry and on the choices that have been made to define the boundary between syntactic and semantic description.

-->

```
<!ELEMENT Syntax - O EMPTY>
<!ATTLIST Syntax
      id              ID            #REQUIRED
```

```
      name            CDATA        #REQUIRED
      comments        CDATA        #IMPLIED
      function        IDREF        #REQUIRED
      category        IDREF        #REQUIRED>
```

<!-- Syntax ELEMENT is the syntactic property of one realisation of slot It is the association of two information

- function is a pointer to a Function, it is obligatory
- category is a pointer to a Category, it is alsoobligatory

-->

```
<!ELEMENT Function - O EMPTY>
<!ATTLIST Function
      id              ID           #REQUIRED
      comments        CDATA        #IMPLIED
      name            CDATA        #REQUIRED
      subject         (YES|NO]     NO
      predicative     (YES|NO]     NO>
```

<!-- Function ELEMENT describes the syntactic (or functional) relation that links the realisation and the lexical unit (self) in the frame. It is described via two binary attributes that are elementary properties of functions, and its name can make it more precise. The list of possible names of functions is not given in this DTD but has to be defined for any given lexicon.-->

```
<!ELEMENT Category - O EMPTY>
<!ATTLIST Category
      id              ID           #REQUIRED
      name            CDATA        #IMPLIED
      comments        CDATA        #IMPLIED
      type            (TERM|NON_TERM)  NON_TERM
      label
(N|AJ|PR|V|AV|CJ|AD|DT|AR|IN|S|VP|NP|AP|ADVP|DETP)
                                   NP
      valued_feat_l  IDREF        #REQUIRED>
```

<!-- Category ELEMENT describes the syntactic class of the realisation by means of the kind of constituent it is. It is an association of three types of information

- type for the type of constituent (terminal or non-terminal).
- label to precisely constrain the constituent. A finite list is proposed, but could be extended to the needs of some languages
- valued_feat_l to constrain the constituent with different morphosyntactic or syntactic or lexical features.

-->

```
<!ELEMENT Semantics - O EMPTY>
<!ATTLIST Semantics
      id              ID           #REQUIRED
      name            CDATA        #IMPLIED
      comments        CDATA        #IMPLIED
      themat_rol      IDREF        #REQUIRED
      semant_class    IDREF        #REQUIRED>
```

<!-- Semantics ELEMENT describes the semantic properties of the realisation. It is an association of two types of information

- themat_rol gives the type of thematic relation that links the element to the lexical unit that subcategorises for it.
- semant_class refers to the semantic class compatibility constraining the semantic interpretation of the realisation.

-->

```
<!ELEMENT Thematic_rol - O EMPTY>
<!ATTLIST Thematic_rol
      id              ID            #REQUIRED
      comments        CDATA         #IMPLIED
      name            CDATA         #REQUIRED
      role_feat_l     IDREFS        #IMPLIED>
```

<!-- Thematic_rol ELEMENT can specify the thematic role combining elementary role properties. The set of elementary properties is associated with a name that, when present, it defines. The only obligatory information is the name; the list of possible thematic roles (and possible values of name) is not constrained by this model, but has to be defined for each lexicon. -->

```
<!ELEMENT Semantic_class - O EMPTY>
!ATTLIST Semantic_class
      id              ID            #REQUIRED
      comments        CDATA         #IMPLIED
      name            CDATA         #REQUIRED
      is_a_sem_cl_l   IDREFS        #IMPLIED>
```

<!-- Semantic_class ELEMENT can semantically constrain the semantic interpretation of a realisation that is associated with it to be compatible with it. It is defined by its name and, when present, by its hierarchical relations to other classes, expressed through the is_a_sem_cl_l which refers to no, one, or several Semantic_class under which the semantic class is located (the linking relation is a IS_A relation. The more specific class points to the more general ones it inherits from. -->

```
<!ELEMENT Role_feature - O EMPTY>
<!ATTLIST Role_feature
      id              ID             #REQUIRED
      name            CDATA          #REQUIRED
      value           CDATA          #REQUIRED>
```

<!-- Role_feature is an elementary feature for characterising thematic roles. It has name and value, completely unconstrained by this proposal.-->

# Bibliography


Alsina, A. & Mchombo, S. (1988). Lexical mapping in the Chichewa applicative constructions, *Technical report*, Stanford University and San Jose State University, Stanford.

Asher, N. & Lascarides, A. (1993). Lexical disambiguation in a discourse context, *Technical report*, Centre for Cognitive Science, University of Edinburgh, Edinburgh.

Austin, J. (1962). *How to do things with words*, Clarendon Press, London.

Boguraev, B. & Briscoe, T. (eds) (1989). *Computational lexicography for natural language processing*, Longman, London.

Bresnan, J. (ed.) (1982). *The mental representation of grammatical relations*, The MIT Press, Cambridge, Ma.

Bresnan, J. & Kanerva, J. (1988). Locative inversion in Chichewa: A case study of factorization in grammar, *Technical report*, Stanford University and Xerox PARC, Stanford.

Bresnan, J. & Moshi, L. (1989). Object asymmetries in comparative Bantu syntax, *Linguistic Inquiry* **21**.

Chomsky, N. (1955). *The logical structure of linguistic theory*, Plenum, New York. Published in 1975.

Chomsky, N. (1965). *Aspects of the theory of syntax*, The MIT Press, Cambridge, Ma.

Chomsky, N. (1981). *Lectures on government and binding*, Foris Publications, Dordrecht.

Chomsky, N. (1986a). *Barriers*, The MIT Press, Cambridge, Ma.

Chomsky, N. (1986b). *Knowledge of language: Its nature, origin and use*, Praeger, New York.

Davis, A. (1996). *Lexical semantics and linking in the hierarchical lexicon*, D. Reidel, Dordrecht.

Di Sciullo, A. M. & Williams, E. (1987). *On the definition of word*, The MIT Press, Cambridge, Ma.

Dowty, D. (1979). *Word meaning and Montague Grammar*, D. Reidel, Dordrecht.

Dowty, D. (1982a). Grammatical relations and Montague Grammar, *in* P. Jacobsen & G. Pullum (eds), *The nature of syntactic representation*, D. Reidel, Dordrecht, pp. 79–130.

Dowty, D. (1982b). More on the categorial analysis of grammatical relations, *Ohio State University Working Papers in Linguistics* **26**: 102–133.

Dowty, D. (1987). Thematic proto roles and argument selection, *Language* **67**: 547–619.

Dowty, D. (1989). On the semantic content of the notion 'thematic role', *in* G. Chierchia, B. Partee & R. Turner (eds), *Property theory, type theory and natural language semantics*, D. Reidel, Dordrecht.

EAGLES (1996). Synopsis and comparison of morphosyntactic phenomena encoded in lexicons and corpora. A common proposal and applications to european languages, *Technical Report EAG–LWG–Morphsyn*, ILC–CNR, Pisa. Available at:
http://www.ilc.pi.cnr.it/EAGLES96/morphsyn/morphsyn.html.

Fillmore, C. (1968). The case for case, *in* E. Bach & R. Harms (eds), *Universals in linguistic theory*, Holt, Rinehart and Winston, New York.

Foley, W. & van Valin, R. (1984). *Functional syntax and Universal Grammar*, CUP, Cambridge.

Fraser, N. (1994). *Dependency parsing*, PhD thesis, UCL, London.

GENELEX (1993). Projet Eureka GENELEX. Rapport sur la couche syntaxique, *Technical Report Version 4.0*, ASSTRIL, GSI-ERLI, IBM France and Sema Group, Paris.

Gruber, J. (1976). *Lexical structure in syntax and semantics*, North Holland, New York.

Heid, U. & McNaught, J. (eds) (1991). *EUROTRA-7 Study: Feasibility and project definition study of the reusability of lexical and terminological resources in computerised applications*, IMS, University of Stuttgart, Stuttgart.

Hellwig, P. (1983). Dependency Unification Grammar, *in* E. Hajicova (ed.), *Functional Description of Language*, Charles University, Prague, pp. 67–84.

Jackendoff, R. (1972). *Semantic interpretation in generative grammar*, The MIT Press, Cambridge, Ma.

Jensen, K., Heidorn, G. & Richardson, S. (1993). *Natural language processing: The PLNLP approach*, Kluwer Academic Publishers, Boston, Ma.

Kaplan, R. & Bresnan, J. (1982). Lexical-Functional Grammar: A formal system for grammatical representation, *in* J. Bresnan (ed.), *The mental representation of grammatical relations*, The MIT Press, Cambridge, Ma.

Karlsson, F., Voutilainen, A., Heikkilä, J. & Antilla, A. (1995). *Constraint Grammar: A language-independent system for parsing unrestricted text*, Mouton de Gruyter, Berlin.

Levin, L. (1987). Toward a linking theory of relation changing rules in LFG, *Research report CSLI-87-115*, CSLI, Stanford.

Lieber, R. (1980). *On the organization of the lexicon*, PhD thesis, MIT.

Manzini, R. (1983). On control and control theory, *Linguistic Inquiry* **14**: 421–446.

Marcus, M., Kim, G., Marcinkiewicz, M., MacIntyre, R., Bies, M., Ferguson, M., Katz, K. & Schasberger, B. (1994). The Penn Treebank: Annotating predicate argument structure, *Proceedings of the Human Language Technology Workshop*, Morgan Kaufmann, San Francisco.

Marcus, M., Marcinkiewicz, M. & Santorini, B. (1993). Building a large annotated corpus of English: The Penn Treebank, *Computational Linguistics* **19**(2): 313–330.

Montemagni, S. & Pirrelli, V. (1995). Do lexical rules apply across the board? A corpus-based investigation in the machinery of the causative-inchoative alternation in Italian, *Proceedings of the Acquilex-II Workshop on Lexical Rules*, Cambridge.

Montemagni, S., Pirrelli, V. & Ruimy, N. (1995). Ringing things which nobody can ring: A corpus-based study of the causative-inchoative alternation in Italian, *Acquilex-ii working paper*, ILC–CNR, Pisa. Also to appear in Textus.

Procter, P. (1987). *Longman dictionary of contemporary English*, Longman, London.

Pustejovsky, J. (1991). The generative lexicon, *Computational Linguistics* **17**(4).

Pustejovsky, J. (in press). Linguistic constraints on type coercion, *in* P. St. Dizier & E. Viegas (eds), *Computational lexical semantics*, CUP, Cambridge.

Quirk, R., Greenbaum, S., Leech, G. & Svartvik, J. (1985). *A comprehensive grammar of the English language*, Longman, London.

Rappaport, M. (1983). On the nature of derived nominals, *in* L. Levin, M. Rappaport & A. Zaenen (eds), *Papers in Lexical-Functional Grammar*, University Linguistics Club, Bloomington, Indiana.

Rohen Wolff, S., Macleod, C. & Meyers, A. (1994). COMLEX word classes, *Manual*, Linguistic Data Consortium, UPenn.

Sampson, G. (1995). *English for the computer: The SUSANNE corpus and analytic scheme*, Clarendon Press, Oxford.

Sanfilippo, A. (1993a). Grammatical relations in Unification Categorial Grammar, *Lingua e Stile* **XXVII**(2).

Sanfilippo, A. (1993b). LKB encoding of lexical knowledge, *in* T. Briscoe, A. Copestake & V. de Paiva (eds), *Default inheritance in unification-based approaches to the lexicon*, CUP, Cambridge.

Sanfilippo, A. (1995a). Lexical polymorphism and word disambiguation, *Working notes of the AAAI Spring Symposium on The Representation and Acquisition of Lexical Knowledge*, Stanford University.

Sanfilippo, A. (1995b). Thematic roles: Diathesis/sense alternations and argument selection, *Paper presented at the 1995 HPSG Workshop*, Tubingen.

Sanfilippo, A. & Poznański, V. (1992). The acquisition of lexical knowledge from combined machine readable dictionary sources, *Proceedings of the 3rd Conference on Applied Natural Language Processing*, Trento.

Searle, J. (1969). *Speech acts*, The MIT Press, Cambridge, Ma.

Talmy, L. (1985). Lexicalization patterns: Semantic structure in lexical form, *in* T. Shopen (ed.), *Grammatical categories and the lexicon*, Vol. 3 of *Language typology and syntactic description*, CUP, Cambridge.

ten Hacken, P., Maas, H. & Maegaard, B. (1991). Dictionaries in Eurotra, *in* C. Copeland, J. Durand, S. Krauwer & B. Maegaard (eds), *The Eurotra linguistic specifications*, The Commission of the European Communities, Brussels.

Tesnière, L. (1959). *Eléments de syntaxe structurale*, C. Klinksieck, Paris.

Vanocchi, M., Rosini, R., Carenini, M., Prodanof, I. & Calzolari, N. (1994). Italian verbs: Developing a neutral formalism for verbal representation, *Technical Report ILC–NLP–1994–1*, ILC–CNR, Pisa.

Williams, E. (1980). Predication, *Linguistic Inquiry* **11**: 81–114.

Williams, E. (1981a). Argument structure and morphology, *Linguistic Review* **1**: 81–114.

Williams, E. (1981b). On the notions 'Lexically Related' and 'Head of a Word', *Linguistic Inquiry* **12**: 245–274.

# Appendix A

# Appendix: Examples

This appendix provides examples of how the EAGLES Subcategorisation scheme may be applied to a number of languages.

## A.1 Applying the EAGLES subcategorisation standards

By way of summary, here is a description in feature structure format of the frame and frame-list representations which have been used to compile sample applications of the EAGLES Subcategorisation scheme.

$$
\left[\begin{array}{l}
\text{FRAME\_ID } \textit{integer} \\
\text{SLOTS } \left\langle \left[\begin{array}{l}
\text{SLOT\_ID } \textit{integer} \\
\text{OPTIONALITY } \textit{optionality} \\
\text{REALISATIONS } \left\langle \left[\begin{array}{l}
\text{REALISATION\_ID } \textit{integer} \\
\text{CONTROLLED\_BY } \left[\begin{array}{l} \text{SLOT\_ID } \textit{integer} \\ \text{REALISATION\_ID } \textit{integer} \end{array}\right] \\
\text{OBVIATES } \left[\begin{array}{l} \text{SLOT\_ID } \textit{integer} \\ \text{REALISATION\_ID } \textit{integer} \end{array}\right] \\
\text{IN\_ORDER } \textit{in\_order} \\
\text{SYNTAX } \left[\begin{array}{l} \text{CATEGORY } \left[\begin{array}{l} \text{CAT\_LABEL } \textit{cat\_label} \\ \text{CAT\_FEATURES } \textit{cat\_features} \end{array}\right] \\ \text{FUNCTION } \left[\begin{array}{l} \text{FUNC\_LABEL } \textit{func\_label} \\ \text{PRED } \textit{pred} \end{array}\right] \end{array}\right] \\
\text{SEMANTICS } \left[\begin{array}{l} \Theta\text{\_ROLE } \left[\begin{array}{l} \Theta\text{\_ID } \textit{integer} \\ \Theta\text{\_FEATURES } \Theta\textit{\_features} \end{array}\right] \\ \text{SEM\_CLASS } \textit{sem\_class} \end{array}\right]
\end{array}\right] + \right\rangle
\end{array}\right] * \right\rangle \\
\text{SELF } \textit{cat\_features} \\
\text{CONTROL } \textit{control} \\
\text{REL\_ORDER } \left\langle \left[\begin{array}{l}
\text{BEFORE\_SLOT } \textit{integer} \\
\text{AFTER\_SLOT } \textit{integer} \\
\text{BEFORE\_REALISATION } \textit{integer} \\
\text{AFTER\_REALISATION } \textit{integer}
\end{array}\right] * \right\rangle
\end{array}\right]
$$

$$\begin{bmatrix} \text{FRAME\_LIST } \langle \textit{frame}_1,\textit{frame}_2,...,\textit{frame}_n \ \rangle \\ \text{RELATED\_LIST } \langle \begin{bmatrix} \text{1ST} & \begin{bmatrix} \text{FRAME\_ID } \textit{integer} \\ \text{SLOT\_ID } \textit{integer} \\ \text{REALISATION\_ID } \textit{integer} \end{bmatrix} \\ \text{2ND} & \begin{bmatrix} \text{FRAME\_ID } \textit{integer} \\ \text{SLOT\_ID } \textit{integer} \\ \text{REALISATION\_ID } \textit{integer} \end{bmatrix} \end{bmatrix}^{*} \rangle \end{bmatrix}$$

A list of expansions for values other than *integer* is provided below.

| | |
|---|---|
| $optionality$ | $opt \mid no\_opt$ |
| $in\_order$ | $prep \mid post \mid irrelevant$ |
| $cat\_label$ | $n \mid aj \mid pr \mid v \mid av \mid cj \mid ad \mid dt \mid ar \mid in \mid s \mid vp \mid np \mid ap \mid advp \mid detp$ |
| $cat\_features:$ VERB | $fin \mid nfin$ |
| | $ind \mid sub \mid impr \mid cond \mid inf \mid part \mid ger \mid sup \mid ing \mid bas$ |
| | $pres \mid impf \mid fut \mid past$ |
| | $pfve \mid ifve$ |
| | $hv \mid be$ |
| $cat\_features:$ (PRO)NOUN | $nom \mid gen \mid dat \mid acc$ |
| | $dem \mid indf \mid poss \mid int \mid rel \mid pers \mid refl \mid recp \mid excl$ |
| | $1 \mid 2 \mid 3$ |
| | $passivisable \mid no\_passivisable$ |
| $cat\_features:$ ADJ | $qual \mid poss \mid indf \mid card \mid ord$ |
| $cat\_features:$ LEX | $string$ |
| $cat\_features:$ INTROD | $string$ |
| $control$ | $control \mid subj\_raising \mid obj\_raising$ |
| $func\_label$ | $subj \mid comp \mid dcomp \mid icomp$ |
| $pred$ | $pred \mid npred$ |
| $\Theta\_features$ | $cause \mid change \mid be \mid source \mid path \mid goal$ |
| $sem\_class$ | (to be developed) |

A comparison of these descriptions with the SGML encoding given in section 5.2 will reveal a small degree of deviation. Application of a classifying scheme always requires some adjustments, and we would indeed welcome the possibility of examining future uses of the EAGLES subcategorisation standards in order to update the scheme periodically. The addition and changes introduced in this chapter will eventually be integrated with the formal description given in section 5. Also, a version of the EAGLES Subcategorisation Scheme will soon be implemented in ALE (Carpenter, 1992), as a part of the lexical specification work in the SPARKLE project (LE1–2111).
For ease of presentation, examples are given using macros which abbreviate the FS representations above. A frame macros consists of five primary parts:

- a frame id
- the orthography of the predicate under consideration,
- a category label with a list of morphosyntactic features
- a sequence of slots
- a list of precedence relations between integer indexes indicating the relative positioning of slots or slot realisations.

$$\left(\begin{array}{l}\textbf{frame}{:}1\\ \text{like,}\\ \text{v[bs,...],}\\ slot_1\\ \vdots\\ slot_n\\ [N_{ii} > N_{jj}, ..., N_{in} > N_{jn}]\end{array}\right)$$

Slots are represented as a disjunctions of slot realisations associated with a slot id:

$$\left\{\begin{array}{l}\textbf{slot}{:}1\\ reals_1\\ \vdots\\ reals_n\end{array}\right\}$$

Each slot realisation consists of

- a slot realisation id, and
- a list of values for the following attributes:
  - category and morphosyntactic features
  - grammatical function and predicative value
  - obligatoriness
  - control
  - obviation
  - order with respect to the head (*in_order*)
  - thematic role index and features
  - semantic class

For example:

**realz**:1
np[nom]
subj[npred]
no_opt
control(1-1-1)
obv(1-1-2)
post
Θ1[ cause,change]
[sem_class]

Values are omitted when not relevant or unspecified. Here is a frame example with reference to the English verb ‘like’ (some extra semantic values are introduced for expository purposes) :

**frame**:1
like,
v[bas],
**slot**:1
**realz**:1
np[nom]
subj[npred]
no_opt
$\Theta$1[ emotive,be]
[sentient]
**slot**:2
**realz**:1
np[nom]
comp[npred]
no_opt
$\Theta$21[ emotive,source]
[sem_class]
**realz**:2
vp[ger|inf]
comp[pred]
no_opt
control(1-1-1)
$\Theta$22[ emotive,source]
[event]

A frame list is represented as a structure consisting of a list of frames and a list of relations between frame-slot(-realisation) index couples (triples), as shown in the example below where the inchoative and causative versions of the Italian verb 'affondare' are related:

**frame**:1
affondare,
v[avere],
**slot**:1
**realz**:1
np[nom]
subj[npred]
no_opt
$\Theta$1[ move,cause]
[sem_class]
**slot**:2
**realz**:1
np[nom]
comp[npred]
no_opt
$\Theta$21[ move,change]
[sem_class]
,
**frame**:2
affondare,
v[essere],
**slot**:1
**realz**:1
np[nom]
subj[npred]
no_opt
$\Theta$1[ move,change]
[sem_class]

[rel(1-2-1,2-1-1)]

## A.2 Spanish examples

### Basar (*base*)

Transitive verb with bound prepositional complement. It may be pronominalised reducing the second argument (basarse). However there is not a direct relation between the arguments of one realisation and the other — contrary to decausativisation of verbs like *cerrar*, — see below.

(157) a. El presidente basó su discurso en la productividad
*The president based his speech on productivity*

b. No sé en qué se basa para afirmar esto
*I don't know on what he bases himself to say that*

c. La productividad se basa en la buena gestión
*Productivity is based on good management*

```
[ ( frame:1                      ( frame:2                    ]
[   basar,                         basar,                     ]
[   v[inf],                        v[inf],                    ]
[   { slot:1                       { slot:1                   ]
[     realz:1                        realz:1                  ]
[       np[nom]                        np[nom]                ]
[       subj[ ]                        subj[ ]                ]
[       no_opt                         no_opt }               ]
[       [anim] }                 , { slot:2                   ]
[   { slot:2                         realz:1                  ]
[     realz:1                          np[LEX:"se"]           ]
[       np[acc,3]                      comp[ ]                ]
[       dcomp[ ]                       no_opt }               ]
[       no_opt                     { slot:3                   ]
[       [no-anim] }                  realz:1                  ]
[   { slot:3                           pp[INTROD:"en"]        ]
[     realz:1                          icomp[ ]               ]
[       pp[INTROD:"en"]                no_opt } )             ]
[       icomp[ ]                                              ]
[       no_opt } )                                            ]
[ [rel(1-1-1,2-1-1),rel(1-2-1,2-2-1),rel(1-3-1,2-3-1)]        ]
```

### Caducar (*expire*)

Intransitive verb

(158) a. Su carnet de conducir ha caducado
*His driving license has expired*

```
( frame:1
  caducar,
  v[inf],
  { slot:1
    realz:1
      np[nom]
      subj[ ]
      no_opt
      [no-anim] } )
```

### Cerrar (*close*)

Transitive verb that allows decausativisation by means of the reflexive pronoun ‘se’.

(159) a. Juan cerró la puerta
*John closed the door*

b. La puerta se cerró
*The door closed*

[ ( **frame**:1
cerrar,
v[inf],
{ **slot**:1
**realz**:1
np[nom]
subj[ ]
no_opt
[anim] }
{ **slot**:2
**realz**:1
np[acc]
dcomp[ ]
no_opt
[no-anim] } ) ,
( **frame**:2
cerrar,
v[inf],
{ **slot**:1
**realz**:1
np[nom]
subj[ ]
no_opt
[no-anim] }
{ **slot**:2
**realz**:1
np[LEX:“se”]
comp[ ]
no_opt } )
[rel(1-2-1,2-1-1)] ]

### Cocinar (*cook*)

Transitive passivisable verb with object deletion. In this example we will represent passivisation as alternating frames with related slots. In other examples, the possibility to passivise of the transitive frame is dealt with by means of the morphosyntactic feature *passivisable* encoded in the direct complement.

(160) a. Hoy cocina mi marido
*Today my husband does the cooking*

b. Cocino el arroz a fuego lento
*I cook the rice on a low flame*

c. Este plato ha sido cocinado sin sal
*This dish has been cooked without any salt*

**frame**:1
cocinar,
v[inf],
**slot**:1
**realz**:1
np[nom]
subj[ ]
no_opt
[anim]
**slot**:2
**realz**:1
np[acc]
dcomp[ ]
opt
[no-anim]
,
**frame**:2
cocinado,
v[part],
**slot**:1
**realz**:1
np[nom]
subj[ ]
no_opt
[no-anim]
**slot**:2
**realz**:1
np[INTROD:"por"]
by-obj[ ]
opt
[anim]
[rel(1-1-1,2-2-1),rel(1-2-1,2-1-1)]

## Coincidir (*coincide*)

Prepositional verb with bound pcomp introduced by 'con' (*with*) which may be omitted provided that the subject is plural (coordinated or not).

(161) a. Tu opinión no coincide con la mía
*Your opinion doesn't coincide with mine*

b. Tu opinión y la mía no coinciden
*Your opinion and mine don't coincide*

c. Nuestras opiniones no coinciden
*Our opinions don't coincide*

**frame**:1
coincidir,
v[inf],
**slot**:1
**realz**:1
np[nom]
subj[ ]
no_opt
**slot**:2
**realz**:1
np[INTROD:"con"]
icomp[ ]
no_opt
,
**frame**:2
coincidir,
v[inf],
**slot**:1
**realz**:1
np[nom,pl]
subj[ ]
no_opt
[rel(1-1-1,2-1-1),rel(1-2-1,2-1-1)]

## Dar (*give*)

Ditransitive non-passivisable verb with optional indirect object.

(162) a. Juan dió un libro a María
*John gave a book to Mary*

b. El profesor dió los resultados de los exámenes
*The teacher gave the results of the exams*

**frame**:1
dar,
v[inf],
{ **slot**:1
**realz**:1
np[nom]
subj[ ]
no_opt
[anim] }
{ **slot**:2
**realz**:1
np[acc]
dcomp[ ]
no_opt
[no-anim] }
{ **slot**:3
**realz**:1
np[dat,INTROD:“a”]
icomp[ ]
opt
[anim] }

### Gustar (*like*)

Indirect object control verb.

(163) a. A Juan le gusta viajar en avión
*John likes travelling by plane*

b. Me gusta que seas responsable
*I like that you are responsible*

c. Me gusta el vino de La Rioja
*I like the wine from La Rioja*

**frame**:1
gustar,
v[inf],

**slot**:1
**realz**:1
np[nom]
subj[ ]
no_opt
post
**realz**:2
s[fin,sub]
subj[ ]
no_opt
post
**realz**:3
s[inf]
subj[ ]
no_opt
post
control(1-2-1)

**slot**:2
**realz**:1
np[dat,INTROD:“a”]
icomp[ ]
no_opt
pre
[anim]

### Ir (*go*)

Directional verb with optional weakly bound pcomps. Optionality follows this restriction: whenever the third argument (goal) is not present the second (origin) cannot be there either.

(164) a. Hoy iré andando.
*Today I’ll walk*

b. Juan va hacia su casa.
*John is going towards his house*

c. María va de casa a la escuela a pie.
*Mary goes from home to school by foot*

```
[ ( frame:1
    ir,
    v[inf],
    { slot:1
      realz:1
        np[nom]
        subj[ ]
        no_opt }
    { slot:2
      realz:1
        pp[INTROD:“de”|desde]
        icomp[ ]
        opt }
    { slot:3
      realz:1
        pp[INTROD:“a”|“hacia”|“hasta”]
        icomp[ ]
        no_opt } ) ,
  ( frame:2
    ir,
    v[inf],
    { slot:1
      realz:1
        np[nom]
        subj[ ]
        no_opt } )
  [rel(1-1-1,2-1-1)] ]
```

## Llevar (*carry*)

Movement transitive verb with passivisable direct complement. May have either two directional complements (origin, goal), with the first being optional, or an animate indirect object. Since one excludes the others, we have chosen to have two different frames, related by a common frame set.

(165) a. El tren llevaba carbón de Manchester a Londres.
*The train carried charcoal from Manchester to London*

b. Juan llevó a su hermana al colegio.
*John brought his sister to school*

c. Juan llevó un regalo a su hermana
*John brought a present to his sister*

```
[ ( frame:1                          ( frame:2
    llevar,                            llevar,
    v[inf],                            v[inf],
    { slot:1                           { slot:1
      realz:1                            realz:1
        np[nom]                            np[nom]
        subj[ ]                            subj[ ]
        no_opt }                           no_opt }
    { slot:2                           { slot:2
      realz:1                            realz:1
        np[acc,passivisable]               np[acc,passivisable]
        dcomp[ ]                           dcomp[ ]
        no_opt }                  ,        no_opt }
    { slot:3                           { slot:3
      realz:1                            realz:1
        np[dat,INTROD:"a"]                 pp[INTROD:"de"|"desde"]
        icomp[ ]                           icomp[ ]
        opt } )                            opt }
                                       { slot:4
                                         realz:1
                                           pp[INTROD:"a"|"hacia"|"hasta"]
                                           icomp[ ]
                                           no_opt } )
  [rel(1-1-1,2-1-1),rel(1-2-1,2-2-1),rel(1-3-1,2-4-1)] ]
```

## **Mejorar (*improve*)**

Ergative verb.

(166) a. El nuevo gobierno ha mejorado la situación económica
*The new Government has improved the economical situation*

b. La situación económica ha mejorado
*The economical situation has improved*

```
[ ( frame:1
    mejorar,
    v[inf],                         ( frame:2
    { slot:1                          mejorar,
      realz:1                         v[inf],
        np[nom]                       { slot:1
        subj[ ]                         realz:1
        no_opt }                  ,       np[nom]
    { slot:2                              subj[ ]
      realz:1                             no_opt
        np[acc,passivisable]              [no-anim] } )
        dcomp[ ]
        no_opt
        [no-anim] } )
  [rel(1-2-1,2-1-1)] ]
```

## **Ocurrir (*happen*)**

Intransitive verb with optional indirect object and possibly sentential subject.

(167) a. Me ocurrio un percance el otro dia
*A misfortune happened to me the other day*

b. Ocurre a menudo que nos vemos al salir de clase
*It happens often that we meet after the courses*

```
( frame:1
  ocurrir,
  v[inf],
  { slot:1
    realz:1
      np[nom,3]
      subj[ ]
      no_opt
      post
      [no-anim]
    realz:2
      s[fin,ind]
      subj[ ]
      no_opt
      post }
  { slot:2
    realz:1
      np[dat,INTROD:"a"]
      icomp[ ]
      opt
      [anim] } )
```

## Oir (*hear*)

Perception verb. Bears an optional direct complement NP which may be passivised. It may also bear an infinitival clause controlled by the NP object, if it is present. The functional status of this sentential argument is not clear. Following LFG, we call it a predicative complement subject to control (XCOMP). This complement may also appear without the direct object when the verb is taken as an impersonal.

(168) a. Oigo a mis sobrinos.
*I hear my nephews*

b. Oigo a mis sobrinos jugar.
*I hear my nephews playing*

c. Oigo llover
*I hear the rain*

d. Oigo cantar
*I hear singing (I hear someone singing)*

**frame**:1
oir,
v[inf],
{ **slot**:1
**realz**:1
np[nom]
subj[ ]
no_opt
[anim] }
{ **slot**:2
**realz**:1
np[acc,passivisable]
dcomp[ ]
opt }
{ **slot**:3
**realz**:1
s[inf]
comp[pred]
opt
control(1-2-1) }

### Parecer (*seem*)

Subject raising verb.

(169) a. Este libro parece interesante.
*This book seems interesting*

b. Juan parece estar en buena forma.
*John seems to be in good shape*

c. Parece que Juan está en buena forma.
*It seems that John is in a good shape*

**frame**:1
parecer,
v[inf],
**slot**:1
**realz**:1
np[nom]
subj[ ]
no_opt
**slot**:2
**realz**:1
np[nom]
comp[pred]
no_opt
**realz**:2
ap[ ]
comp[pred]
no_opt
**realz**:3
pp[ ]
comp[pred]
no_opt
**realz**:4
vp[inf]
comp[pred]
no_opt
subj_raising(1-1-1)
,
**frame**:2
parecer,
v[inf],
**slot**:1
**realz**:1
s[fin,sub]
subj[ ]
no_opt
post
[rel(1-2,2-1)]

### Pensar (*think*)

Subject control verb. The second argument is optional and may have several realisations, with slight differences in meaning.

(170) a. Pienso luego existo
*I think, therefore I am*

b. He pensado un plan para solucionar el problema
*I've come up with a plan to solve our problem*

c. Pienso que será bastante fácil
*I think that it will be quite easy*

d. Juan piensa comprarse un coche
*John is thinking in buying a new car*)

e. Pienso a menudo en mis padres
*I think often of my fathers*

f. No pensé en decirte que apagaras la luz
*I didn't remember to tell you to turn the lights off*

**frame**:1
pensar,
v[inf],
**slot**:1
**realz**:1
np[nom]
subj[ ]
no_opt
[anim]
**slot**:2
**realz**:1
np[acc]
dcomp[ ]
opt
[no-anim]
**realz**:2
s[fin,ind]
dcomp[ ]
opt
**realz**:3
s[inf]
dcomp[ ]
opt
control(1-1-1)
**realz**:4
pp[INTROD:"en"]
icomp[ ]
opt

### Permitir (*permit*)

Ditransitive control verb, with the indirect object controlling the infinitive clause, if present.

(171) a. El profesor no permitió a María abandonar el aula.
*The teacher didn't permit Mary to leave the room*

b. El profesor permitió a María que abandonara el aula.
*The teacher permited Mary to leave the room (that she left the room)*

c. El gobierno no permite la importación de coches.
*The Govern doesn't allow the importation of foreign cars*

**frame**:1
permitir,
v[inf],

**slot**:1
**realz**:1
np[nom]
subj[ ]
no_opt
**realz**:2
s[inf]
subj[ ]
no_opt

**slot**:2
**realz**:1
np[acc,3,passivisable]
dcomp[ ]
no_opt
[no-anim]
**realz**:2
s[inf,passivisable]
dcomp[ ]
no_opt
control(1-3-1)
**realz**:3
s[fin,sub]
dcomp[ ]
no_opt

**slot**:3
**realz**:1
np[dat,INTROD:"a"]
icomp[ ]
opt
[anim]

### Poder (*can, may*)

Modal subject raising verb.

(172) a. La situación puede cambiar.
*The situation may change*

b. Puede que la situación cambie.
*It may be that the situation changes*

$$\left[\begin{array}{l}
\left(\begin{array}{l}\textbf{frame}{:}1\\ \text{poder,}\\ \text{v[inf],}\\ \left\{\begin{array}{l}\textbf{slot}{:}1\\ \textbf{realz}{:}1\\ \quad\text{np[nom]}\\ \quad\text{subj[ ]}\\ \quad\text{no_opt}\end{array}\right\}\\ \left\{\begin{array}{l}\textbf{slot}{:}2\\ \textbf{realz}{:}1\\ \quad\text{vp[inf]}\\ \quad\text{dcomp[ ]}\\ \quad\text{no_opt}\\ \quad\text{subj_raising(1-1-1)}\end{array}\right\}\end{array}\right), \left(\begin{array}{l}\textbf{frame}{:}2\\ \text{poder,}\\ \text{v[inf],}\\ \left\{\begin{array}{l}\textbf{slot}{:}1\\ \textbf{realz}{:}1\\ \quad\text{s[fin,sub]}\\ \quad\text{subj[ ]}\\ \quad\text{no_opt}\\ \quad\text{post}\end{array}\right\}\end{array}\right)\\ \text{[rel(1-2-1,2-1-1)]}\end{array}\right]$$

### Ser (*be*)

Copular verb.

(173) a. Juan es mi profesor.
*John is my teacher*

b. La comida fue excelente.
*The meal was excellent*

c. La fiesta será en mi casa.
*The party will be in my home*

d. Es una lástima que no vengas.
*It is a pity that you don't come*

**frame**:1
ser,
v[inf],

**slot**:1
**realz**:1
np[nom]
subj[ ]
no_opt
**realz**:2
s[inf]
subj[ ]
no_opt
**realz**:3
s[fin,sub]
subj[ ]
no_opt

**slot**:2
**realz**:1
np[nom]
comp[pred]
no_opt
**realz**:2
ap[ ]
comp[pred]
no_opt
**realz**:3
pp[ ]
comp[pred]
no_opt
**realz**:4
advp[ ]
comp[pred]
no_opt

## A.3 German examples

### vergehen (*pass* )

Intransitive verb

(174) a. Der Abend verging.
*The evening went by.*

```
( frame:1
  vergehen,
  v[inf],
  { slot:1
    realz:1
      np[nom]
      subj[ ]
      no_opt
      [time] } )
```

### oeffnen (*open*)

Transitive verb that allows decausativisation by means of the reflexive pronoun ‘sich’.

(175) a. Hans oeffnete die Tuer
*John opened the door*

b. Die Tuer oeffnete sich
*The door opened*

```
[ ( frame:1                          ,  ( frame:2
    oeffnen,                              oeffnen,
    v[inf],                               v[inf],
    { slot:1                              { slot:1
      realz:1                               realz:1
        np[nom]                               np[nom]
        subj[ ]                               subj[ ]
        no_opt                                no_opt
        [anim] }                              [no-anim] }
    { slot:2                              { slot:2
      realz:1                               realz:1
        np[acc,3,passivisable]                np[LEX:“sich”]
        dcomp[ ]                              comp[ ]
        no_opt                                no_opt } )
        [no-anim] } )
  [rel(1-2,2-1)]                                                ]
```

### sich treffen (*meet*)

Prepositional verb with bound pcomp introduced by ‘mit’ (*with*) which may be omitted provided that the subject is plural (coordinated or not).

(176) a. Ich treffe mich mit meinen Freunden
*I meet my friends*

b. Meine Freunde und ich treffen sich
*My friends and I come together*

c. Meine Freunde treffen sich
*My friends come together*

```
[ ( frame:1                          ,  ( frame:2
    treffen,                              treffen,
    v[inf],                               v[inf],
    { slot:1                              { slot:1          } { slot:2              }
      realz:1                               realz:1             realz:1
        np[nom]                               np[nom,pl]          np[LEX:"sich"]
        subj[ ]                               subj[ ]             comp[ ]
        no_opt }                              no_opt              no_opt            )
    { slot:2
      realz:1
        np[LEX:"sich"]
        comp[ ]
        no_opt }
    { slot:3
      realz:1
        np[INTROD:"mit"]
        pcomp[ ]
        no_opt } )
  [rel(1-1,2-1),rel(1-3,2-1)]                                                          ]
```

## geben (*give*)

Verb with dative and accusative objects.

(177) a. John gab seiner Freundin das Buch
*John gave the book to his girl friend*

```
( frame:1
  geben,
  v[inf],
  { slot:1
    realz:1
      np[nom]
      subj[ ]
      no_opt
      [anim] }
  { slot:2
    realz:1
      np[acc]
      dcomp[ ]
      no_opt
      [no-anim] }
  { slot:3
    realz:1
      np[dat]
      icomp[ ]
      opt
      [anim] } )
```

### aergern (*irritate*)

Indirect object control verb.

(178) a. Der letzte zu sein, aergert mich
*Being the last one irritates me*

b. Dass es regnet, aergert ihn
*That it rains annoys him*

c. Das schlechte Wetter aergert ihn
*The bad weather annoys him*

**frame**:1
aergern,
v[inf],
  **slot**:1
  **realz**:1
    np[nom]
    subj[ ]
    no_opt
  **realz**:2
    s[fin,sub]
    subj[ ]
    no_opt
  **realz**:3
    s[inf]
    subj[ ]
    no_opt
    control(1-2-1)
  **slot**:2
  **realz**:1
    np[acc]
    icomp[ ]
    no_opt
    [anim]

### kochen (*cook*)

Transitive verb with optional object amenable to passivasation.

(179) a. Heute kocht mein Mann
*Today my husband does the cooking*

b. Ich koche den Reis auf niedriger Flamme
*I cook the rice on a low flame*

c. Die Kartoffeln haben zu lange gekocht
*The potatoes have been cooked too long*

```
[ ( frame:1              ( frame:2
    kochen,                kochen,
    v[inf],                v[part],
    { slot:1               { slot:1
      realz:1                realz:1
        np[nom]                np[nom]
        subj[ ]                subj[ ]
        no_opt                 no_opt
        [anim] }               [no-anim] }
    { slot:2           ,   { slot:2
      realz:1                realz:1
        np[acc]                np[INTROD:"bei"]
        dcomp[ ]               comp[ ]
        opt                    opt
        [no-anim] } )          [anim] } )
  [rel(1-1,2-2),rel(1-2,2-1)]                       ]
```

**hoeren** (*hear*)

Perception verb. Bears an optional direct complement NP which may be passivised. It may also bear an infinitival clause controlled by the NP object, if it is present.

(180) a. Ich hoere die Glocke.
*I hear the bell*

b. Ich hoere die Engel singen.
*I hear the angels singing*

```
( frame:1
  hoeren,
  v[inf],
  { slot:1
    realz:1
      np[nom]
      subj[ ]
      no_opt
      [anim] }
  { slot:2
    realz:1
      np[acc,passivisable]
      dcomp[ ]
      opt }
  { slot:3
    realz:1
      s[inf]
      comp[pred]
      opt
      control(1-2-1) } )
```

## A.4 Italian examples

### Piovere (*rain*)

Weather verbs obligatorily require no complements; i.e. their slot list is empty.

(181) a. Oggi piove
*Today it rains*

$$\left(\begin{array}{l}\textbf{frame}{:}1\\ \text{piovere,}\\ \text{v[inf, avere|essere],}\\ \left\{\begin{array}{l}\textbf{slot}{:}0\\ nil\end{array}\right\}\end{array}\right)$$

### Ridere (*laugh*)

Intransitive verb.

(182) a. Ha riso nervosamente
*(S)he laughed nervously*

$$\left(\begin{array}{l}\textbf{frame}{:}1\\ \text{ridere,}\\ \text{v[inf, avere],}\\ \left\{\begin{array}{l}\textbf{slot}{:}1\\ \textbf{realz}{:}1\\ \quad\text{np[nom]}\\ \quad\text{subj[ ]}\\ \quad\text{[human]}\end{array}\right\}\end{array}\right)$$

### Parlare (*talk*)

Verb with two prepositional complements introduced respectively by “di” (*about*) and “con” (*with*) which may both be omitted.

(183) a. Giovanni ha parlato di letteratura con Maria
*John talked about literature with Mary*

b. Giovanni ha parlato di letteratura
*John talked about literature*

c. Giovanni ha parlato con Maria
*John talked with Mary*

d. Giovanni ha parlato
*John talked*

**frame**:1
parlare,
v[inf, avere],
{ **slot**:1
**realz**:1
np[nom]
subj[ ]
[human] }
{ **slot**:2
**realz**:1
np[INTROD:“di”]
icomp[ ]
opt }
{ **slot**:3
**realz**:1
np[INTROD:“con”]
icomp[ ]
opt
[human] }

## **Avvalersi** (*avail oneself of*)

Intransitive verb in pronominal form with prepositional complement introduced by “di”.

(184) a. Giovanni si avvalse del mio aiuto
*John availed himself of my help*

**frame**:1
avvalersi,
v[inf, essere, pronominal],
{ **slot**:1
**realz**:1
np[nom]
subj[ ]
[human] }
{ **slot**:2
**realz**:1
np[INTROD:“di”]
icomp[ ] }

## **Mangiare** (*eat*)

Transitive verb with optional object.

(185) a. Giovanni ha mangiato pasta
*John ate pasta*

b. Giovanni ha mangiato
*John ate*

**frame**:1
mangiare,
v[inf, avere],
{ **slot**:1
**realz**:1
np[nom]
subj[ ]
[anim] }
{ **slot**:2
**realz**:1
np[acc]
dcomp[ ]
opt
[edible] }

### Pagare (*pay*)

Verb with two complements (a direct object and a prepositional complement introduced by “a” *to*) whose optionality is constrained by logical conditions, i.e. omission of the direct object requires omission of the indirect object too (note, incidentally, that the two objects can be omitted simultaneously).

(186) a. Giovanni ha pagato
*John paid*

b. Giovanni ha pagato il conto
*John paid the bill*

c. Giovanni ha pagato il conto a Maria
*John paid the bill to Mary*

d. *Giovanni ha pagato a Maria
*John paid to Mary*

[
( **frame**:1
pagare,
v[inf, avere],
{ **slot**:1
**realz**:1
np[nom]
subj[ ]
[human] }
{ **slot**:2
**realz**:1
np[ ]
dcomp[ ]
opt } ),
( **frame**:2
pagare,
v[inf, avere],
{ **slot**:1
**realz**:1
np[nom]
subj[ ]
[human] }
{ **slot**:2
**realz**:1
np[acc]
dcomp[ ]
no_opt }
{ **slot**:3
**realz**:1
np[dat, INTROD:“a”]
icomp[ ]
no_opt } )
[rel(1-1-1,2-1-1),rel(1-2-1,2-2-1)]
]

### Friggere (*fry*)

Transitive verb undergoing the causative-inchoative alternation with no morphosyntactic marking in the inchoative reading.

(187) a. Giovanni ha fritto le patate
*John fried potatoes*

b. Le patate friggono
*The potatoes fry*

$$\left[\begin{array}{l} \left(\begin{array}{l} \textbf{frame}{:}1 \\ \text{friggere,} \\ \text{v[inf, avere],} \\ \left\{\begin{array}{l} \textbf{slot}{:}1 \\ \textbf{realz}{:}1 \\ \quad \text{np[nom]} \\ \quad \text{subj[ ]} \\ \quad \text{[human]} \end{array}\right\} \\ \left\{\begin{array}{l} \textbf{slot}{:}2 \\ \textbf{realz}{:}1 \\ \quad \text{np[acc]} \\ \quad \text{dcomp[ ]} \\ \quad \text{[food]} \end{array}\right\} \end{array}\right), \left(\begin{array}{l} \textbf{frame}{:}2 \\ \text{friggere,} \\ \text{v[inf, avere],} \\ \left\{\begin{array}{l} \textbf{slot}{:}1 \\ \textbf{realz}{:}1 \\ \quad \text{np[nom]} \\ \quad \text{subj[ ]} \\ \quad \text{[food]} \end{array}\right\} \end{array}\right) \\ \text{[rel(1-2-1,2-1-1)]} \end{array}\right]$$

## Rompere (*break*)

Transitive verb undergoing the causative-inchoative alternation with morphosyntactic marking in the inchoative reading.

(188) a. Giovanni ha rotto il vetro
*John broke the window-pane*

b. Il vetro si ruppe
*The window-pane broke*

$$\left[\begin{array}{l} \left(\begin{array}{l} \textbf{frame}{:}1 \\ \text{rompere,} \\ \text{v[inf, avere],} \\ \left\{\begin{array}{l} \textbf{slot}{:}1 \\ \textbf{realz}{:}1 \\ \quad \text{np[nom]} \\ \quad \text{subj[ ]} \end{array}\right\} \\ \left\{\begin{array}{l} \textbf{slot}{:}2 \\ \textbf{realz}{:}1 \\ \quad \text{np[acc]} \\ \quad \text{dcomp[ ]} \\ \quad \text{no\_opt} \end{array}\right\} \end{array}\right), \left(\begin{array}{l} \textbf{frame}{:}2 \\ \text{rompere,} \\ \text{v[inf, essere, pronominal],} \\ \left\{\begin{array}{l} \textbf{slot}{:}1 \\ \textbf{realz}{:}1 \\ \quad \text{np[nom]} \\ \quad \text{subj[ ]} \end{array}\right\} \end{array}\right) \\ \text{[rel(1-2-1,2-1-1)]} \end{array}\right]$$

## Promettere (*promise*)

Subject control verb.

(189) a. Giovanni ha promesso a Maria di andare
*John promised Mary to go*

b. Giovanni ha promesso di andare
*John promised to go*

**frame**:1
promettere,
v[inf, avere],
- **slot**:1
  **realz**:1
  np[nom]
  subj[ ]
  [human]
- **slot**:2
  **realz**:1
  np[dat,INTROD:“a”]
  icomp[ ]
  opt
  [human]
- **slot**:3
  **realz**:1
  s[nfin, inf, INTROD:“di”]
  dcomp[ ]
  no_opt
  control(1-1-1)

### Informare (*inform*)

Subject control verb with two complements, a direct object and a prepositional or clausal complement.

(190) a. Giovanni ha informato Maria dello sciopero
*John informed Mary about the strike*

b. Giovanni ha informato Maria che lo sciopero era finito
*John informed Mary that the strike was over*

c. Giovanni ha informato Maria di aver vinto la gara
*John informed Mary that he won the competition*

**frame**:1
informare,
v[inf, avere],

**slot**:1
**realz**:1
np[nom]
subj[ ]
[human]

**slot**:2
**realz**:1
np[acc]
dcomp[ ]
no_opt
[human]

**slot**:3
**realz**:1
np[INTROD:“di”]
icomp[ ]
opt
**realz**:2
s[fin, INTROD:“che”]
icomp[ ]
opt
**realz**:3
s[ nfin, inf, INTROD:“di”]
icomp[ ]
opt
control(1-1-1)

## Persuadere (*persuade*)

Object control verb.

(191) a. Giovanni ha persuaso Maria ad andare
*John persuaded Mary to go*

b. Giovanni ha persuaso Maria
*John persuaded Mary*

**frame**:1
persuadere,
v[inf, avere],
**slot**:1
**realz**:1
np[nom]
subj[ ]
[human]
**slot**:2
**realz**:1
np[acc]
dcomp[ ]
no_opt
[human]
**slot**:3
**realz**:1
s[nfin, inf, INTROD:“a”]
icomp[ ]
opt
control(1-2-1)

## Ordinare (*order*)

Oblique control verb.

(192) a. Giovanni ha ordinato a Maria di andare
*John ordered Mary to go*

**frame**:1
ordinare,
v[inf, avere],
**slot**:1
**realz**:1
np[nom]
subj[ ]
[human]
**slot**:2
**realz**:1
np[dat, INTROD:“a”]
icomp[ ]
no_opt
[human]
**slot**:3
**realz**:1
s[nfin, inf, INTROD:“di”]
dcomp[ ]
no_opt
control(1-2-1)

## Sembrare (*seem*)

Subject raising verb.

(193) a. Questo libro sembra interessante
*This book looks interesting*

b. Questo libro sembra un capolavoro
*This book looks like a masterpiece*

c. Questo libro sembra fuori commercio
*This book seems to be out of print*

d. Questo libro sembra essere interessante
*This book seems to be interesting*

e. Sembra che questo libro sia interessante
*(It) seems that this book is interesting*

```
[ ( frame:1
    sembrare,
    v[inf, essere],
    { slot:1
      realz:1
        np[nom]
        subj[ ] }
    { slot:2
      realz:1
        np[acc]
        comp[pred]
        no_opt
      realz:2
        ap[ ]
        comp[pred]
        no_opt
      realz:3
        np[INTROD:prep]
        comp[pred]
        no_opt
      realz:4
        vp[inf]
        comp[pred]
        no_opt
        subj_raising(1-1-1) } ) ,
  ( frame:2
    sembrare,
    v[inf, essere],
    { slot:1
      realz:1
        s[fin,sub]
        subj[ ]
        no_opt
        post } )
  [ ] ]
```

## **Considerare** (*consider*)

Object raising verb.

(194) a. Considero la tua proposta una bella idea
*I consider your proposal a good idea*

b. Considero la tua proposta molto interessante
*I consider your proposal very interesting*

**frame**:1
considerare,
v[inf, avere],

- **slot**:1
  **realz**:1
  - np[nom]
  - subj[ ]
  - [human]
- **slot**:2
  **realz**:1
  - np[acc]
  - dcomp[ ]
  - no_opt
- **slot**:3
  **realz**:1
  - np[ ]
  - dcomp[pred]
  - no_opt

  **realz**:2
  - ap[ ]
  - dcomp[pred]
  - no_opt $(obj\_raising(2-1-1))$