# ET-10/51
# Deliverable 10

# Final Report

co-ordinated by

**John Sinclair**

**Geoff Barnbrook, Nicoletta Calzolari, Stefano Federici,
Martin Hoelter, Simonetta Montemagni, Carol Peters, John Sinclair**

**Sprachwissenschaftliches Institut**
**Ruhr-Universität Bochum**
**D-44780 Bochum**
**Federal Republic of Germany**

**For further information contact:**

Martin Hoelter

Sprachwissenschaftliches Institut
Ruhr-Universität Bochum
D-44780 Bochum
Germany

Phone: +49 234 700 2461
Fax: +49 234 7094 137

Internet: hoelter@linguistics.ruhr-uni-bochum.de

# Contents

# 1 Introduction

In May 1992 a new research project brought together the authors of this report. With the help and support of several other people and institutions, they worked steadily for two years, trying to improve the design and building of machine-usable lexicons, for automatic translation and many other applications.

The starting point was clear. Around 1989 Helmut Schnelle of the Ruhr-Universität Bochum became interested in the way in which words were defined in a new kind of dictionary called Cobuild. He thought that since they were couched in sentences of apparently ordinary English, and had distinctive and repetitive shapes according to their meanings, it should be possible to represent them in logical form by means of regular rules.

He shared this view with me on several occasions, and we began to see that there might be a powerful strand of research coming out of Helmut Schnelle's observation. No such venture had been foreseen when the defining style of the dictionary was worked out in the period 1984–6; at that time there was no suggestion that it could have any greater significance than to ease access to the definitions.

The origin of the full-sentence definition in Cobuild was developed from the study of spoken discourse that was 'seventies research in Birmingham; when planning a dictionary for non-native speakers it becomes obvious that the traditional style of definition is somewhat distant from their everyday language. There is a natural way in which people explain the meaning of words, and that is carefully reflected in the Cobuild defining style. What actually happened was that sometime in 1984 I accepted a challenge from my lexicographical colleagues, claiming that I could dispense with all the different type-faces, non-standard symbols, abbreviations, odd phraseology and tricks like the use of *etc.* to conceal an inability to specify something. I took a few draft entries which had been compiled in the traditional style and just rewrote them in ordinary English prose. There was a directness and freshness about this style which gradually won over the team, and in discussions over the following few months, the published style evolved. I was obliged to surrender some points, e.g. that the headword should be in bold and the examples in italics, but the final version was still seen as a remarkable innovation in lexicography.

Following the publication in 1987 of the first Cobuild Dictionary, I remained personally intrigued by the simplicity and flexibility of the style, and published a paper on its structure in 1990, following Hanks' (1987) account in *Looking Up*. Soon after that my colleague Geoff Barnbrook joined me in preliminary research into the possibilities of automating the analysis. We found a useful framework for pilot work in The Chamberlain Project, a joint venture of IBM and the University of Birmingham, and we gradually gained confidence in two hypotheses.

First of all, the automation appeared to be achievable by a computationally straightforward procedure, far less complex than was and is the norm for NLP parsers; here might be a genuine sublanguage, showing a radical simplification compared with the requirements of a general grammar. Further, we found that the analysis underlying the parser was unusual, revealing aspects of meaning that were not normally codified in grammars, but tended to be consigned to the grey area of inference. Perhaps the repetitive and restricted nature of the language of definition would highlight aspects of meaning that had not featured in general grammars - but were everyday usages in the language as a whole. (We were of course half expecting to find that the sublanguage was so specialised that it had developed unique structures and patterns, in the same way as traditional lexicography had done, but, remembering that the efforts of the compilers were devoted to rendering the meanings in ordinary English sentences, we did not think that these would be of great importance.)

When Helmut Schnelle and I began to plan this project, we invited the Istituto di Linguistica Computazionale of Pisa to join us because of their expertise in building lexicons using Typed Feature Structures. We answered a Call for Tender under the ET-10 scheme, the final round of activity in Eurotra, where new approaches were called for. What we proposed three years ago was certainly a new approach.

We suggested that we could first of all work out in Birmingham a fully automatic parser for the dictionary definitions; this we would pass on to our partners for further stages in formalisation. Bochum would recast the parsed text into a logical regimentation that would remove ambiguities and prepare the ground for a totally abstract formal treatment; Pisa would recast the parsed text according to the conventions of Typed Feature Structures.

The aim was to make the description so formal and general that it would be independent of the language in which the sentences were originally written. When a whole dictionary had been thus processed, it would be possible to claim that the lexicon of a language had been made explicit in terms which were ready for further processing by machine. The meaning of English would thus have been computerised. Then we could look ahead to the time when dictionaries like Cobuild would be available for other languages, and a similar route could be devised to process a lexicon in another language. The two lexicons, expressed in identical formalisms, could then be compared and from this exercise there would emerge a new and powerful tool for automatic translation.

The project would thus be an important feasibility study to see how far the process could be taken, using just one small dictionary of English, and only a few hundred words from that.

In detailed planning with the EC, we were pressed to adopt the new language formalism called ALEP (Advanced Language Engineering Platform), which was taking shape. It was anticipated that by the time we would need it, the grammatical resources of ALEP would be sufficient for our needs. The advantages of standardising on ALEP were obvious; our project was designed to fit into a broad sweep of technical innovation, including the other ET-10 projects. Having them all compatible with each other through ALEP would be a big step forward in a research area noted for difficulties of harmonisation.

To help with ALEP we were assigned consultants from the Centre for Language Technology in Copenhagen, particularly Dr. Annelise Bech and Anne Neville. They conducted two substantial seminars for us, familiarising us with ALEP and bringing us up to date with developments, and supported us thereafter. In the event, ALEP did not develop fast enough for us to create our versions of the parser under its formalism, but we kept it in sight throughout, and we believe that when ALEP is ready it will not be a major task to adapt our interface to whatever its requirements will be.

In another move towards standardisation, the CEC required all the ET-10 projects to use the same text on which to carry out all their experimentation. The chosen text was called "The ITU Corpus", a multilingual text on the topic of satellite communications. A small section of this was made available and used as a source for the test vocabulary. Towards the end of the project the ITU text was used again as a source of concordances to compare with the Bank of English, a large reference corpus of ordinary English held in Birmingham.

Because of the way in which this project was designed, the reference to the ITU corpus was a slight diversion. The Cobuild dictionaries are specifically non-technical, and the project was based on the smallest of them; so many words to do with telecommunications have no entry, and many others are defined only in non-technical senses. We supplied missing definitions ad hoc, using experienced lexicographers.

As the project took shape and began its work, some issues became clear. One was the state of ALEP, mentioned above; another was the nature of a common interface for the work of all three partners. This was resolved to be Head-driven Phrase Structure Grammar (HPSG), adapted by Bochum to meet the specific needs of the project. It was also a revelation to the team how rich the definitions were in inference and pragmatic information, particularly in the left-part. A person used to ordinary dictionaries might expect that the most interesting part of the linguistic structure would be towards the end, in the discriminators. But in Cobuild many of the surprises were in the early part of the definition, and this became the object of primary interest for Pisa.

The left-part of the definition statement is the part that is not found in traditional lexicography. For example, if a verb definition is to be put in a full sentence, the likelihood is that it will require a subject. That requirement is not present in the usual dictionaries; it gave Cobuild an opportunity to select a subject that reflected the usage of the verb in the corpus, its collocations and other meaning relations, e.g. in meaning 3 of *connect* the subject is *a telephone operator*, and in meaning 4 it is *a train*,

*plane or bus*. The same specificity is given for the object, and all other elements of structure where the evidence justifies it.

However, for many verbs there is such a wide range of possible subjects that it is impossible to go beyond basics such as human or non-human, often given in the dictionary as *someone* or *something*. Even here the English language allows a number of subtle options - instead of *someone* it is common to have *you* when the activity is something the user of the dictionary is likely to identify with, and *a person* or *someone* when it is unpleasant or unusual. All the normal syntactic possibilities are available, such as passivisation, and each gives an important slant on the definition.

It was not possible within the dimensions of the project to cover the whole vocabulary, even of one of the smallest dictionaries. The parser worked fast, but the transfer to the formalisms required a lot of detailed and individual study. A word list gradually took shape, covering those used in a number of pilot experiments, a set of words that are important in dictionary definitions, and a representative list from the ITU corpus. The target was 200, but the researchers were often tempted beyond, and the final list numbered 373. Most types of word and definition are included in this list, and about half of it is technical vocabulary and usage.

The progress meetings during the project were always interesting because of the discoveries that were reported, both about the language itself and the nature of definition. On the computing side, the definition parser took shape and over the period of the project became more and more flexible, until at the end it could cope with all but a tiny number of the unusual or complex definitions. The common interface also took shape, accommodating the output of the parser, and the steps needed to formalise the definitions were automated.

The design of the project had been that Bochum and Pisa would take two different routes to a final result which would be expressed in ALEP. What actually happened was that the parser output and the common interface led them to forms which were different in emphasis rather than structure. The proximity of the two made the meetings more, rather than less, interesting.

Towards the end of the project we considered the future possibilities arising from the work, and indicated some immediate uses, for example in practical lexicography. We also had an opportunity to compare technical and non-technical uses of the same words, where the dictionary differed from the ITU usage, and found some interesting results.

In our opinion, the potential of this research is immense. In general terms we have shown how the two main traditions of computational linguistics can marry and work together to mutual advantage; for the Birmingham work is entirely corpus-based (both the compilation of the original dictionary and the parsing strategy), while Bochum and Pisa are experienced NLP groups. Also in general we have shown the kind of adaptations that are necessary for a formal representation to cope with the kind of meanings that arise from a corpus description, which are not noticed in the usual building of lexicons.

In more practical terms, we have provided a blueprint by which a large formal lexicon can be devised, largely automatically. The parser can parse most definition sentences, and contains no lexicon itself. The parser output can be received in the common interface and can in principle be interpreted in terms of TFS and HPSG. We anticipate that thousands of words will follow one or other of the types that have been provided for and therefore can be made available for NLP by fully automatic process. Given the high cost of building lexicons at present, the work of this project could lead to a cheap method, and the advantages of a common core in several development projects should not be ignored.

We are thus confident that our work has given value, and that our own future research will benefit from the findings and the tools that have been developed in this project. We hope that others will find both the results and the tools of interest.

*John Sinclair*
*Co-ordinator*

**Acknowledgements**

The authors of this report would like to thank all their colleagues who helped in the project for their support. In particular, the Project Officer, Mr E. Valentini, was a source of encouragement and help, and took a keen personal interest in the work. Our Danish consultants gave us their expertise in ALEP as it evolved during the project period, and kept us in touch. In Birmingham, Dr. G. Allport worked substantially on the project and took responsibility for Birmingham's input to Deliverables 4 and 6. The output format of the definition parser was his suggestion, and it became the standard transfer medium for parsed data.

Special thanks go to Frank Wegmann and Rolf Wilkens for their continuous technical support and for supplying two invaluable TEX macro packages—`avm.tex` and `yamp.tex` —without which the complex information derived from the dictionary could hardly have been represented in this report.

We would like to thank HarperCollins, publishers, who were generous with supplies of the *Collins Cobuild Student's Dictionary*, both in published form and also in electronic form, and gave us permission to use the text as the starting- point of the research. This text represents the results of many person-years' work by corpus linguists and lexicographers, and unrestricted use of it was essential for us. The co-operation in Birmingham between the University and Cobuild is manifest in everything we do. We are also grateful to IBM (UK), who supported the early research, especially with the original 6150 work station.

# 2 Parsing Cobuild Entries

## 2.1 Introduction

The language of dictionary definition has been for centuries a highly specialised form, related to English but with substantial variations from normal writing.

> Having the character anciently ascribed to the planets, wandering; erratic; as, a *planetary* career
> To go in a gallop, as a horse ... Quadrupedal motion by a regular succession of leaps
>
> (*Practical Standard Dictionary.* London: Funk and Wagnall, 1925)

Some recent projects (e.g. ACQUILEX) have provided analyses of this kind of language to aid knowledge extraction from conventional dictionaries. The growing awareness of "reusability" as a concept in language technology gave strength to the task, and made it worthwhile for special extraction tools to be developed.

In contrast to all other dictionaries, the Cobuild definitions are written in ordinary English sentences; a tool that can parse these is a partial parser of English, a way of interpreting the structure of the sentences that establishes a systematic relationship with the other sentences of English. Reusability is further pursued in that the effort expended on the construction of the parser is a contribution to the larger task of parsing the whole language adequately.

From the outset Birmingham regarded the definition text as a set of sentences in a sublanguage. A sublanguage is an important concept in natural language processing; it is a variety of a language which has two special characteristics:

a. It relates to a homogeneous specialised area of human activity and hence communication.
b. It forms a coherent subset of the language as a whole.

It is assumed that by narrowing the subvariety, usually in a technical context, the actual structure of a language will simplify, and thus become more amenable to automatic processing. A sublanguage is thus defined simultaneously by internal and external criteria, but the internal criteria are crucial; if the sublanguage is not very substantially simpler than the whole language, then the narrowing of focus does not aid the parsing task.

It was clear on cursory examination of the definitions that not all the structures of English were involved. For example there were only declarative sentences, so no need to worry about interrogatives or imperatives. There was therefore no need to write or borrow a comprehensive parser for English.

Further study revealed that the restrictions on the syntax were very substantial, and a new fact emerged—the most efficient parser for the sublanguage was not necessarily a subset of the parser for the general language. So specialised were the definitions that a special grammar was written for them. At some points it overlaps with a general grammar, but particularly in the more abstract statements, the functions of the definitions take precedence over the superficial similarity to English as a whole.

It may be, of course, that this observation merely reflects the somewhat primitive state of automatic parsing at the present time. A thorough and comprehensive parse might well recognise a definition statement as different from other kinds of statement, and incorporate some or all of the sublanguage parser we have written. Certainly we would have been misguided to have made the initial assumption that a general language parser would be both adequate and adapted to our needs.

It would be inadequate because we were obliged to recognise several major structural features which are not available in current parsers, for example *matching*. In the definitions, certain words and phrases that occur in the earlier part of the definition statement are to be matched with later occurrences of predictable items. This is a criterion of well-formedness akin to pairs like *not only ... but also*, but unique to this sublanguage. In certain circumstances the matching does not take place and that circumstance affects the identification of the *definiendum* itself.

As to relevance, we find that the overall structure of most of the definition sentences can be expressed as:

*headword*-in-*cotext*, explanation

This is a structural statement that has not been offered before for ordinary English parsing. It is related to subject/predicate, to theme/rheme and to several other familiar parse categories, but it is truly none of them. If we had started with an existing parser we would have uncritically chosen a variety of clause and sentence structures that can realise this generalisation, but would not have been led to find or express it. The parser would have been much more complicated and much less adequate.

The methodology of extraction for Birmingham, then, is built into the design of the grammar that is implemented in the parser. For example, a number of headwords have restrictions on their usage which are realised by words or phrases placed immediately in front of the headword. An important subcategory of these items is a noun with a possessive affix, as in "a bird's **beak**". Two classes can then be prepared, namely those headwords that suffer such restrictions, and those nouns that can by occurring in that structure restrict the headword. It happens that in most cases the two words are uniquely associated with each other, so there is considerable doubt as to the value of identifying the classes, whereas in general grammar we would expect classes with much more freedom of the members to co-occur.

Structure and function coincide here, and in most other places in the grammar. This is a huge simplification compared with normal grammars. There is no need in the sublanguage grammar for two different sets of terms, one for the structures and one for the functions, because the flexibility to combine structural elements in different functions is not available.

Absolute and relative position are major determinants of structure/function. Common words like *is* and *of* do different things according to their place in the definition, even though their role, as seen by a conventional grammar, is the same. So again it would have been misleading to have begun with a conventional grammar; there are no warnings of likely variance of this kind.

In addition to the kinds of functions that are reasonably expected in grammar writing, the study of the definitions brought out several strands of meaning that are usually thought to be inferential rather than directly structural. For example when the headword is a verb, the definition structure may express a subject for that verb. The subject may belong to one of several classes, of which a small subset are interesting from a quasi-inferential point of view. These are *you* and *someone*. The choice of *you* indicates that the action of the verb is something considered normal in social behaviour; one would not be ashamed or embarrassed to engage in this activity. *Swim, think, forgive* are examples. But the choice of *someone* as subject indicates that the average person would prefer to create a distance between himself or herself and the activity; as if to say that such things are done but people like us do not do them. Verbs such as *burp, cheat, gloat* refer to actions that are prejudged as undesirable, and have *someone* as their subject.

The wording of the definitions is so carefully organised that distinctions like the above can be associated directly with structural choices. Perhaps more general grammars will one day find that a choice of this nature can be built in as part of a larger picture, and show that there is more overlap between the sublanguage grammar and the general grammar than can be claimed at the present time.

## 2.2 The grammar

### 2.2.1 Introduction

A grammar is an account of the way utterances in a language are organised to create meaning. In a specialised variety of a language the organisation is not expected to have all the features of the language as a whole, and it may have features which are not found in the language as a whole. Hence the grammar of a specialised variety may be quite distinct from the grammar of the whole language, and may not be a proper subset of it.

The language of whole-sentence dictionary definitions—as shown, in modern times, by the Cobuild dictionaries—is a specialised variety of current English. There are severe restrictions on it compared with general English, mostly seen in the higher units of language patterning. While most of the words and phrases in it are used in their normal senses and patterns, the range of sentence types, the functions of the sentences and the higher organisation of the discourse are markedly specialised.

The point of specialisation would not be obvious to a general grammar, especially a generative grammar. A generative grammar that was at all adequate would indeed generate all the sentences of this variety, and many others. It would also of course generate many more sentences, and the likely strategy for a grammar of the sublanguage would be that the rules of the general grammar would be reduced to prohibit other sentences being produced. Indeed, in the received wisdom about sublanguages, that claim is made.

The approach of the general grammarian, then, is primarily reductionist. In contrast, the approach adopted in this project, arising from a detailed study of the sublanguage, is to devise a unique grammar that stays very close to the functions of the sublanguage, and ignores received categories.

We believe that the grammar is largely successful—a true reflection of the way the sublanguage is organised to create meaning. Since the definition sentences are in ordinary English, the specialised grammar might help to improve general grammars. All the definition sentences in our everyday usage now have an alternative grammar. Either they can be treated like any other sentence, or they can be described by this grammar, which assumes that they are intended as definitions.

Experiment will tell us whether the sublanguage grammar is always superior to the general grammar, or whether there are some conditions where it is better to ignore the potential of some sentences as definitions. The likelihood is that such a specialised grammar will outperform a general grammar, and that raises some interesting questions for the future of grammars.

Following a successful description of the grammar of definitions, another project could tackle another coherent function class of sentences—perhaps another sublanguage such as newspaper horoscopes, or legal judgements. Again a grammar unique to the functions of the variety would be produced, and the specialised grammar would shed light on this new area. And so on. No doubt there would be problems of reconciliation among the set of grammars that would emerge, and some generalisations would be difficult to capture. But the resultant general grammar would be many times richer and more relevant than existing ones.

This project provides a parser that can handle whole-sentence definitions, so any such sentences occurring in the language at large can be routed through the present parser, which will automatically offer an interpretation that is true to their function. A battery of similar grammars can be envisaged, which among them might describe convincingly many if not most of the sentences in ordinary text.

At present the parser is only applied to genuine defining sentences; it is not equipped with the means of deciding which sentences are suitable for being parsed by it; also it is aided by the prior identification of the headword, so there are still a few steps to take in development of the parser as a piece of analytic software. For example if we pass the following bogus definition through the parser:

A **dog** is a damned nuisance,

it will report:

```
op-word          'A'
match
headword         'dog'
hinge            'is'
match            'a'
discriminator    'damned'
superordinate    'nuisance'
```

It could well happen that some sentences would be acceptable in more than one sublanguage, and thus parseable by several of the specialised parsers, either inappropriately, as above, or, in the case of a multifunctional sentence, quite satisfactorily. So the provision of a battery of specialised parsers would need to have routeing software as well to direct sentences into the correct parser or parsers.

The idea of differential grammars for different sentence functions is interestingly similar to a strategy is proposed by Gross (e.g. 1993) at another level of language description. He devises "local grammars" to deal with organised text that normal grammars do not handle, such as dates and addresses in correspondence. Local grammars might also (Gross, personal communication) handle unique structures like the *ne...pas* negative in French.

The general grammar would be seen as a structurally oriented grammar, capable of performing labelled bracketing on the sentences but not able to work out their functions. In conventionally asking general grammars to cover both structure and function with one set of categories we may be overloading them.

We are so accustomed to the apparatus of a conventional grammar that it is difficult at first to avoid assumptions that are not stated in the text. "Main" and "subordinate" clauses are not necessary categories except in the detailed analysis of the discriminators, and are replaced by more specialised and useful categories, such as "projection" and "left-part". The sublanguage is much more postitionally restricted than the general language, and this leads to major simplifications in the grammar.

### 2.2.2 The concepts of the grammar

This is a sentence grammar. No attempt is made to describe second or subsequent sentences in a definition statement. No connections are postulated between one sentence and another. Each sentence is a text in itself.

Superficially a sentence can be divided typographically, since at least one word is picked out in bold face. This is the *definiens* of traditional lexicography, the *headword* of modern usage.

The function of a sentence is to make explicit the meaning of its headword(s).

#### 2.2.2.1 First-level

A sentence is divided into two parts, called the *left-part* and the *right-part*. The left part contains one or more words that have been picked out in bold face as headword—that which is to be defined. (Occasionally the bold face word or words appears towards the end of a sentence, and in such cases the sentences are considered to be *reversed*. The "left-part", as conventionally named in the grammar, is physically displaced to the right, and the "right-part" occurs on the left.)

There are two principal relations between the left part and right part. One is that of *equivalence*. Essentially the two parts of the sentence are held to mean the same thing. (From equivalence arise the two powerful notions of *paraphrase* and *substitutability*. Paraphrase is defined as the replacement of a word by its definition, or vice versa. Substitutability is defined as a segment of text which stands in an equivalence relation with another.)

Part of the realisation of equivalence is in the syntactic feature of *matching*. Elements in the left-part prospect matching elements to be present in the right-part. The structure of the right-part depends partly on the matching or non-matching of these prospections.

The other relation between the left part and the right part of a sentence is of *explicitness*. The right part is held to be normally more explicit than the left part. This is not so in the case of simple *synonymy* (see below). English often uses a phrase in preference to a single word to express meanings which are very similar; for example the Dictionary contains the entry:

PHRASE
If a structure **gives way**, it collapses.

The lexicographer has assumed that *collapses* will be a word available to the user, so no further explanation is given; the phrase is disambiguated and glossed with a close synonym.

The boundary between the left-part and the right-part is normally a *hinge* or a comma. Sentences which do not divide into two parts are treated separately.

#### 2.2.2.2 The left-part

The left-part contains the headword. Before the headword there may be *cotext*, and after the headword, more cotext. The cotext occurring before the headword is called *cotext 1*, and the cotext that occurs after the headword is called *cotext 2*. The structure of the left-part, then, may be:

```
[cotext 1] headword [cotext 2]
```

(where square brackets indicate options).

The function of the cotexts is to express conditions on the occurrence of the headword in the sense being defined. Each cotext sets up one or more *matches* which prospect suitable elements in the right-part.

Before cotext 1 and the headword there may be an *op-word*, usually the articles *a* or *an*, or the infinitive marker *to*. The op-word sets up a match.

```
[op-word] [cotext 1] headword [cotext 2]
```

Before the op-word, cotext or headword there may be a hinge. The function of the hinge is to indicate the relationship between the left-part and the right-part. Hinges also express *inferences* (or, rather, meanings that are conventionally assumed to be inferences). In conventional grammar a hinge would turn the rest of the left-part into a subordinate clause structure, but this is not a formal requirement of the sublanguage grammar. The main hinges are *if* and *when*, in the left-part, and *is*, *means* at the beginning of the right-part.

```
[hinge] [op-word] [cotext 1] headword [cotext 2]
```

Occasionally there is another cotext, at the very beginning of the sentence. This is called *cotext 0*. Its function is to express very broad and general restrictions on the sense of the headword.

```
[cotext 0] [hinge] [op-word] [cotext 1] headword [cotext 2]
```

The left-part of some definition statements contains a further elaboration called a *projection*. A projection is realised by the occurrence of a reporting structure after the hinge. In conventional grammar a projection would turn the rest of the left- part into a reported structure, but this is not a formal requirement of the sublanguage grammar.

```
[cotext 0] [hinge] [projection] [op-word] [cotext 1] headword [cotext 2]
```

#### 2.2.2.3 The right-part

The function of the right part is to make explicit the meaning of the headword by means of one of various techniques. The most common technique is that of analysis of the "genus-species" kind. Hence the right-part normally consists of a *superordinate* and at least one *discriminator*. These are realisations of explicitness, since they are prima facie substitutes for the headword. The terms are reciprocal, since a superordinate requires a discriminator to express its relation to the headword, and a discriminator discriminates among hyponyms.

```
superordinate discriminator
```

Occasionally there is no contrast between superordinate and discriminator in the right-part, and even at times just a single unmatched word. This is called a *synonym*. A synonym has the feature of substitutability but not explicitness. If there is more than a single word it is called a *synonymic phrase*, to indicate the different technique of definition. A synonymic phrase has the feature of explicitness, and the usual technique of definition is by concatenating words that, taken together, are the equivalent in meaning to the headword.

```
(superordinate discriminator) / (synonymic phrase) / synonym
```

(where the slash indicates options, and curved brackets enclose single choices.)

The right-part may have a hinge at the beginning:

```
[hinge] (superordinate discriminator) / synonymic phrase / synonym
```

Also in the right-part there will be the matching elements to those prospected in the left-part. Structurally, the matching allows the main components of the definition to be identified, as those not matched, and therefore not concerning the restrictions on the use of the sense. The places of occurrence of the matching elements vary with other choices in the phrasing and cannot be generalised; hence the notation as below:

```
{matches} [hinge] (superordinate discriminator) / synonymic phrase / synonym
```

(where curly brackets state elements of structure without fixed position).

#### 2.2.2.4 Elements of structure

`op-word` — *to*, *the*, *a* or *an*

`cotext` — words in the left-part which are not hinge or headword. They are serially numbered as follows:

| | |
|---|---|
| `cotext 0` | at the very beginning, before the hinge or op-word |
| `cotext 1` | between hinge/op-word and headword |
| `cotext 2` | after (the first part of) the headword |
| `cotext 3,` | etc. between and after subsequent parts of the headword |

`discriminator` — a stretch of text which is grammatically subordinate to the superordinate and expresses some distinguishing feature of a headword.

`headword` — a word or words which is typographically distinct from the other words of the definition

`hinge` — in the left-part: *if* *when*
in the right-part: *is* *are* *were* *means* *consists of*

`match` — a word or phrase in the left-part which prospects a reciprocal word or phrase in the right-part, according to the detailed lists below.

`projection` phrases involving a reporting structure, which are matched

`superordinate`

a. simple:
a single word, with matches, expressing the "genus" to which the headword belongs.

b. complex:
a phrase which is analysable into semantic strata, such that one element of it is the "genus" word and the rest is an appropriate environment for that word, usually involving a very general superordinate such as *piece, item.*

#### 2.2.2.5 Matching

Certain words and phrases are identified in the left-part as potential matches, and the right-part is searched for the items which would confirm the match. The principal matches are:

| | | |
|---|---|---|
| a/an | → | a/an |
| you | → | you |
| someone | → | they, them |
| a person | → | them |
| something | → | it |
| things | → | they |
| one thing | → | it |
| another | → | it |

In the cotexts, matches are assigned according to the local syntax; hence *a place, a machine* will be matched by *it, a person* etc., and plurals such as *cards* will be matched by *they, them*, etc.

One of the main rules of well-formedness is that all the op-words and cotexts must be matched in the right-part. Given the function of the right-part, there is no place for leftover cotext. If on the first pass there are some words or phrases that are not matched, then they are considered to be part of the headword, and the definition is reparsed.

This feature finally gives structural status to the headword. At the beginning of the analysis, the headword is superficially identified by the bold face type that is used in the dictionary. This is convenient, if not necessarily consistent with structural categorisation. In the vast majority of cases, the headword is confirmed by analysis as the word or phrase in bold type, but there are a few where the *definiendum* is in fact rather more than what the lexiographers have picked out. Usually the headword is extended by an adjacent word, a preposition or a very common verb. Because the left-part and the right-part are aligned with each other in the grammar, these extensions will show up as non-matches, and must then be incorporated in the headword, with consequent changes to the right-part.

It may not be coincidental that in translations of the definitions, being done in quite separate projects, the same phenomena are reported—that the definiendum has to be extended in a number of cases in order to achieve a sensible translation. In general the growing recognition of the importance of phrases in current English is undermining some of the very basic assumptions of lexicography and lexicon-building—that meaning inheres largely in the word.

## 2.3 The parser

### 2.3.1 Development

The parser was developed on the basis that the definitions in the Cobuild dictionary range form a sub-language which is sufficiently restricted in both lexis and syntax to allow automatic analysis. A taxonomy of definition patterns was constructed (as described in Deliverables 1 and 2) and parsing

strategies were devised to deal with each suitable pattern. The dictionary used for the development of the parser was the *Collins Cobuild Student's Dictionary*, containing 31418 pieces of text labelled as definitions. The definition language used in this dictionary is less complex than that used in other larger editions, and provided the best starting-point for exploration and development of an automatic method for extracting semantic and syntactic information from the definitions. The final version of the software takes definition texts extracted automatically from the machine readable text of the *Student's Dictionary* and produces an analysis capable of direct input to the software used by Pisa and Bochum.

### 2.3.2 Definition input

#### 2.3.2.1 Extraction of data from the dictionary text

The machine readable version of the *Student's Dictionary* contains much more information than is needed for the ET-10 analysis. As an example, the full entry for **abide**, including the field codes which provide typesetting information, is:

```
[EB]
[LB]
[HW]abide
[PR]/%eb*a*!id/,
[IF]abides, abiding, abided.
[LE]
[MB]
[MM]1
[FB]
[GR]PHRASE
[DT]If you [FH]can't abide [DC]something, you dislike it very much.
[XB]
[XX]He likes you but he can't abide Dennis.
[XE]
[FE]
[ME]
[MB]
[MM]2
[GR]VB
[DT]If something [HH]abides, [DC]it continues to happen or exist for a long
time.
[ME]
[CB]
[VB]
[VW]abide by.
[GR]PHR VB
[MB]
[DT]If you [HH]abide by [DC]a law, agreement, or decision, you do what it says.
[XB]
[XX]Both parties must agree to abide by the court's decision.
[XE]
[ME]
[VE]
[CE]
[EE]
```

The information needed for the analysis is:

a. the definition text;
b. the sense number;
c. the grammar note;
d. a sequence number representing the position in the dictionary of the selected sense; and
e. the forms of the lemma of the headword.

As can be seen from the full dictionary text, most of this information is available in different places within the set of entries for the headword. A set of simple extraction and editing programs was written (using the awk programming language and the tr and sed utilities) to collect this information and to convert the various dictionary field delimiters (such as [HH], [DC] etc.) to the uniform " | " field separator. This greatly facilitated later processing, but did not in itself carry out any of the necessary analysis. The entries for **abide** in the file used for initial input to the parsing software were:

```
if you |can't abide |something, you dislike it very much. |1 |phrase |25 |abide
abides abiding abided
if something |abides, |it continues to happen or exist for a long
time. |2 |vb |26 |abide abides abiding abided
if you |abide by |a law, agreement, or decision, you do what it says. |0 |phr
vb |27 |abide abides abiding abided
```

The only pieces of information contained in these entries which are not present in the original dictionary are the definition number, calculated by the extraction program to facilitate reference to individual definition texts, and the zero sense number assigned to items for which no sense number exists in their original entries. For example, **abide by**, which has no sense number in the *Student's Dictionary*, has been allocated the number "0" for uniformity of processing. The forms of the headword are taken from the text used at the start of each entry in the dictionary relating to the same headword, and do not necessarily all apply to the specific sense of the word under consideration.

#### 2.3.2.2 Pre-processing

*Additional notes*

The information contained in the definition text is generally restricted to the words needed to define that sense of the headword. In some cases, however, extra notes are included which restrict the operation of the definition given or provide examples or further details of normal use. It is important to separate this from the basic definition text before analysis into definition types can be carried out. This information can take the form of a note before the main definition text begins, as in sense 3 of **queen**:

```
in chess, the |queen |is the most powerful piece, which can be
moved in any direction. |3 |count n |21701
```

Alternatively, it can be appended to the definition text after a semi-colon or full stop, as in **abacus**:

```
an |abacus |is a frame used for counting. it has rods with
sliding beads on them. |0 |count n |5
```

A pre-processing program written in awk automatically identifies these parts of the definition and puts them into separate sections of the record for easier handling by the parsing programs. After pre-processing, the two definitions quoted above become:

```
the |queen |is the most powerful piece, which can be moved in any
direction. |3 |count n |21701 |queen queens | |in chess,
an |abacus |is a frame used for counting. |0 |count n |5 |abacus abacuses
|it has rods with sliding beads on them. |
```

This reveals the real pattern of the definition text and enables proper identification of the definition type for later processing. The information contained in the notes is also preserved for later reporting as a standard part of the final output format.

*Complex headwords*

The normal structure for the majority of the definitions in the *Student's Dictionary* is:

```
text before headword |single or multiple word headword| text
after headword.
```

In other words, the vertical bars corresponding to the bold type codes in the original dictionary text enclose one continuous piece of text which corresponds to the headword. There are some more complex definitions, however, such as sense 1 of **deal**:

```
|a good deal |or |a great deal |of something is a lot of
it. |1|quantif|6665
```

Here, there are two alternative pieces of headword text, in this case split by the word "or", which is not in bold type in the dictionary. In order to allow the parser to treat these definitions properly, the extra field separators generated by the extraction programs are replaced by "*", which is used during parsing to identify the alternatives in the definition. After this pre-processing the above definition becomes:

```
|a good deal *or *a great deal |of something is a lot of
it. |1|quantif| 6665|deal deals dealing dealt| |
```

This restores the basic pattern described above but preserves the original separation for later use.

### 2.3.3 The parsing software

#### 2.3.3.1 Outline of processing

The parsing software submitted as Deliverable 7 is designed to:

a. identify the definition types and extract individual definitions into files for processing by the appropriate parsing strategy
b. carry out basic analysis of definition texts into major functional components;
c. convert the record-structures created by stage b) into the output formats needed by Pisa and Bochum, analysing the basic components in more detail as it does so;
d. sort the resulting definition analyses into order of headword appearance in the dictionary; and
e. check the final output for correct descriptions of matching items, synonyms etc.

The processing is performed by shell scripts and awk programs under the control of the main shell script shown below:

```
echo Extracting definition types
gawk -f et10brk.awk defs.sed
echo Parsing individual types
parseA defsA
parseB defsB
parseC defsC
parseD defsD
parseE defsE
parseF defsF
parseG defsG
parseH defsH
echo Putting them back together
cat defs*.dispa | gawk -f sortdefs.1 | sort -n | gawk -f sortdefs.2 > alldefs
echo Checking matches etc.
gawk -f rematch.awk alldefs > alldefs.disp
```

FIGURE 1: The main shell script

Details of the processing involved at each stage is given in the descriptions in the following sections.

#### 2.3.3.2 Definition type extraction

The definition texts, pre-processed as described above, are put through an analysis program which uses pattern-matching to identify the type to which each definition belongs and to allocate it to the appropriate file for later parsing. In some cases complete allocation can be carried out using only the pattern of the definition text, but some definitions also need a check on the grammar code. The complete program is shown in Figure 2 below. In the awk programming language $1 labels the first field, i.e. the information contained in the section of the definition record before the first field separator, " | ", and $n labels the *n*th field. In the definition information used for parsing, $5 is the grammar code. $0 addresses the entire record contents. The output files, named "defsA" to "defsH", are dealt with by their individual parsing strategies.

```
BEGIN    {FS=``|''}
    {
    if (($1 ~ /^((a|an|the|your) |$)/) && ($0 !~ ``means''))
        {
        if ($5 ~ /adj/) print $0 > ``defsH''
        else print $0 > ``defsA''
        }
    else if ($0 ~ ``means'') print $0 > ``defsC''
    else if ($3 ~ /^(is|are) (a|an|the|some) /)
        {
        if ($5 ~ /adj/) print $0 >> ``defsH''
        else print $0 >> ``defsA''
        }
    else if (($1 ~ /^(if|when) ([a-z]*|(a|an|the|some(one|thing)*|one)
(in )*[a-z]*)((,|,* or) (a |an )*[a-z]*)* $/) && !($1 ~ /
(a|an|the|some|one) $/) && !($1 ~ / that( |$)/)) print $0 > ``defsB''
    else if (($1 ~ /^(if|when) /) && ($1 ~ / that( |$)/)) print $0 > ``defsD''
    else if ($1 ~ /^(if|when) /)
        {
        if ($5 ~ /vb/) print $0 >> ``defsB''
        else if ($5 ~ /adj/) print $0 > ``defsF''
        else print $0 > ``defsG''
        }
    else print $0 > ``defsE''
    }
```

FIGURE 2: Extraction of definition types

#### 2.3.3.3 Basic functional analysis

Each of the shell scripts "parseA" to "parseH", invoked by the controlling script to deal with files "defsA" to "defsH", contains two main stages, exemplified by "parseB":

```
gawk -f splitB.awk $* > $*.split
gawk -f displayB $*.split > $*.dispa
```

FIGURE 3: Main stages of functional analysis

In the first stage, the file given as argument to the "parseB" command ("defsB") is processed by the awk program "splitB.awk" to produce a basic analysis into the main functional components. This program uses a combination of positional data, punctuation and pattern matching for potential pronouns to split the definitions into:

- initial hinge ("if" or "when");
- cotext 1;
- headword;
- cotext 2;

- matching pronoun(s) for cotext 1; and
- remainder of definition text.

In this case, analysis into superordinate and discriminator elements and the identification of matching text for cotext 2 are both performed by the program "displayB", which also handles the details of output formatting. As an example, the original definition input record for sense 1 of **abide** is:

```
if you |can't abide |something, you dislike it very much.|1|
phrase|25|abide abides  abiding  abided |  |
```

After "splitB.awk" has processed it this becomes:

```
if<tab>you<tab>can't abide |something, |you |dislike it
very much.|1|phrase|25|abide abides  abiding  abided |  |
```

For ease of later processing, the fields in the output from this program are separated by a mixture of tab (shown above as <tab>) and " | " characters. The last part of the text contains a mixture of superordinate, matching pronoun (for cotext 2) and discriminator. This is more fully analysed by the second stage program, which is described in the next section.

#### 2.3.3.4 Detailed analysis and output formatting

The second stage of the parse routine, exemplified by the program "displayB", generates appropriate labels and hierarchical brackets to produce an initial output format suitable for input to the Pisa and Bochum analysis systems. Still using sense 1 of **abide** as an example, the output is laid out as:

```
(   (def_number    25)
    (sense    1)
    (def_type    1)
    (lemma    '(abide abides  abiding  abided ))
    (grammar    '(phrase))
    (pre
        (co-text0
        )
    )
    (op-word
        (hinge
            '(if)
        )
    )
    (lhs-1
        (co-text1
          (match1
            '(you)
          )
        )
        (head1
            '(can't abide )
        )
        (co-text2
          (match2
            '(something)
            '(,)
          )
        )
    )
    (rhs-2
        (match1
            '(you)
        )
        (superordinate
            '(dislike)
        )
        (match2
            '(it)
        )
        (discriminator
            '(very)
            '(much)
            '(.)
        )
    )
    (post
        (note
            '()
        )
    )
)
```

The matches, superordinates etc. identified in this output are as yet only potential analysis headings. Where necessary, this labelling is adjusted by a later program in the last stages of parsing.

#### 2.3.3.5 Merging and sorting

For ease of processing and reference by the project partners, the individual output files for each definition type are merged to form one file which is then sorted by definition number. The resulting file contains provisional analyses for the complete test vocabulary in order of the definitions' appearance in the dictionary.

#### 2.3.3.6 Checking matches and other provisional labels

The final stage of parsing is carried out on the merged file produced from the previous process. Each entry relating to a single definition is treated as a single record, and a simple awk program, "rematch.awk", checks that each potential match is realised by the existence of an appropriate matching item. If not, the "match" label is replaced by "non-match". Similarly, superordinates are checked for the presence of discriminators. If none exist, they are relabelled as "synonym".

### 2.3.4 Definition types and their parsing strategies

The type extraction routine shown in 2.2.3.2 above tests for eight different definition types, labelled "A" to "H" in the software. The distinguishing characteristics, shown in regular expression form in the awk program listing, are described in more detail below, together with the main operations of the parsing programs associated with each type.

#### 2.3.4.1 Type A

Type A definitions are typically used for nouns, and often begin with an article (especially the indefinite article) which is matched after a hinge consisting typically of "is" or "are". Two alternative sets of conditions are used in the extraction program to route definitions into the type A file:

- the text preceding the headword begins with "a", "an", "the" or "your", or there is no preceding text, and the definition does not contain the word "means"
- the text immediately following the headword begins with "is" or "are" followed by "a", "an", "the" or "some"

The first condition is tested at the beginning of the processing of the complete set of definitions. The second applies only to those definitions not already selected by the first or by the type C condition which removes those containing the word "means". Definitions matching either of the two conditions whose grammar code contains the symbol "adj" are put into type H, discussed below, since some adjectives are defined using a superficially similar definition structure. The remainder go into type A. Typical type A definitions include:

4 COUNT N
An **abstract** of an article or speech is a short piece of writing that summarizes the main points of it.

11 PLURAL N
Bed **covers** are the sheet, blankets, and bedspread that you have on top of you.

The parsing software for type A definitions uses a combination of simple positional information (based on the awk field-splitting system, using the " | " character translated from the dictionary typesetting codes) and pattern-matching to produce the first analysis. The definition is first broken down into:

- initial article
- cotext 1
- headword
- cotext 2

- hinge
- matching article
- initial discriminator(s) and superordinate(s)
- subsequent discriminator(s)

If any of these elements is missing, field separators are inserted to give an appropriate empty field. Only the boundary between the superordinate and subsequent discriminator sections involves any complexity in its pattern-matching. Because the Cobuild definitions often omit the relative clause marker at the start of the discriminator, as in the definition of orbit:

1 COUNT N OR UNCOUNT N
An **orbit** is the curved path followed by an object going round a planet, a moon, or the sun.

The boundary in this definition is formed by the past participle followed, rather than by a phrase such as "which is followed". This extends the set of possible boundary markers from around 30, if only relative pronouns, prepositions etc. could be used, to well over 200, and makes the task of exhaustively cataloguing them problematic. A second analysis program works on the chunk of text forming the preceding discriminator(s) and superordinate(s) and analyses them into their two constituent units. The more delicate analysis into precise output categories is carried out by the formatting program, which also generates the matching bracket sets for the quasi-Lisp output designed by Graham Allport. The result of this set of analyses for the definitions already cited as examples is given below:

```
(   (def_number    100)
    (sense    4)
    (def_type    3)
    (lemma    '(abstract  abstracts
                abstracting  abstracted ))
    (grammar    '(count n))
    (pre
        (co-text0
        )
    )
    (lhs-1
        (match_article
            '(an)
        )
        (head1
            '(abstract )
        )
        (co-text2
            '(of)
          (match1
            '(an)
            '(article)
            '(or)
            '(speech)
          )
        )
    )
    (link-word
        (hinge
            '(is)
        )
    )
    (rhs-2
        (match_article
            '(a)
        )
        (discriminator
            '(short)
        )
        (superordinate
            '(piece)
            '(of)
            '(writing)
        )
        (discriminator
            '(that)
            '(summarizes)
            '(the)
            '(main)
            '(points)
            '(of)
        )
        (match1
            '(it)
        )
    )
    (post
        (note
            '()
        )
    )
)
```

```
(   (def_number    6007)
    (sense    11)
    (def_type    3)
    (lemma    '(cover covers  covering
                covered ))
    (grammar    '(plural n))
    (pre
        (co-text0
        )
    )
    (lhs-1
        (cotext1
            '(bed )
        )
        (head1
            '(covers )
        )
    )
    (link-word
        (hinge
            '(are)
        )
    )
    (rhs-2
        (article
            '(the)
        )
        (superordinate
            '(sheet)
            '(,)
            '(blankets)
            '(,)
            '(and)
            '(bedspread)
        )
        (discriminator
            '(that)
            '(you)
            '(have)
        )
        (discriminator
            '(on)
            '(top)
            '(of)
            '(you)
        )
    )
    (post
        (note
            '()
        )
    )
)
```

```
(   (def_number    18894)
    (sense    1)
    (def_type    3)
    (lemma    '(orbit orbits  orbiting  orbited ))
    (grammar    '(count n or uncount n))
    (pre
        (co-text0
        )
    )
    (lhs-1
        (article
            '(an)
        )
        (head1
            '(orbit )
        )
    )
    (link-word
        (hinge
            '(is)
        )
    )
    (rhs-2
        (article
            '(the)
        )
        (discriminator
            '(curved)
        )
        (superordinate
            '(path)
        )
        (discriminator
            '(followed)
        )
        (discriminator
            '(by)
            '(an)
            '(object)
            '(going)
            '(round)
            '(a)
            '(planet)
            '(,)
            '(a)
            '(moon)
            '(,)
            '(or)
            '(the)
            '(sun)
        )
    )
    (post
        (note
            '()
        )
    )
)
```

#### 2.3.4.2 Type B

Type B definitions typically define verbs. The initial "if" or "when" which forms an invariable element of this type acts as a hinge between the two halves of the definition. The primary condition for type B definitions combines three text pattern criteria:

- text preceding the headword consists of "if" or "when", followed by a word or phrase capable of being the subject of the headword
- text preceding the headword does not end with "a", "an", "the", "some", or "one"

- text preceding the headword does not contain the word "that"

Within the first of these requirements, a word or phrase is considered capable of being the subject of the headword if it consists of:

- one word between the initial "if" or "when" and the headword, or
- the word "a", "an", "the", "someone", "something", or "one", followed optionally by a phrase made up of "in" plus one other word, followed optionally by a list of alternatives separated by commas or comma and "or", each with an optional "a" or "an" and a one-word subject element

These primary criteria are applied immediately after the extraction of type A, type H and type C definitions, as described above. The extraction program removes definitions beginning with "if" or "when" in which the text before the headword contains the word "that" (type D definitions), and then applies a second, more general test to the remaining items. From the remaining definitions which begin with "if" or "when", all those whose grammar code contains the symbol "vb" are appended to the file of type B definitions. Typical type B definitions include:

**4** VB WITH OBJ
If you **apply** a rule, system, or skill, you use it in a situation or activity.

**7** VB WITH OBJ
If reporters, newspapers, or television companies **cover** an event, they report on it.

**1** VB WITH OBJ
When you **design** something new, you plan what it should be like.

**4** VB WITH ADJUNCT OR REPORT VB
When someone in authority **rules** on a particular matter, they give an official decision about it;

The analysis programs for type B definitions have already been described in some detail in sections 2.2.3.3 and 2.2.3.4. The definitions cited above produce the following output:

```
(   (def_number    1103)
    (sense    4)
    (def_type    1)
    (lemma    '(apply applies  applying
                applied ))
    (grammar    '(vb with obj))
    (pre
        (co-text0
        )
    )
    (op-word
        (hinge
            '(if)
        )
    )
    (lhs-1
        (co-text1
          (match1
            '(you)
          )
        )
        (head1
            '(apply )
        )
        (co-text2
          (match2
            '(a)
            '(rule)
            '(,)
            '(system)
            '(,)
            '(or)
            '(skill)
            '(,)
          )
        )
    )
    (rhs-2
        (match1
            '(you)
        )
        (superordinate
            '(use)
        )
        (match2
            '(it)
        )
        (discriminator
            '(in)
            '(a)
            '(situation)
            '(or)
            '(activity)
            '(.)
        )
    )
    (post
        (note
            '()
        )
    )
)
```

```
(   (def_number    6003)
    (sense    7)
    (def_type    1)
    (lemma    '(cover covers  covering  covered ))
    (grammar    '(vb with obj))
    (pre
        (co-text0
        )
    )
    (op-word
        (hinge
            '(if)
        )
    )
    (lhs-1
        (co-text1
          (match1
            '(reporters,)
            '(newspapers,)
            '(or)
            '(television)
            '(companies)
          )
        )
        (head1
            '(cover )
        )
        (co-text2
          (match2
            '(an)
            '(event)
            '(,)
          )
        )
    )
    (rhs-2
        (match1
            '(they)
        )
        (superordinate
            '(report)
        )
        (discriminator
            '(on)
        )
        (match2
            '(it)
            '(.)
        )
    )
    (post
        (note
            '()
        )
    )
)
```

```
(   (def_number    7131)
    (sense    1)
    (def_type    1)
    (lemma    '(design designs  designing
                designed ))
    (grammar    '(vb with obj))
    (pre
        (co-text0
        )
    )
    (op-word
        (hinge
            '(when)
        )
    )
    (lhs-1
        (co-text1
          (match1
            '(you)
          )
        )
        (head1
            '(design )
        )
        (co-text2
          (match2
            '(something)
            '(new)
            '(,)
          )
        )
    )
    (rhs-2
        (match1
            '(you)
        )
        (superordinate
            '(plan)
        )
        (discriminator
            '(what)
        )
        (match2
            '(it)
        )
        (discriminator
            '(should)
            '(be)
            '(like)
            '(.)
        )
    )
    (post
        (note
            '()
        )
    )
)
```

```
(   (def_number    23356)
    (sense    4)
    (def_type    1)
    (lemma    '(rule rules  ruling  ruled ))
    (grammar    '(vb with adjunct or report vb))
    (pre
        (co-text0
        )
    )
    (op-word
        (hinge
            '(when)
        )
    )
    (lhs-1
        (co-text1
          (match1
            '(someone)
            '(in)
            '(authority)
          )
        )
        (head1
            '(rules )
        )
        (co-text2
            '(on)
          (match2
            '(a)
            '(particular)
            '(matter)
            '(,)
          )
        )
    )
    (rhs-2
        (match1
            '(they)
        )
        (superordinate
            '(give)
        )
        (discriminator
            '(an)
            '(official)
            '(decision)
        )
        (discriminator
            '(about)
        )
        (match2
            '(it)
            '(;)
        )
    )
    (post
        (note
            '()
        )
    )
)
```

#### 2.3.4.3 Type C

Type C definitions are used mainly for verbs (in which case the headword is preceded by the infinitive marker "to", matched in the second half of the definition text) and for adjectives (which normally have no text preceding the headword). The extraction criterion for type C definitions is very simple:

- the definition text contains the word "means"

This condition is applied after the first extraction of type A and type H definitions, and selects entries such as:

**3** ADJ
**High** also means great in amount, degree, or intensity.

**1** VB WITH OBJ
To **perform** a task, action, or service means to do it.

The simple structure of type C definitions makes their preliminary analysis straightforward. The parsing program splits the definition text into the following units, inserting field separators to allow for any unrealised items:

- initial infinitive marker
- headword
- hinge
- remaining definition text

The final analysis of the remaining definition text is performed by the output program, which generates the full bracketed list format. The definitions cited above yield the following output:

```
(   (def_number     13009)
    (sense     3)
    (def_type     1)
    (lemma     '(high higher  highest  highs ))
    (grammar     '(adj))
    (pre
        (co-text0
        )
    )
    (lhs-1
        (head1
            '(high )
        )
    )
    (link-word
        (hinge
            '(also means)
        )
    )
    (rhs-2
        (superordinate
            '(great)
        )
        (discriminator
            '(in)
            '(amount)
            '(,)
            '(degree)
            '(,)
            '(or)
            '(intensity)
        )
    )
    (post
        (note
            '()
        )
    )
)
```

```
(   (def_number    19806)
    (sense    1)
    (def_type    1)
    (lemma    '(perform performs  performing  performed ))
    (grammar    '(vb with obj))
    (pre
        (co-text0
        )
    )
    (op-word
        (inf
            '(to)
        )
    )
    (lhs-1
        (head1
            '(perform )
        )
        (co-text2
            '(a)
          (match2
            '(task)
            '(,)
            '(action)
            '(,)
            '(or)
            '(service)
          )
        )
    )
    (link-word
        (hinge
            '(means)
        )
    )
    (op-word
        (match-inf
            '(to)
        )
    )
    (rhs-2
        (synonym
            '(do)
        )
        (match2
            '(it)
        )
    )
    (post
        (note
            '()
        )
    )
)
```

#### 2.3.4.4 Type D

Type D definitions begin with "if" or "when", but are distinguished from type B by the nature of the remaining text before the headword. They contain an element labelled "projection" in the definition language grammar, and define headwords which can only be dealt with through metalinguistic statement. For example, sense 3 of **depends**:

**3** VB
If you say that something **depends** on something else, you mean that it will only happen if the circumstances are right.

The initial projection "you say that" is matched by "you mean that", and between them these elements place the headword and its explanation in a reported speech context, defining it directly in terms of the circumstances of its usage. After all type A, type H and type C definitions have been extracted, and after the first selection of type B definitions, type D is extracted using the condition:

- text preceding the headword contains the word "that"

The first type D parsing program splits the definition text into the following elements, creating blank fields for unrealised items:

- hinge
- projection
- cotext 1
- headword
- cotext 2
- remaining definition text

The final parsing program analyses the remaining definition text into matching projection, superordinate(s) and discriminator(s) and creates the final output format. The entry for **depends** cited above produces the output:

```
(   (def_number    7047)
    (sense    3)
    (def_type    1)
    (lemma    '(depend depends  depending  depended ))
    (grammar    '(vb))
    (pre
        (co-text0
        )
    )
    (op-word
        (hinge
            '(if)
        )
    )
    (lhs-1
        (projection
            '(you say that)
        )
        (co-text1
          (match1
            '(something)
          )
        )
        (head1
            '(depends )
        )
        (co-text2
            '(on)
          (non-match
            '(something)
            '(else)
            '(,)
          )
        )
    )
    (rhs-2
        (match_projection
            '(you mean that)
        )
        (match1
            '(it)
        )
        (superordinate
            '(will)
            '(only)
            '(happen)
        )
        (discriminator
            '(if)
            '(the)
            '(circumstances)
            '(are)
            '(right)
        )
    )
    (post
        (note
            '()
        )
    )
)
```

#### 2.3.4.5 Type E

Within the context of the test vocabulary type E definitions form the default option, those definitions left after types A, B, C, D, F, G and H have been extracted. Because of the restricted nature of the test vocabulary this produces a completely uniform set of definitions, typified by:

**0** ADJ
Something that is **debatable** is not definitely true or not certain.

**1** ADJ
Something that is **low** measures a short distance from the bottom to the top.

The distinguishing characteristic of these definitions is the use of the introductory phrase "something that" or "someone who" before a hinge "is" which is either matched or replaced in the second part of the definition. An extraction program based on the whole dictionary would select type E definitions with the simple condition:

- text preceding the headword consists of "someone" or "something", followed by "who" or "that", followed by "is" or "are"

The initial parsing program for type E definitions splits the definition text into:

- cotext 1
- left hand hinge
- headword
- right hand hinge
- remaining definition text

The second parsing program checks the relationship between the left hand and right hand hinges and labels them appropriately, analyses the remaining definition text and produces the normal output format. The output for the two definitions already quoted is given below:

```
(   (def_number    6700)
    (sense    0)
    (def_type    3)
    (lemma    '(debatable ))
    (grammar    '(adj))
    (pre
        (co-text0
        )
    )
    (lhs-1
        (co-text1
            '(something  that)
        )
        (match_hinge
            '(is )
        )
        (head1
            '(debatable )
        )
    )
    (link-word
        (match_hinge
            '(is)
        )
    )
    (rhs-2
        (discriminator
            '(not)
            '(definitely)
        )
        (superordinate
            '(true)
        )
        (disjunct
            '(or)
        )
        (discriminator
            '(not)
        )
        (superordinate
            '(certain)
        )
    )
    (post
        (note
            '()
        )
    )
)
```

```
(   (def_number    16269)
    (sense    1)
    (def_type    3)
    (lemma    '(low lower  lowest  lows ))
    (grammar    '(adj))
    (pre
        (co-text0
        )
    )
    (lhs-1
        (co-text1
            '(something  that)
        )
        (lhs_hinge
            '(is )
        )
        (head1
            '(low )
        )
    )
    (link-word
        (rhs_hinge
            '(measures)
        )
    )
    (rhs-2
        (superordinate
            '(a)
            '(short)
            '(distance)
        )
        (discriminator
            '(from)
            '(the)
            '(bottom)
            '(to)
            '(the)
            '(top)
        )
    )
    (post
        (note
            '()
        )
    )
)
```

#### 2.3.4.6 Type F

After the type A, B, C, D and H definitions have been removed, type F is identified from the remaining definitions beginning with "if" or "when" by a grammar code containing the symbol "adj". Two examples of type F definitions contained in the test vocabulary are given below:

**0** ADJ
If you are **dependent** on someone or something, you need them to survive.

**7** ADJ
If the quality or standard of something is **low**, it is bad.

The first parsing program for type F definitions splits the definition text into the following units and inserts field separators to allow for unrealised items:

- hinge
- cotext 1
- left hand hinge
- headword
- cotext 2
- right hand hinge
- matching cotext 1
- remaining definition text

The second program, "displayF", analyses the remaining definition text and generates the normal bracketed list output, shown below for the two definitions already cited.

```
(   (def_number     7053)
    (sense     0)
    (def_type     1)
    (lemma     '(dependent ))
    (grammar     '(adj))
    (pre
        (co-text0
        )
    )
    (op-word
        (hinge
            '(if)
        )
    )
    (lhs-1
        (co-text1
          (match1
            '(you)
          )
        )
        (lhs_hinge
            '(are)
        )
        (head1
            '(dependent )
        )
        (co-text2
            '(on)
          (match2
            '(someone)
            '(or)
            '(something)
            '(,)
          )
        )
    )
    (rhs-2
        (match1
            '(you)
        )
        (superordinate
            '(need)
        )
        (match2
            '(them)
        )
        (discriminator
            '(to)
            '(survive)
        )
    )
    (post
        (note
            '()
        )
    )
)
```

```
(   (def_number    16275)
    (sense    7)
    (def_type    1)
    (lemma    '(low lower  lowest  lows ))
    (grammar    '(adj))
    (pre
        (co-text0
        )
    )
    (op-word
        (hinge
            '(if)
        )
    )
    (lhs-1
        (co-text1
          (match1
            '(the)
            '(quality)
            '(or)
            '(standard)
            '(of)
            '(something)
          )
        )
        (lhs_hinge
            '(is)
        )
        (head1
            '(low, )
        )
    )
    (rhs-2
        (match1
            '(it)
        )
        (rhs_hinge
            '(is)
        )
        (synonym
            '(bad.)
        )
    )
    (post
        (note
            '()
        )
    )
)
```

### 2.3.4.7 Type G

Type G definitions are those beginning with "if" or "when" whose grammar codes specify that they are nouns. In the analysis of the test vocabulary this is a default option after type B, D and F definitions have been removed from those beginning with "if" or "when". Typical examples are:

**6** UNCOUNT N
If you have **charge** of something or someone, you have responsibility for them.

**0** COUNT N OR UNCOUNT N
When there is an **emission** of gas or radiation, it is released into the atmosphere;

The first parsing program for type G definitions splits them into the following units, creating blank fields for unrealised items:

- hinge
- cotext 1
- left hand hinge
- article
- headword
- matching cotext
- right hand hinge
- remaining definition text

The second program adjusts the descriptions of these elements where necessary, analyses the remaining definition text and produces the normal bracketed output. The fully analysed output for the definitions already used as examples is given below:

```
(   (def_number    4213)
    (sense    6)
    (def_type    1)
    (lemma    '(charge charges  charging
                charged ))
    (grammar    '(uncount n))
    (pre
        (co-text0
        )
    )
    (op-word
        (hinge
            '(if)
        )
    )
    (lhs-1
        (co-text1
          (match1
            '(you)
          )
        )
        (lhs_hinge
            '(have)
        )
        (head1
            '(charge )
        )
        (co-text2
            '(of)
          (match2
            '(something)
            '(or)
            '(someone)
            '(,)
          )
        )
    )
    (rhs-2
        (match1
            '(you)
        )
        (rhs_hinge
            '(have)
        )
        (superordinate
            '(responsibility)
        )
        (discriminator
            '(for)
        )
        (match2
            '(them)
        )
    )
    (post
        (note
            '()
        )
    )
)
```

```
(   (def_number    8665)
    (sense    0)
    (def_type    1)
    (lemma    '(emission emissions ))
    (grammar    '(count n or uncount n))
    (pre
        (co-text0
        )
    )
    (op-word
        (hinge
            '(when)
        )
    )
    (lhs-1
        (co-text1
          (non-match
            '(there)
          )
        )
        (lhs_hinge
            '(is)
        )
        (article
            '(an)
        )
        (head1
            '(emission )
        )
        (co-text2
            '(of)
          (match2
            '(gas)
            '(or)
            '(radiation)
            '(,)
          )
        )
    )
    (rhs-2
        (match2
            '(it)
        )
        (rhs_hinge
            '(is)
        )
        (superordinate
            '(released)
        )
        (discriminator
            '(into)
            '(the)
            '(atmosphere)
        )
    )
    (post
        (note
            '()
        )
    )
)
```

#### 2.3.4.8 Type H

The extraction of these definitions has already been described: they are the definitions which meet the type A criteria but have "adj" in their grammar codes. Typical examples are:

**6** ADJ
A **high** position in a profession or society is an important one.

5 ADJ
A **steady** person is sensible and reliable.

The first parsing program splits the definitions into the following units, creating blank fields for unrealised items:

- article
- headword
- cotext 2
- hinge
- remaining definition text

The second program analyses the remaining text and creates the output format shown below for the definitions used as type H examples earlier.

```
(   (def_number    13012)
    (sense    6)
    (def_type    3)
    (lemma    '(high higher highest
                highs ))
    (grammar    '(adj))
    (pre
        (co-text0
        )
    )
    (lhs-1
        (article
            '(a )
        )
        (head1
            '(high )
        )
        (co-text2
          (match2
            '(position)
            '(in)
            '(a)
            '(profession)
            '(or)
            '(society)
          )
        )
    )
    (link-word
        (hinge
            '(is)
        )
    )
    (rhs-2
        (article
            '(an)
        )
        (synonym
            '(important)
        )
        (match2
            '(one)
        )
    )
    (post
        (note
            '()
        )
    )
)
```

```
(   (def_number    26335)
    (sense    5)
    (def_type    3)
    (lemma    '(steady steadier  steadiest
                steadies  steadying  steadied ))
    (grammar    '(adj))
    (pre
        (co-text0
        )
    )
    (lhs-1
        (article
            '(a )
        )
        (head1
            '(steady )
        )
        (co-text2
            '(person)
        )
    )
    (link-word
        (hinge
            '(is)
        )
    )
    (rhs-2
        (synonym
            '(sensible)
            '(and)
            '(rel able)
        )
    )
    (post
        (note
            '()
        )
    )
)
```

## 2.4 The implications of the system

### 2.4.1 Cobuild definitions and technical vocabulary

For each of the words dealt with below, a concordance listing taken from the ITU corpus was produced, and compared with the definitions in the Student's Dictionary. Two main sources of difference became apparent:

a. words in the ITU text are generally used in a technical and impersonal context, whereas the definitions in the Student's Dictionary, based on a general corpus, assume a personal subject. A process of 'dehumanisation', referred to in the examples below, would make the dictionary definitions more appropriate in many cases;
b. in some cases, such as the word **assembly** given below, the dictionary defines the process associated with the word while the corpus uses it for the product of that process.

Where a definition has been parsed as part of the test vocabulary it is asterisked.

**Assembly**

*Concordance listing*

```
rcular rail supports the mechanical assembly and allows its rotation in azimuth,
nna is composed of: -the electrical assembly (as described in  5.2.2) which cons
xed-coupling systems Mixed transmit sub-assembly configurations, employing both
acing between an analogue multiplex assembly (e.g. a 60 channels FDM supergroup)
and monitoring), -compact equipment assembly (e.g. all equipment contained in a
upergroup)  and a digital multiplex assembly (e.g. two 30 channels PCM groups) -
tal networks (ISDNs) Rec. X.3Packet assembly/disassembly facility (PAD) in a pub
inal equipment accessing the packet assembly/disassembly facility (PAD) in a pub
 channels in the baseband multiplex assembly. ii) In consequence, SSB transmissi
nsmit) the terrestrial FDM baseband assembly into the minimum number of supergro
 and since 1986 (CCIR XVIth Plenary Assembly), it has been studying the performa
h-over-elevation (Az-El) mechanical assembly of the wheel and track type. In thi
tive elements. It is composed of an assembly of various telecommunication sub-sy
n digital. The complete cable is an assembly of multiple individual (10-50) coax
---------------+ Note 1.- Group: an assembly of 12 telephone channels derived fr
 kHz). Note 2.- Supergroup (SG): an assembly of 60 telephone channels derived fr
be determined by the XVIIth Plenary Assembly of the CCIR, taking  5.2.4  Antenna
tion and user data between a packet assembly/disassembly (PAD) facility and a pa
ture, thus converting it into a new assembly possessing different statistical an
ture, thus converting it into a new assembly possessing different statistical an
lack of flexibility in the transmit sub-assembly; -restrictions in the transmit
e baseband of a telephony multiplex assembly. The transmitted RF signal is then
mmetrical) terrestrial FDM baseband assembly; -the so-called ßatellite multiple
system which permits the electrical assembly to be steered in any possible orien
stal) which supports the electrical assembly (usually on two orthogonal movable
```

*Definitions*

```
1
COUNT N
An [assembly] is a large number of people gathered together, especially a group
of people who meet regularly to make laws.
2
UNCOUNT N
[Assembly] is the gathering together of people for a particular purpose.
*3
UNCOUNT N
The [assembly] of a machine or device is the process of fitting its parts
together.
```

Sense 3, the parsed definition, is probably closest because of its reference to "a machine or device", but it refers to the process rather than the product, an assembly or sub-assembly as a component of a machine, device or system, which is the usage in ITU.

## Message

*Concordance listing*

```
TV carrier (Tx4) in addition to the message carriers (Tx1, 2, 3). If TV transmis
d satellite channels, an assignment message containing connection information is
 way that the uniqueness of a given message is accentuated, thereby allowing a b
 way that the uniqueness of a given message is accentuated, thereby allowing a b
-transmit the erroneous part of the message. It is clear that the necessity of r
s in accordance with the assignment message sent from a transmit station. The in
Modulation,         FM-FDM-FDMA for message: Similar to Standard A Similar to St
dundancy into the message word. The message source delivers information bits at
dundancy into the message word. The message source delivers information bits at
with access but with control of the message source and more specifically with vo
with access but with control of the message source and more specifically with vo
 more than 20% by removing from the message the time slots corresponding to the
 more than 20% by removing from the message the time slots corresponding to the
o an observer) from the transmitted message. The characterization of these types
o an observer) from the transmitted message. The characterization of these types
 bits/sample), the DCME will send a message to the telephone exchange (generally
 is used which, for each elementary message to be transmitted, transmits a code
 is used which, for each elementary message to be transmitted, transmits a code
ould normally be employed where the message traffic is simple and consists mainl
-a PCM system designed for analogue message transmission is readily adapted to o
                                   message transmission in
pecially intended for high capacity message transmission4.  Their main features
ntroduction of  redundancy into the message word. The message source delivers in
ntroduction of  redundancy into the message word. The message source delivers in
                    FM-FDM-FDMA for message(1):                    - One Tx chai
8 Modulation,       FM-FDM-FDMA for message(1):                - One Tx chain (m
ission                             message:
 Modulation,        FM-FDM-FDMA for message(1):                    Similar to St
                   FM-SCPC-FDMA for message:                      Similar to Sta
                DSI-PSK-TDM-TDMA for message(1)                   Transponder hopp
```

*Definitions*

```
1
COUNT N
A [message] is a piece of information or a request that you send to someone or
leave for them.
2
COUNT N with SUPP
A [message] is also the idea that someone tries to communicate to people, for
example in a play or a speech.
```

**Sense 1, suitably dehumanised, is closest.**

## Power

*Concordance listing*

**Because of the large number of citations for power in the ITU text a random sample of about 1 in 10 has been given.**

```
ter-facility link 3 dB couplers (low power) (A L'ITALIENNE)  5.4.8.4 Comparison
d output power of earth station high power  amplifier (see Chapter 5,  5.4.7). i
e added in the input combiner of the power amplifier sub-system. Similarly, each
effect of an increase in TR. 5.1.1.3 Power amplifiers The order of magnitude of
in power amplifiers Non-linearity in power amplifiers causes such effects as int
                              When the power and spectrum of each intermodulation
is limited by the overall TWT output power and, more precisely, by the output po
 be used for up to about 3 kW output power at 6 GHz. However, at higher frequenc
 reduce the satellite average output power by 50% or more to reduce intermodulat
on which links the  C/N ratio of the power C of the modulated signal after the t
```

```
ossibility of operating with reduced power consumption (for reduced HF power). K
itable and can generally be used for power consumption not exceeding 500 W. 5.1.
rate, Rb. If S is the desired signal power, Eb the average transmitted energy pe
 the case of satellite transmission, power efficiency is of great importance, lo
wer sub-systems are as follows: -the power entrance facility; including the high
                          (16) The power flux-density (pfd) radiated in a give
e to meet the needs of the station's power generating equipment, should be provi
current of the power line and on the power generators in the station. 5.7.6  Sum
nna system with superposed grids d)  Power handling Each generation of satellite
 IMPATT diode amplifiers with higher power have also been proposed but are scarc
ides total isolation from commercial power interruptions and line disturbances i
 of operation in that the routing of power may be derived from either or both tr
er measurement A : alarm Po : output power measurement Pr : reflected power meas
e formula for the calculation of the power of each third-order intermodulation p
23 -Intermodulation noise The output power of each carrier divided by the interm
ossible variations: - of the average power of all the bursts in the TDMA frame (
l oscillator, which has to produce a power of about 10 dBm, may be a conventiona
e formula for the calculation of the power of each third-order intermodulation p
ation HPA and of the required output power of this HPA. The activity factor to b
 2.20 gives an example of the output power per carrier versus total input power
    where: TT/N : test-tone-to-noise power ratio (dB), C/N : carrier-to-noise ra
nna having an effective area Ae, the power reaching the antenna is equal to: W
 not sufficient to deliver the total power required by all the RF channels. In a
tivated and this permits up to a 60% power saving in the satellite transponder (
----------------+ As a general rule, power should be supplied to equipments in t
antenna, telecommunication equipment power supply, administration and support se
two main sources of power: -the main power supply, with stand-by capability, -th
n the correct design of its electric power supply. There are two main sources of
 mass, the EOL (end-of-life) primary power, the RF (radio frequency) power, the
 by the payload, -supply of electric power to the payload, -eclipse operation. A
d with lightning arrestors. Two main power transformers should be used to step d
e receiving amplifiers and with high power transmit amplifiers. They can handle
tor and/or heat pipes in the case of high-power TWT; -special heat conditioning
g up to 30 W. At 11-12 GHz band, low power TWTAs of between 10 and 20 W and medi
tical and costly than earth terminal power. Typically, one might have to reduce
lite     mass, BOL(1)   power        power     operation    life
-------------+   Maximum transmitted power            34.8 dBW           16
```

## *Definitions*

```
1
UNCOUNT N
Someone who has [power] has control over people and activities.
2
UNCOUNT N with SUPP
Your [power] to do something is your ability to do it.
3
COUNT N or UNCOUNT N with SUPP
If someone in authority has the [power] to do something, they have the
legal right to do it.
*4
UNCOUNT N with SUPP
The [power] of something is its physical strength.
*5
UNCOUNT N
[Power] is energy obtained, for example, by burning fuel or by using the wind
or the sun.
6
UNCOUNT N
Electricity is often referred to as [power].
7
VB with OBJ
To [power] a machine means to provide the energy that makes it work.

To [come to power] means to take charge of a country's affairs.
If someone is [in power], they are in charge of a country's affairs.
If something is [within] or [in] your [power], you are able to do it.
```

Senses 4 and 5 convey the idea of a force, and sense 4 brings in the synonym **strength** which is appropriate in describing the power of a signal. Senses 5 and 6 both cover the idea of power as energy. The concept of capacity which is involved in many of the ITU citations is not covered explicitly.

## Step

*Concordance listing*

```
cs of the signal to be quantized. The step adaptation is then selected in such a
nstantaneous  value of the quantizing step at time n. This value is adapted as a
o-system. The antenna beam is steered step by step so as to obtain a stronger si
 power transformers should be used to step down the incoming voltage. The transf
, it is logical to envisage going one step further and entrusting the satellite
ather superfluous manner. The logical step is therefore to design an adaptive co
e quantizing, in which the quantizing step is made variable in time as a functio
ary techno-economic studies The first step is to assess the service requirements
ather superfluous manner. The logical step is therefore to design an adaptive co
oefficients   are corrected; -at each step j of the registers, a multiplier calc
ry rarely employed. a) Step-track The step-track method uses a so-called climbi
osition as shown in Fig. 5.12. If the step-steering of the antenna beam has decr
and only a simple beacon receiver and step-track processor are required.  Howeve
creased the receive signal level, the step-track processor will command the ante
ber of possible vectors, some kind of step search algorithm has certain computat
ber of possible vectors, some kind of step search algorithm has certain computat
nd 5 of Table 5.III), - errors due to step size and to beacon signal level measu
e delta modulation (ADM) the variable step size increases during a steep segment
ng an adaptive quantizer, wherein the step size is varied automatically in accor
. The antenna beam is steered step by step so as to obtain a stronger signal fro
 = 1 in Table examples). FIGURE 5.12 -Step-track system antenna boresight axis a
carrier of a satellite beacon. In the step-track system, no special tracking fee
as, especially when combined with the step-track system. A typical block diagram
 tracking the satellite direction are step-track (Table 5.III - Type No. 4) and
beam, is now very rarely employed. a) Step-track The step-track method uses a so
ed for wholly terrestrial systems) to step through the international network fro
 _ _ 26.3o   - Tracking: monopulse or step track  Rec. 465 applies.
:  in any direction of the- Tracking: step track type       Typical environment
e partially overcome by combining the step track with a program or memory which
                                    (step track)    0.015      0.12 dB
(Type No. 5) and 0.015  (r.m.s.) with step tracking (Type No. 4). Note. -The ant
               - All medium sized     Step tracking
stems     efficient              4    Step tracking
age fused disconnect switch feeding a step-down transformer. The transformer sec
ntaneous amplitude of the signal. The step variation rule makes use of the multi
.   If f is the frequency synthesizer step, which should be at least as wide as
ay have simple tracking devices (e.g. step-track), while small antennas generall
al) is transmitted, the    quantizing step will be multiplied by M4 (> 1) at the
```

*Definitions*

```
1
COUNT N
A [step] is the movement made by lifting your foot and putting it down in a
different place.
2
VB with ADJUNCT
If you [step] on something, you put your foot on it.
3
VB with ADJUNCT
If you [step] in a particular direction, you move in that direction.
*4
COUNT N
A [step] is also one of a series of stages, or a single action taken for a
particular purpose.
```

```
5
COUNT N
A [step] is also a raised flat surface, often one of a series, on which you put
your feet in order to walk up or down to a different level.

If someone tells you to [watch] your [step], they are warning you to be more
careful about your behaviour so that you don't get into trouble.
If you do something [step by step], you do it by progressing gradually from one
stage to the next.
If a group of people are walking [in step], they are moving their feet forward
at exactly the same time as each other.
PHR
step aside
PHR VB
PHR
step back
PHR VB
If you [step back], you think about a situation in a fresh and detached way.
PHR
step down
PHR VB
If you [step down] or [step aside], you resign from an important job or
position.
PHR
step in
PHR VB
If you [step in], you start to help in a difficult situation.
PHR
step up
PHR VB
If you [step up] something, you increase it.
```

Sense 4 seems to match the main usage almost perfectly, although the phrasal use "**step** through" is not covered, and the definition given for the phrase **step down** is not appropriate.

### Address

*Concordance listing*

```
tream of packets each with the same address, and it is the recognition of this a
ket contains a header containing an address followed by a user data block. A sin
CCIR has been meeting since 1983 to address issues related to ISDN performance o
, and it is the recognition of this address that enables a receiver to select th
```

*Definitions*

```
1
COUNT N
Your [address] is the number of the house, the name of the street, and the town
where you live.
2
VB with OBJ
If a letter [is addressed] to you, your name and address are written on it.
3
COUNT N
An [address] is also a formal speech.
*4
VB with OBJ
If you [address] a group of people, you give a speech to them.
5
VB with OBJ
To [address] a problem means to deal with it.
```

Sense 1 is closest to the main usage, but it needs to be "dehumanised" to take away the "house" element and replace it with a more general location explanation. Sense 5 matches the single instance of a verbal usage.

### 2.4.2 The parser in the practice of lexicography

In this section the work done in Project ET10/51 will be evaluated from the point of view of potential users of the parser and its derivatives. There are two cross-cutting dimensions:

a. The type of user. We distinguish three:

   i. The human being—unassisted,
   ii. The human being—assisted by machines,
   iii. The machine—unassisted by human beings

b. The software package. Because of the design of the Project, we again distinguish three states of software corresponding to our respective laboratories:

   i. The Dictionary Parser: Birmingham
   ii. The HPSG Interface: Bochum
   iii. The Typed Feature Structures: Pisa

Because this is a pilot study, there are many applications that can be proposed but which require the software to be extended to cover more than a sample of the language. Hence, within each category, we shall reflect two stages of the software:

   i. As it is on handover at the end of the Project
   ii. As it might be when extended

The parser reveals the way in which the meaning of a sentence is organised. In the practice of lexicography it offers an important tool for the dictionary compiler and editor, and it can be adapted to provide many more. A few will be described briefly below.

The parser deals with whole-sentence definitions, and therefore it is a partial parser of English. Most existing printed dictionaries use a specialised set of conventions for presenting their information, including all sorts of codes, abbreviations, non-standard characters and graphic symbols. Hence the parser would have to be adapted for traditional definitions; projects such as ACQUILEX may be covering this ground.

#### 2.4.2.1 Evaluation of draft definitions

Lexicography is a team pursuit usually, with staff, often isolated, working to explicit compilation rules. Incoming work can be checked as follows:

- Does the word class (noun, verb etc.), check with the definition type used?
- Do the cotextual restrictions fall into one or other of the established classes of cotexts?
- Classify the superordinate (eventually using draft thesaurus, see below). If this is a new or unexpected one, check that it is appropriate and consistent with policy.
- Examine the discriminators. Are they similar to members of established classes?
- Check all the matches and what has happened if a mismatch has been found.

#### 2.4.2.2 Enhancement of definitions

As well as simple checking, the parser can be used to improve the practice of definition. Definitions are expected to be consistent in phraseology, so that variation in phrasing is significant. The parser can compare and contrast. Also dictionaries try to use the defining vocabulary in a controlled way. One current dictionary claims to present all its definitions within a finite and preset wordlist, and all are sensitive to this question, even dictionaries for the native speaker.

The Cobuild dictionary uses several different types of whole sentence definition, but not all. The parser is capable of further development until it can identify and analyse all sentences in English that function as definition sentences. The development in the main should be just filling out lists of operating classes.

The parser can be used to suggest improvements to the definitions that already exist. For example in analysing a definition such as:

1 COUNT N OR UNCOUNT N
**Grassland** is land covered with wild grass.

it is clear that *covered* introduces a discriminator. However there are hundreds of *-ed* words that perform this function, and enhancements of the definition types would extend the list to virtually any past participle in English. From the point of view of economy in the grammar, it would be helpful to rewrite all such definitions like:

**Grassland** is land *which is* covered with wild grass.

Now the discriminator is *which*, a word already used in this function in other definition types, and the class is not added to. Since space is not critical electronically, there is no disadvantage in making such a change throughout the set of definition statements. The greater explicitness of phrasing exposes the structure of the definition, and this could be an advantage to many users. The definition statement with respect to this feature falls in with such as:

1 COUNT N
An **instrument** is a tool or device that is used to do a particular task.
1 UNCOUNT N
**Equipment** consists of things which are needed for a particular activity.

Similarily the structure of the definition is obscured in:

4 ADJ
An **active** volcano has erupted recently.

It would be more explicit to expand this to:

An **active** volcano is one that has erupted recently.

The hinge is clearly is, which is one of the commonest hinges; *one* can match *volcano* and realise a very interesting definition type—where the superordinate is a match, and therefore not part of the *definiens*.

There is thus a range of automatic rewriting possibilities, some of which would help the human user and some which would help the machine. A thorough study of the standardisation of phraseology would lead to better support for the lexicographers ( see below).

#### 2.4.2.3 Sense discrimination

A comprehensive and accurate dictionary should give a lot of help in sense discrimination. But first, from a computational point of view, the dictionary text is heavily ambiguous. For example, *velvet* is defined as "soft material...", which invokes *material* sense 2 (out of 5): "**Material** is cloth". In turn, the sense of *cloth* that we need is no. 1 out of 2, "**Cloth** is fabric ...", and *fabric* 1 (out of 3) tells us that it is "a type of cloth"—not the most helpful definition in the route that we have taken to it. However, *type* is a common word in complex superordinates, and we mean *type* 1 out of 5, of which the superordinate is *class* (4 out of 6) of which the superordinate is *group* (2 out of 4), leading to *set* (1 out of 22), leading to *number* (phrase "**a number of** things" following 3 senses), and so on.

It is clear from this exercise that the sense used in the definition is not always no. 1, and that there is a huge amount of ambiguity to clear up before the dictionary can be reliably used by a machine. But when that is done, and if it can be done by automatic process, the same techniques will form a basis for the resolution of ambiguity in ordinary texts in the language.

From the user's perspective, a parsed dictionary will be an excellent test bed for the development of disambiguation techniques. Each sentence is self-contained, and the process of definition in the Cobuild style is likely to provide a supportive phrasing.

#### 2.4.2.4 New definitions

The set of definitions forms a set of blueprints for making new definitions. For example it would be simple to devise a routine which, given basic word-class information about a new word, constructed a framework definition that guided the lexicographer—almost like a form to fill in.

It should be possible also to work out systematic relations between the full-sentence definitions of Cobuild and the traditional type of definition. Some kinds of information are only found in Cobuild, but most kinds are found in both but organised differently. Dictionaries of the future may well require both kinds of definition, and it would be helpful to be able to move from one to another with the minimum of human input.

#### 2.4.2.5 New tools: thesaurus

We may expect that the availability of the parser will give rise to a number of new tools in natural language processing. The most obvious and urgent is a thesaurus, where again the whole-sentence style of definition will open up interesting kinds of thesaurus classification, and with the dictionary in the background, the relevant meaning of the word will always be available.

#### 2.4.2.6 Automatic procedures in lexicography

It was pointed out above that a blueprint for the definition of a new word could be supplied automatically. This is one of the steps towards automatic lexicography that may be envisaged with the parser at the centre. Adding new entries, new senses to existing dictionary entries, revising and checking existing dictionaries—all these require a set of tools which will process corpus evidence and derive material of lexicographic interest from it; then compose a statement about it for inclusion in a reference book or database. The parser, and a generator which can be made by reversing the parser, are two of the central tools.

It was also mentioned above that the parser could be used diagnostically—that it could read text and decide which sentences were definition ones, and therefore parseable. This device could thus be an extraction tool for definition information from texts, and could be run against very large text streams to provide automatic drafts of new words and senses.

#### 2.4.2.7 Large lexicons

The parser is the first stage of a process at the end of which are formal statements of the semantic features of the words parsed. There is thus a clear and automatic route from a Cobuild definition to a lexicon entry. This connection may speed up and extend the building of machine usable lexicons—over 30,000 words are to be found in the smallest of the Cobuild dictionaries. For many purposes the provision of full-sentence definitions may be much easier and cheaper than constructing the lexicons with highly-paid formal linguists.

Further, given the development envisaged above, to turn the parser into an extraction tool, lexicons may be constructed quite automatically in the future. From a corpus of the chosen kind of material the extractor will find definition sentences, which the parser will parse and the Bochum or Pisa formalisms will be applied.

NICOLETTA CALZOLARI, STEFANO FEDERICI, SIMONETTA MONTEMAGNI & CAROL PETERS

# 3 Extracting, Representing and Using Syntactic-Semantic Information from Cobuild Definitions

## 3.1 Introduction

In the previous section, a detailed description has been given of the definition statements contained in Cobuild Student's Dictionary (CSD) and of the grammar and parser that have been developed at Birmingham to categorise these definitions and analyse them systematically, identifying the functions of the different components and tagging them accordingly. In this section, we will present the work that has been carried out in Pisa on the design and implementation of a specialised parser to extract lexical information from this syntactically tagged definition input provided by Birmingham and to represent it formally in a computationally tractable way. Our aim has been (i) to analyse the definitions in great detail in order to define strategies that would permit us to extract as much useful information as possible—not only syntactic and semantic but also morphological, and pragmatic, (ii) to devise efficient representation mechanisms in order to construct lexical entries that could find a range of applications, not only in NLP but also in other language activities where formalized lexical data is needed.

In this work our primary objective, as specified in the Technical Annex to the project, has been to study a method to represent formally the actual usage of words. This has differentiated our results from those of other studies which have also analysed machine readable dictionary definitions in order to derive and formalize the lexical data contained in them. Such studies have tended to concentrate on classifying the headword mainly in terms of the particular semantic features that can be derived from the genus items and their differentiae, and then on establishing taxonomic chains throughout the dictionary, representing the lexicon as a hierarchical inheritance network. An important example is the ACQUILEX project which has concentrated on deriving a formalized representation of meaning, analysing traditional dictionaries (for English, Dutch, Spanish and Italian) and developing a common type system at the top level of the hierarchy for the four languages (see Calzolari et al. (1993)). However, it is clear that the computational lexicons of the future will need much more than purely taxonomic information. As is known, words do not have value in isolation but acquire sense from their lexical/syntactic context. Thus one of the most important features to encode in a formalized lexical entry will regard the combinatorial properties of the headword. For this reason, the decision was taken to analyse a dictionary belonging the Cobuild range—the Cobuild Student's dictionary—in this project.

Cobuild dictionaries have two very special features: (i) they are compiled on the basis of the evidence provided by a very large corpus of contemporary English; (ii) the definitions appear as natural language sentences, i.e. with the definiendum inserted in its typical sentential context. In fact, each definition statement is the result of a careful analysis by the lexicographers of all the contexts they had available for that word sense, extracted from a corpus of more than twenty million words of running text of contemporary spoken and written English. From their studies of the corpus, it was clear to the Cobuild lexicographers that, in real language usage, it is context that disambiguates sense. In particular, it became increasingly evident that not only were particular structural patterns frequently associated with particular senses but that very often corpus data showed not just a typical syntax but a typical pattern of lexical collocation as well (see Sinclair (1987)). Therefore, they realised that, in order to represent meaning faithfully, the importance of context had to be recognized in some way in the definition. It was thus decided that, whenever possible, the Cobuild definition statements should not only explain the headword in each of its identified senses (as in other dictionaries) but should place it in its most typical context, as revealed by corpus evidence. Therefore, while in traditional lexicography, statements are made about what

words mean but very little about their use, the innovative form of the Cobuild definition equation gives attention not only to explaining the meaning of the headword (in the RHS) but also to illustrating its use (in the LHS).

In order to do this effectively, a number of explanatory strategies were studied. The intention was to write the definition in a language which was as close as possible to natural language in order not just to help the reader to decode a text but to provide him/her with useful models for encoding. To do this, whenever possible, the user was to be provided with the typical grammar structures and typical collocates for each sense of a lexical entry. Compare, for example, the first sense of the verb **to kill** as defined in the Cobuild Student's dictionary:

1 VB WITH OR WITHOUT OBJ OR REFL VB
When someone or something **kills** a person, animal or plant, they cause the person, animal or plant to die

with the primary sense of the same verb in OALDCE, defined by "To put to death; cause the death of" or in LDOCE "to cause to die" (examples taken from Hanks (1987)). All three entries give the basic meaning (to cause death) but, from the Cobuild definition, it can also be easily inferred that this sense of the verb requires an animate direct object, whereas the subject can be either human or inanimate.

A similar example is given by the definition for the verb **diagnose**:

1 VB WITH OBJ
When a doctor **diagnoses** a disease that someone has, he or she identifies what is wrong.

From the LHS of this definition it can be inferred that the required or preferred subject of this verb is a doctor, the required direct object a disease, and that this disease (i.e. a disease diagnosed by a doctor) is particular of human beings; in addition to stating the meaning, the RHS indirectly assigns the features human, male/female to the argument **doctor** and inanimate to **disease**. This kind of information is not found in the same way in other dictionaries, cfr. OALDCE: "determine the nature of (esp a disease) from observation of symptoms", from which we can only infer that **disease** is related in some way with **diagnose** but the exact nature of the relationship is not immediately recognizable from the definition text.

Indeed, different senses of a word can frequently be distinguished by the different kinds of arguments or collocates associated with them. Cobuild attempts to make this clear. For example, the two senses of **adore** in the Student's dictionary are explained as follows:

1 VB WITH OBJ
If you **adore** someone, you love and admire them.

2 VB WITH OBJ
If you **adore** something, you like it very much.

These senses are differentiated by the fact that while, in both cases, the typical subject is human, in the first the required object is also human whereas in the second it is inanimate. Such strategies give a very good idea of how a word should be used.

Other important information on the user perspective of the verb, e.g. whether socially reprehensible, possible, inherent, is also intentionally implied in the definitions. This is given in the examples above by the use of 'when' or 'if' as initial operators and the choice of 'you' or 'someone' as indicators of human arguments. For a discussion of the significance of the different Cobuild defining strategies, see Hanks (1987).

The LHS of the Cobuild definition (integrated by certain data from the RHS) thus contains information that cannot be found systematically in traditional dictionaries; in fact, such dictionaries provide this kind of information only occasionally, in the form of example sentences and very rarely as specifications within the definition text. By contrast, in Cobuild this information is usually

encoded in a consistent, coherent way. It is the linking of a number of elements, i.e. (i) the statement of meaning, (ii) the syntactic environment, (iii) the selectional preferences or restrictions on arguments, and (iv) information on the user perspective of the verb, which, on the one hand, is unique to Cobuild and, on the other, is of primary importance for the solution of many problems in NLP. We therefore decided to attempt to identify explicitly this type of implicit knowledge as it gives important cues for the identification of arguments and/or modifiers of lexical items with the most appropriate features. We think that the extraction and represention of such information is of particular relevance for the construction of computational lexicons for NLP applications because it should be possible to use it in a number of tasks or subtasks, such as parsing, word-sense disambiguation, subject/object assignment, PP attachment, etc. For this reason, suitably formalized, this kind of information should be very useful for systems for automatic analysis and generation of language.

Our aim in the first place has thus been to verify to what extent useful NLP-exploitable information could be extracted, on a large-scale, from this first part of the Cobuild definition statement. This is another point of divergence with previous work on the analysis of machine readable dictionary definitions. Although such investigations have gone into considerable depth, they have tended to examine particular subsets of definitions rather than the whole dictionary (see, for example, the work of the ACQUILEX project on words belonging to the "food" domain and on other linguistically interesting semantic subsets). On the contrary, as the methodology that we have devised to treat the LHS relies mainly on formal aspects of the definition, it can be applied extensively, throughout the entire dictionary.

The rest of this section is structured as follows. In the first part (3.2), we briefly outline the methodology adopted, motivating our decision to adopt two types of representation formats. In 3.3 we describe in detail our analysis of the definition data, and the development of the parser that extracts the information and maps it onto the Intermediate Template (IT). The next part (3.4) describes the conversion of the information from the IT into a TFS formalism; mention is also made of how and to what extent our lexical entries can be implemented under ALEP. In the conclusions, we illustrate various ways in which both our lexical representations (IT and TFS) can be used by human and machine users, and in particular describe how the syntactic-semantic information derived from the dictionary definitions could be exploited in procedures for dictionary and text sense disambiguation.

## 3.2 Methodology

The working strategy adopted by Pisa involved a three stage approach as follows:

i. Analysis of the parsed Cobuild definitions provided by Birmingham, extraction of lexical information, in particular syntactic-semantic and collocational data, and its representation on an Intermediate Template;
ii. Evaluation of the different types of information extracted with respect to its representability and its utility for NLP;
iii. Representation in a TFS formalism.

The decision to adopt this approach has had important consequences. It has resulted in two, considerably different, representation structures, one theory-neutral and human oriented (the IT), the other theory-dependent and intended principally for machine use. The main reason for employing the IT was that it is not possible to represent everything that can be usefully extracted in the TFS formalism nor to implement it in ALEP (the problems of using ALEP to represent lexical knowledge have already been pointed out in Montemagni (1992)). The IT gave us the opportunity both to evaluate our first results before attempting to further formalise them and also to store (in an explicit, interpreted way) all the information extracted in a computationally tractable fashion, so that information that is not immediately representable in TFS or in ALEP can be kept for future analysis and exploitation, if and when desired.

This choice is amply validated in the final section (3.5) where we show that both representations can be usefully employed in a wide range of applications. In the next part (3.3), we describe the first of these three stages of action; the second and third will be illustrated in 3.4.

## 3.3 Extraction: from the Birmingham output to the Intermediate Template

In this section, we will describe in detail the first stage of our work: analysis of the parsed definition statements received from Birmingham and the design and development of a specialised parser to extract syntactic, semantic and other useful information from them. The results of this stage are mapped onto the Intermediate Template. We have already explained the reasons which have led us to concentrate our main efforts on the LHS in the introduction. In fact, most of the information described regards this part of the definition. The extraction of information on the RHS has concerned almost exclusively the genus data.

### 3.3.1 Analysis of the LHS

Our analysis of the definitions was based in the first place on the Cobuild literature, i.e. on the description of the intentions of the Cobuild lexicographers and of the structure of the definitions (see Moon 1987; Hanks 1987; Sinclair 1987, 1991).

We thus made a series of initial hypotheses. The next stage was to consult the actual dictionary data to see whether it corroborated these hypotheses. Once we had adequately tested our first hypotheses against the data, we formulated a series of templates on which we could map the results we expected to obtain for each word class. In this section, we will present the templates for nouns, verbs, and adjectives, describing the types of information that we extract and the strategies used for extraction. We will begin by discussing the work regarding verb definitions.

#### 3.3.1.1 Verbs

We began our work by studying verbs as it appeared that, from the point of view of the kind of information we wanted to extract (syntactic/lexical), the most interesting data was contained in the Cobuild definitions for this word class.

To a large extent, the information which we can derive for verbs regards the subject, object, complement and preposition preferences of each word sense. We follow here the Cobuild philosophy, as stated by Hanks (1987), which is that "selection preference" for arguments is a far more appropriate concept than "selection restriction". We attempted to classify the selected arguments in terms of the features which they should bear and/or their typical semantic domain. The information we extracted is not only semantic but also syntactic, or better, a combination of the two. However, we feel that if our results are to be useful as input for other systems we must aim at producing entries which are as complete as possible. For this reason, an early project report by Montemagni et al. (1992) gave a first proposal for the representation of data extracted from Cobuild definitions in which orthographic, phonetic, and syntactic data were also considered. In the following, however, the discussion will be limited to syntactic-semantic and also some pragmatic data.

In fact, some data that we derive from the LHS has helped us to classify the verb type on the basis of certain inferences that can be made from the particular formulation of the definition. For example, we are particularly interested in testing whether distinctions of the if/when, you/someone type in the dictionary were actually reliable and consistent in practice. According to the Cobuild literature (see, for instance, Sinclair (1991)), the hinge "when" is adopted for animate subjects when the action of the verb can be considered as an inherent action or typical activity of the subject (e.g. the verb **reign** is defined by "When a king or queen **reigns**..." and **sneeze** by "When you **sneeze**...") or for inanimate objects when the action appears to be inherent in the nature of the object (e.g. "When the sun **rises**..."). "If", on the other hand, is used for actions which involve some kind of choice or are not inherent (e.g. "If a king or queen **abdicates**...", "If a meeting or trial **adjourns**..."). The "you/someone" alternation in the subject position is used to distinguish between human actions that are considered neutral and those that are not. The implication of "you" is that the sentence

expresses something that anyone might reasonably and normally do while "someone" is reserved for negative or unlikely actions. Other words used by Cobuild to imply particular features for the verb arguments are "somebody", "person", "people", "something", "things", etc. We have attempted to codify the information which can thus be inferred from the use of these particular definition strategies in order to be able to assess to what extent they can be considered reliable and/or useful knowledge for an NLP lexicon.

The information which we derive for verbs is shown in the following template and discussed in detail (3.3.3) [1]. Most of the information regards the LHS of the definition but of course the values for the GENUS_INFO attribute come from the RHS and will be discussed in the following section. It must be remembered that for the extraction of certain values the procedure must also analyse some parts of the RHS. This will be described in the section on the Parser below.

**TEMPLATE FOR VERBS**

```
GENERAL_INFO:
     DEF_No.:
     SENSE_No.:
     DEF_TYPE:
     LEMMA:
     ENTRY_INFO:
     GENUS_INFO:
     INFLECTION:
     GRAM:
     (SEC_GRAM:)*

PREFERENCE_INFO:
      VOICE:
     (INFERENCE:)
     (PHRASE_TYPE:)
     (SUBJ_INFO:)
     (OBJ_INFO:)
     (OBJ2_INFO:)
     (OBLIQUE_INFO:)
     (CLAUSE_INFO:)
     (ADJUNCT_INFO:)
     (TO-INF_INFO:)
```

Each of the INFO type attributes can be further expanded as shown in the examples below:

```
ENTRY_INFO: {ENTRY: value}
            {NORM_ENTRY : value V NORM_EXP_ENTRY: value}

GENUS_INFO:{PROV_SUPERORDINATE:value V PROV_SYNONYM:value}
           ({EXP_SUPERORDINATE:value V EXP_SYNONYM:value})
                 (IS-A     : ISA       : value
                             [DEFNO     : number
                              SENSE_NO  : number])*
                 (SYN      : SYNONYM   : value
                             [DEFNO     : number
                              SENSE_NO  : number])*
                 (PARTOF   : PARTPHR   : value
                              PART      : value
                             [DEFNO     : number
                              SENSE_NO  : number])*
                 (MEMBEROF: MEMBERPHR: value
                              MEMBER    : value
                             [DEFNO     : number
                              SENSE_NO  : number])*
                 (SETOF    : SETPHR    : value
                              SET       : value
                             [DEFNO     : number
                              SENSE_NO  : number])*
```

[1] '*' indicates 0 or more occurrence, '()' indicates optionality.

```
                    (ACTOF    : ACTPHR    : value
                                ACT       : value
                               [DEFNO     : number
                                SENSE_NO  : number])*)+
                          ... OTHERS TO BE IDENTIFIED ...

SUBJ_INFO:({SPECIFIC:value V TYPICAL:value V NULL})
          (SUBJ_FEATURES:features)
          (SUBJ_PREMOD:values)
          (SUBJ_POSTMOD:values)

OBJ_INFO:({SPECIFIC:value V TYPICAL:value V NULL})
         (OBJ_FEATURES:features)
         (OBJ_PREMOD:values)
         (OBJ_POSTMOD:values)

OBLIQUE_INFO:({OBLIG_PREP: prep V PREF_PREP: prep })
             ({SPECIFIC:value V TYPICAL:value V NULL})
             (OBLIQUE_FEATURES:features)
             (OBLIQUE_PREMOD:values)
             (OBLIQUE_POSTMOD:values)
```

It can be seen from the template that we are extracting two types of data. In the first block of General Information, we mainly have data that is already given explicitly in the syntactically parsed definitions received from Birmingham. We simply extract this and store it directly for future use. [2]

The second block is the set of information that is contained only implicitly in the LHS of the definition and which we derive by our analysis. This block consists of information which classifies the verb with respect to the preferred features of its different arguments, to its preferred form, and gives a value to the type of action represented in the given sense. In the following, we will give details and examples of the different attributes in the template and their possible values, and briefly illustrate the types of procedures used to extract the information and map it onto the template.

### GENERAL INFORMATION

As stated, this first block of Attributes mainly regards values which are extracted more or less directly from the Birmingham input and which are necessary to us for our further treatment and management of the data. Apart from SEC_GRAM, all these Attributes are obligatory.

DEF_No.: Extracted directly from the Birmingham input. Used to univocally identify each definition.

SENSE_No.: Extracted directly from the Birmingham input. Identifies the particular sense of the entry, as given in the dictionary.

DEF_TYPE: Extracted directly from the Birmingham input. Identifies the particular definition strategy used by the Cobuild lexicographers when writing the definition statement. These strategies are defined in (Allport et al. 1993a and b).

ENTRY: Normally, the value of the entry item is read from 'head' from the Birmingham input and written in the ENTRY attribute. The base form is stored in LEMMA. However, in some cases this is not sufficient. For example in the case of verbal phrases (where grammar is PHRASE and def_type=1), the value of LEMMA is not equivalent to the base form of the entry item. In this case, we assume the first item in ENTRY to be a verb and search its base form (using a procedure that is very similar to the one described below in 3.3.2.1 for deriving the base form of the genus item) in order to obtain a value for NORM_ENTRY. Thus, **place** 16, PHRASE, defined by "When something takes **place**, it happens", is processed as follows:

[2] We refer to the input data in the following as the Birmingham input and indicate fields from this input using single inverted commas whereas the fields in our template are referred to using capital letters. There is not necessarily a direct one-to-one relationship between the values in the Birmingham data and ours; for example, the Birmingham data gives 'lemma' which contains the inflection values, whereas our LEMMA contains the value for the base form of the lexical item being defined.

```
lemma          : place
entry_info     : entry                    : takes place
                 norm_entry               : take place
```

At times it is necessary to reconstruct the value for ENTRY, e.g. in the case of nouns and adjectives which have been recognized as part of a verbal phrase (see section 3.3.1.4 below.) In such case, NORM_EXP_ENTRY contains the "normalized" value of the reconstructed entry item. This value is normalized by (i) assuming the first item to be a verb and searching the base form, and (ii) eliminating particular strings, e.g. pronouns, personal adjectives, etc. Thus, for **attention 1**,defined by "If you give something your **attention**, ...", our first value for our reconstructed entry item was: **give something your attention**. The normalized value for NORM_EXP_ENTRY is now **give attention**.

LEMMA: Contains the base form of the lexical item being defined (the dictionary headword). This information is derived from the first item contained in the list under 'lemma' in the Birmingham input. The distinction between ENTRY and LEMMA is necessary as the lexical item being defined is not necessarily identical to the headword of the entry. For example, we have three definitions under the LEMMA **abide**, where the values for ENTRY are **can't abide**, **abide** and **abide by**. The value under LEMMA permits us to relate each of these three examples back to its base lemma. All the following attributes in the template refer to values of ENTRY rather than of LEMMA.

GENUS_INFO:PROV_SUPERORDINATE V PROV_SYNONYM:
In a first step, we simply rewrite the Birmingham values given for this field under 'superordinate' or 'synonym' in the RHS. The data is then analysed as described in the section on "Treating Genus Information" in 3.3.2.1 below.

INFLECTION: The Cobuild dictionary gives the total inflection for the entire entry immediately after the headword rather than separately for each grammatical category. This strategy is economic with regard to space but of course relies on the users' knowledge for the correct identification of the inflection of any particular homograph of the lemma. We thus have to analyse the 'lemma' field in the Birmingham input in order to derive the correct inflection, depending on the POS being treated. For non modal verbs we currently simply rewrite the values given in this field. For nouns and adjectives, we must use a different strategy (see below).

GRAM and SEC_GRAM:
The value is taken from the 'grammar' field in the Birmingham input. However, when the grammar field contains one or more "or" this indicates that there is more than one grammar possibility for this particular sense. In this case, the first possibility is written in GRAM, the second (or others) in SEC_GRAM. This distinction is needed by our procedure in order to identify the correct template for the analysis (for details, see the section describing the Parser).

PREFERENCE INFORMATION

In this second block of data, we write the values that we derive from the analysis of the LHS of definitions performed by our parser. Apart from VOICE, all of these attributes are optional.

VOICE: For each word category or subcategory, Cobuild has an unmarked explanation strategy against which marked strategies can be contrasted. For transitive verbs, for example, the unmarked strategy is the active voice against which some verbs are defined using a passive construction to make their preference for this construction clear, e.g. "If someone or something **is acclaimed**, ...". Similarly, other verbs are defined using the progressive voice, e.g. "If someone or something **is acting up**, ..." This attribute thus gives a value for the preferred voice or tense used in this sense of the verb, as shown by the corpus. Possible values: Active/Passive/Progressive.

The value is derived by analysing the value of 'head1' in the Birmingham input and comparing it with the contents of 'lemma' which contains inflection information. If 'head1' contains "is" or "are" and the following string is equal to the third item in the contents of 'lemma', then VOICE: Progressive. If 'head1' contains "is" or "are" and the following string is not equal to the third item in the contents of 'lemma', then VOICE: Passive. Otherwise, VOICE: Active.

INFERENCE: As we have already mentioned, one of the most interesting features of the definitions in the Cobuild dictionary is the use of "if/when" and "you/someone" in an attempt to provide the user implicitly with an idea of the "perspective" of the verb. We thought that it would be both useful and interesting to derive this kind of information which can be used to classify in some way the perspective the user has on the verb.

For definitions where Def_Type = 1, the value of Inference is derived from the value contained in field 'hinge' together with that contained in 'co-text1'. Here below, we give some examples:

— Hinge = if
  co-text1: Does not contain one of the values in List2 (e.g. you, someone, people)
  Value: likely
— Hinge: when
  co-text1: Does not contain one of the values in List2 (e.g. you, someone, people)
  Value: inherent
— Hinge = if
  co-text1 contains "you"
  Value: possible, likely
— Hinge = if
  co-text1 contains "someone"
  Value: possible, negative/unlikely
— Hinge = when
  co-text1 contains "you"
  Value: inherent, likely
— Hinge = when
  co-text1 contains "someone"
  Value: inherent, negative/unlikely
— Hinge = if
  co-text1 contains one of the values in List3 (e.g. group, people, etc.)
  Value: possible, collective
— Hinge = when
  co-text1 contains one of the values in List3 (e.g. group, people, etc.)
  Value: inherent, collective

For definitions where Def_Type=3, Hinge = "if" or "when", and 'Projection' contains "you say that", "you describe", Value: Subjective. Currently, for other values of 'hinge', we make no inference.

PHRASE_TYPE: Indicates the kind of phrase we are treating. So far we have only treated verbal phrases. If DEF_TYPE = 1, then the value of PHRASE_TYPE is verbal. The definition is then analysed in the same way as that of a verb.

### ARGUMENT PREFERENCE INFORMATION

The main type of information that we extract from the LHS of the definitions for verbs regards syntactic and semantic information on the preferences of their arguments. Examples of how this

type of information is structured have been given above in the Template. Let us now explain the meaning of the different attributes and the kind of values they can take.

The values for the preference data are acquired from an analysis of the data contained in 'co-text1' and 'co-text2' in the Birmingham input. The co-texts contain information on the preferred arguments of the verb, presented at different levels of generality, to give the user as clear as possible an idea of the kind of arguments required. We have identified three levels of arguments: (i) the almost obligatory argument of the "If a horse **gallops**..." type, (ii) the very general argument of the "If you **copy** something ..." type and (iii) an intermediate level in which the range of arguments is restricted to a particular area or areas, e.g. "If something such as a path or river **forks**...".

In the first case, the definition is stating that the preferred subject of the verb **gallop** is normally a horse; the inference is that when using this verb the subject slot must be filled by something that is definable as a horse (e.g pony, stallion, mare, Black Beauty, and so on). In our template, this kind of argument has been tagged as SPECIFIC.

In the second case, Cobuild uses a finite list of words to express various kinds of very general arguments. Depending on the particular word used, different features can be derived for the arguments of the given sense of the verb in question. The example above implies that **copy** in this sense requires a human subject but an inanimate object. From our analysis of the data, we have constructed a list of these words.Thus, when our procedure finds one of the words contained in this list, the relevant features, as given in the list, will be derived for that argument of the verb. The possible values for features are currently:

```
FEATURES: Animate, Hum-m, Hum-f, Hum-Coll, Coll, Animal,
Plant, Count
```

In this stage of extraction, the features are simply stored as above in a flat list. They are modelled in a hierarchical structure in the TFS representation.

In the third case shown above, the definition implies that the subject of the verb **fork** in this sense will be inanimate and typically belongs to the semantic domain(s) exemplified by the particular words which appear after the "such as" trigger. In this case, we derive general features, as above, for the argument from the general word used (here it is "something") and write the strings following the trigger as values of TYPICAL. The possible strings used as triggers are contained in another List.

SUBJ_INFO also contains SUBJ_PREMOD and SUBJ_POSTMOD where SUBJ_PREMOD is used for any strings preceding the identified subject and SUBJ_POSTMOD is used for any strings following it. This distinction closely models how these modifiers are syntactically realised within the definitions. We have made it in order to facilitate future processing. In fact, whereas the data which we tag as PREMOD is normally an adjective or attributive noun, frequently the data which is tagged as POSTMOD is a clause which will probably be subjected to further analysis later on.

OBJ_INFO is used for values for the object of the verb and expanded as above for SUBJ_INFO

OBJ2_INFO is used for values for second object when the verb is used with two objects. The value of 'grammar' will be VB WITH OBJ AND OBJ. The values for this Attribute are as shown above for SUBJ_INFO and OBJ_INFO.

OBLIQUE_INFO

is expanded as follows:

```
OBLIQUE_INFO:({OBLIG_PREP: prep V PREF_PREP: prep V NULL})
             ({SPECIFIC:value V TYPICAL:value})
             (OBLIQUE_FEATURES:features)
             (OBLIQUE_PREMOD:values)
             (OBLIQUE_POSTMOD:values)
```

OBLIG_PREP is used when the value of the 'grammar' field for the verb specifies the use of a preposition, e.g. the ninth sense of **break** is categorised as "VB WITH for" and "for" will be the value of OBLIG_PREP.

PREF_PREP instead is used when the value of a preferred preposition for the verb is not specified in 'grammar' but can be derived from an analysis of the LHS of the definition. For example, with sense 2 of **advance**, defined as "If you advance in something you are doing...", the preposition "in" is not specified in the grammar field but is derived as value of PREF_PREP.

The distinction between OBLIG_PREP and PREF_PREP is thus determined by the way in which these prepositions appear in the lexical entries in the dictionary and reflects a parallel distinction made by the lexicographer, clearly on the basis of corpus evidence. However, on a first examination of the data, it is not always evident why in one case a particular preposition has been classified explicitly as part of the grammar field whereas another, which appears to be equally "obligatory" can only be inferred from the definition. For example, compare the first two senses of **listen**:

1 VB
If you **listen** to someone who is talking or to a sound, ...

2 VB WITH 'for'
If you **listen** for a sound, ...

Once we have processed all the definitions, it will be easy for the lexicographers to compare and evaluate this kind of data, correcting it if necessary.

#### 3.3.1.2 Nouns

The noun definitions in the Student's dictionary are not so rich in contextual information as those for verbs. Noun collocates tend to be many and various and do not necessarily occur in any regular structural relationship with the noun itself. Unlike verbs, it is not true that lexical selection preferences are generally associated with particular syntactic structures (see Hanks, 1987). Thus frequently it was not possible for the Cobuild lexicographer to contextualise the LHS of the definition in any way. Thus we find definitions of the following type: "A **bank** is a place where you keep your money in an account", and "A **bank** is also the raised ground along the edge of a river or lake", where it is entirely the semantic content of the RHS that distinguishes between the two senses of **bank**.

However, on the other hand, the corpus did show that there were many cases where the combined collocational and syntactic preferences of a noun provided a basis for the adoption of an explanation strategy to give the user implicit information on such preferences. For example, one sense of **administration** is defined by "The **administration** of a company, an institution, or a country is ..." to indicate that **administration** in this sense typically selects an argument introduced by the preposition "of" which is normally followed by words such as "company", "institution","country".

We have thus used similar strategies to those developed for analysing the LHS of verb definitions to extract, where possible, information on the syntactic and collocational preferences for nouns. So far, we have only tagged this information as collocational (*COLLOC_INFO) without attempting to further define its role (as has been done for the collocational information for verbs - tagged in terms of their syntactic role). In a second stage, we intend to analyse this data in more depth.

Here below, we illustrate the template and describe the information we are currently deriving for nouns.

GENERAL TEMPLATE FOR NOUNS

```
GENERAL_INFO:
     DEF_No.:
     SENSE_No.:
     DEF_TYPE:
     GRAM:
     (SEC_GRAM:)*
     LEMMA:
     ENTRY_INFO:
```

```
        GENUS_INFO:
        INFLECTION:
        (PHRASE_TYPE:)

    PREFERENCE_INFO:
        (FORM:)
        (PRECOLLOC_INFO:)
        (POSTCOLLOC_INFO:)
```

Each of the INFO type attributes can be further expanded as shown in the examples below:

```
GENUS_INFO:{PROV_SUPERORDINATE:value V PROV_SYNONYM:value}

PRECOLLOC_INFO: values
POSTCOLLOC_INFO:({OBLIG_PREP:prep V PREF_PREP:prep })
                ({SPECIFIC:value V TYPICAL:value V NULL})
                (COLLOC_FEATURES:features)
```

*Attributes and Values*

Only the information types which differ from those described for the verb template will be discussed.

INFLECTION: For uncount, mass and collective nouns, no value is read. For count nouns, the first two values of the 'lemma' field of the Birmingham input are read.

FORM: In the definition statements for count nouns, the unmarked strategy is to use the singular form with the indefinite article, e.g. "A **bag** is ...". However, if corpus evidence suggests that a particular noun is commonly used in the plural form then this is reflected in the definition, e.g. "**Beads** are ...". We tag this information explicitly by assigning the value "plural" to the FORM attribute.

*COLLOC_INFO:
This tag cover two attributes: PRECOLLOC_INFO and POST_COLLOC which contain the noun collocates found, respectively, in 'co-text1' and 'co-text2' of the Birmingham input for which, so far, we do not provide a more fine-grained interpretation.

PRECOLLOC_INFO:
used for any string which precedes the entry item (found in co-text1). This attribute does not have a complex structure as its values only range over adjectives or attributive nouns, which we are unable to differentiate as our input does not provide POS tagged data. For example, sense 11 of **cover** is defined by "Bed **covers** are ..." In this case, the value of PRECOLLOC_INFO will be "bed".

POSTCOLLOC_INFO:
used for any string or sequence of strings which follows the entry item (found in co-text2). The information is analysed in much the same way as described above for verb arguments. Again, we have identified three levels of generality for the preferred collocates: very specific collocates as in "A **box** in a theatre is ...", very general ones of the "An **abundance** of something..." type, or an intermediate level of generality where the typical area is indicated e.g. "A **book** of something such as stamps, matches, or tickets ..." In the same way as for verbs, in the first case we derive "theatre" as value for SPECIFIC, in the second case we derive values for COLLOC_FEATURES from the features assigned to "something", and in the third case we derive "stamps", "matches", and "tickets" as values of TYPICAL and values for COLLOC_FEATURES from "something". The preposition is assigned to OBLIG_PREP when this information is specified in the 'grammar' field, otherwise it is assigned to PREF_PREP, similarly as for verbs. In the three cases above, the preposition "in" or "of" will be assigned as value for PREF_PREP.

Another type of noun collocate is exemplified by the following piece of definition for one sense of **bunch**: "A **bunch** of bananas, grapes or other fruit is..." In this contextualization of **bunch** we are

given the information that this sense of the word is typically used with "bananas" and "grapes" and "bananas" and "grapes" both belong to the "fruit" domain. The problem is that it would also seem possible to infer that **bunch** could also be generally collocated with "fruit" whereas, correctly, it can only be collocated with types of fruit which have the particular characteristic of bananas and grapes, i.e. they grow in bunches. We need to examine more examples of this type before deciding how this information should be treated. At the moment, "bananas", "grapes" and "fruit" are all acquired as values of specific, thus loosing the information that "bananas and grapes are fruit (potentially important for sense matching) and apparently acquiring (incorrectly) the information that we can have a bunch of any fruit.

#### 3.3.1.3 Adjectives

Similar strategies have been used to analyse definitions for adjectives. The template adopted is shown below. However, we have spent less time on studying this word class and may well alter and/or expand this template as our analysis becomes more exhaustive.

GENERAL TEMPLATE FOR ADJECTIVES

```
GENERAL_INFO:
     DEF_No.:
     SENSE_No.:
     DEF_TYPE:
     GRAM:
     (INFLECTION:)
     LEMMA:
     ENTRY_INFO:
     GENUS_INFO:
     (PHRASE_TYPE:)

PREFERENCE_INFO:
     (COLLOC_INFO:)
```

For which we have the following expansions of values:

```
GENUS_INFO:{PROV_SUPERORDINATE:value V PROV_SYNONYM:value}

COLLOC_INFO:({SPECIFIC:value V TYPICAL:value})
            (COLLOC_FEATURES:features)
```

The attributes and their values are very similar to those described above for nouns. We are currently extracting information on the type of noun collocate which is qualified by a given adjective. In the case of adjectives too, the Cobuild definitions present different levels of generality for the preferred collocates. Thus, we have examples ranging from very specific noun collocates, e.g. "A **fizzy** drink is..." or "**Floral** cloth, paper or china has ..." to very general indications such as "Someone who is **livid**...", "An **effusive** person is ...", "If you are **flabbergastered**...". In the first cases, we derive values for the SPECIFIC attribute, in the second we infer values for the COLLOC_FEATURES attribute. With reference to the values for INFLECTION, if the 'lemma' field of the Birmingham input contains more than one item, then the first three values are read. Otherwise, no value is read.

Definitions for adjectives fall into three main groups: def_types 1, 2, and 3.

*Def_type 3 Adjectives*

Def_type 3 definitions consist of two phrases linked by a hinge at the beginning of the RHS. The hinge is normally "is", "are", "means". From our examination of these definitions, we find that when the hinge used is "means" then the definition normally regards the adjective directly, i.e. these are definitions of a more traditional type. For example, we have:

4 ADJ
**Low** means small in amount, value, or degree.

**3** ADJ
**High** also means great in amount, degree, or intensity.

**5** ATTRIB ADJ
**High** also means advanced or complex.

with the following genus terms tagged on our Birmingham input: small, great, advanced or complex. However, when we have the hinges "is" or "are", what is defined, at least explicitly, is a noun with the property attributed by the collocating adjective:

**6** ADJ
A **high** position in a profession or society is an important one.

**9** ADJ
A **high** sound is close to the top of a range of notes.

**13** ADJ
A **low** light is dim rather than bright.

**14** ADJ
Someone who is feeling **low** is unhappy.

with provisional genus terms tagged by Birmingham as: important, close, dim, and unhappy.

We are still considering the implication of this difference and the best way to represent it. However, it is important to note that in both cases, the PoS of the genus term is the same as that of the entry item.

Our parser currently gives only the following results for Def_type 3 adjectives. In the first case:

```
def_no        : 13009
sense_no      : 3
def_type      : 3
gram          : ADJ
lemma         : high
entry_info    : entry              : high
genus_info    : prov_superordinate1: great
                is-a1              : great
inflection    : high higher highest
```

and

```
def_no        : 13011
sense_no      : 5
def_type      : 3
gram          : ATTRIB ADJ
lemma         : high
entry_info    : entry              : high
genus_info    : prov_synonym1      : advanced
                prov_synonym2      : complex
                syn1               : advanced
                syn2               : complex
inflection    : high higher highest
```

whereas, in the second case, we also present the information on the noun collocates as follows:

```
def_no        : 13015
sense_no      : 9
def_type      : 3
gram          : ADJ
lemma         : high
entry_info    : entry              : high
genus_info    : prov_superordinate1: close
                is-a1              : close
inflection    : high higher highest
colloc_info   : colloc1            : sound
```

(Certainly, this example highlights one of the problems of any procedure which attempts to automatically identify the genus term in a definition. Probably the correct genus for **high 9** is "close to the top" but, at the moment, it is only possible to recognize this by means of a manual intervention.) or

```
def_no        : 13012
sense_no      : 6
def_type      : 3
gram          : ADJ
lemma         : high
entry_info    : entry              : high
genus_info    : prov_synonym1      : important
                syn1               : important
inflection    : high higher highest
colloc_info   : colloc1            : specific : position in a
                                                profession
                colloc2            : specific : position in a
                                                society
```

The function of prepositional phrases of the type shown in this last example are now being studied in order to treat them appropriately.

*Def_type 1 Adjectives*

The definitions for adjectives which have been classified in the Birmingham input as being of Def_type 1 are very different from the Def_type 3 set. In these cases, although the formal headword is an adjective, the definition really regards a verbal phrase. This case is discussed below in section 3.3.1.4.

*Def_type 4 Adjectives*

The last class regards adjectives tagged by Birmingham as def_type 4. Type 4 definitions have two parts but no hinge. In this group, we have definitions of the following type:

**1** ADJ
A **flexible** object or material can be bent eas[illegible]ithout breaking.

**1** ADJ
A **high** structure or mountain measures a great amount from the bottom to the top.

**1** ADJ
Something that is **low** measures a short distance from the bottom to the top.

where the meaning of the adjectives is in each case paraphrased with a VP, tagged by Birmingham as provisional superordinate: can be bent, measures a great amount, measures a short distance.

Similarly to def_type 3 adjectives with hinge=is, the definition does not concern the adjective in isolation but considers it together with its modified noun(s). (The difference in specificity in the definitions for **high** and **low** is interesting, and reflects a real difference in usage.) However, in each case the genus term is a verbal phrase. We are now considering the most appropriate way to treat definitions of this type. At the moment, our parser processes them as shown below but we feel that the VP must be further analysed:

```
def_no        : 13006
sense_no      : 1
def_type      : 4
gram          : ADJ
lemma         : high
entry_info    : entry              : high
genus_info    : prov_superordinate1 : measures a great amount
                is-a1               : measure a great amount
inflection    : high higher highest
colloc_info   : colloc1            : specific : structure
                colloc2            : specific : mountain
```

#### 3.3.1.4 Processing nouns and adjectives as verbs

The procedure which analyses the definitions begins by examing the POS contained in the 'grammar' field of the Birmingham input and the value assigned to 'Def_type' in order to determine the correct processing strategy to be adopted. The different definition types used by Cobuild are discussed in (Allport et al. 1993a and b). In the definitions we have studied so far, noun definitions are generally assigned to Def_Type 3. However some noun definitions are of Def_Type 1, which is characterized as being of the "if/when" type and is typically used for verb definitions, e.g. "When you focus a **camera** you ...", "If you follow someone's **instructions**...". We found that, when this defining format is used for nouns, the noun defined appears in fact as argument of the verb which is head of this part of the definition. For example, the two senses of **attention** in the Student's dictionary are both classified as uncount nouns and are defined as follows:

1 UNCOUNT NOUN
If you give something your **attention**, you look at it, listen to it, or think about it carefully

1 UNCOUNT NOUN
If something is getting **attention**, it is being dealt with

whereas "If you **pay attention** to something, you watch it, listen to it or take notice of it" is listed at the end of the entry for **attention**, as a Phrase, with **pay attention** evidenced as the headword.

Examining these definitions we found that, in all three cases, the RHS superordinate or synonym was a verb or verbal phrase and that, although the formal headword in the first two cases was a noun, what was really being defined in every case was a verbal phrase. Thus, when we find a noun definition with Def_Type 1, we construct the value for ENTRY by taking not only the value found in 'head' in the Birmingham input but also preceding it with the contents of 'co-text1'. Thus for the first two examples above, our constructed values for ENTRY are **give something your attention** and **getting attention**. At a later stage, the values **give attention** and **get attention** will be derived from this data. The entry is then tagged explicitly as being a verbal phrase, and we analyse it in the same way as for verbs. In the results produced by our parser for the three definitions cited above, it can be seen that, by processing the definitions in this way, the formal difference between them maintained in the dictionary disappears and instead their similarity is evidenced.

```
def_no              : 1498
sense_no            : 1
def_type            : 1
gram                : UNCOUNT N
lemma               : attention
entry                                   : give something
   your attention
genus_info          : superordinate3    : think about
                      superordinate2    : listen to
                      superordinate1    : look at
phrase_type         : verbal
voice               : active
infererence         : possible likely
subj_info           : subj_features1    : +anim, +hum

def_no              : 1499
sense_no            : 2
def_type            : 1
gram                : UNCOUNT N
lemma               : attention
entry                                   : is getting
                                          attention
genus_info          : synonym1          : is being dealt
                                          with
phrase_type         : verbal
voice               : progressive
inference           : possible
subj_info           : subj_features1    : -anim
```

```
def_no              : 1502
sense_no            : 2
def_type            : 1
lemma               : attention
entry                                   : pay attention
genus_info          : synonym3          : take notice of
                      synonym2          : listen to
                      synonym1          : watch
inflection          : attention
gram                : PHRASE
phrase_type         : verbal
voice               : active
infererence         : possible likely
subj_info           : subj_features1    : +anim, +hum
obj_info            : obj_features1     : -anim
```

It appears to us that this use of def_type1 to define nouns occurs when we have nouns used together with so-called "support" or lexically empty verbs, see "give" and "get" in the examples above. The way in which we treat this phenomenon is described below in 3.3.2.1.

When we examined the definitions for adjectives, we found exactly the same situation: adjectives with def_type 1 had a verb or verbal phrase as a genus term rather than another adjective. We have treated them in the same way as described for nouns. Thus, for example, the dictionary entry for **dependent 1** is: "If you are **dependent** on someone or something, you need them to survive". This definition has been classified by Birmingham as def_type 1 and **need** has been tagged as superordinate. Our parser, therefore, processes the definition as a verbal phrase, reconstructing the value for entry as: **be dependent** with "on" tagged as the preferred preposition.

We feel that this kind of information should be interesting because it exploits another special feature of Cobuild, i.e. the attention paid by Cobuild to phrases in their defining strategy and, at the same time, provides explicit information on phrasal constructions which would be of great importance to NLP lexicons. The fact that our treatment reveals the similarity between definitions treated by the dictionary as Nouns and Adjectives and others classified as Phrases may be found useful by the Cobuild lexicographers who may want to re-examine their treatment of these items in order to give information on the existence of verbal phrases explicitly, rather than implicitly as at present.

### 3.3.2 Analysis of the RHS

Our initial intention in our study of the RHS was to attempt to define strategies to extract semantic information from the genus terms and differentiae, similarly to what has been done in other projects. However, our first analysis evidenced a number of problems. In fact, the superordinate and discriminator data found in the Cobuild definitions are considerably different from those of more traditional dictionaries. In particular, we found that in many cases, depending on the defining formula adopted, it was not possible to identify a superordinate and construct significant taxonomies directly. For example, compare the Cobuild definition for **frenzy**, "Someone who is in a **frenzy** is very excited and violent or uncontrolled", with that of OALDCE: "violent excitement". In the Cobuild definition, the word has been contextualized and thus the definition itself also gives the typical usage, whereas, in OALDCE, this information is given in an example following the definition: "In a frenzy of despair/excitement". From the OALDCE definition, a superordinate (excitement) and a discriminator (violent) can be extracted directly but the contextualization in Cobuild means that the noun **frenzy** has been described in terms of three adjectives "excited", "violent", "uncontrolled" (actually the definition statement regards a person in a frenzied state rather than the word "frenzy" in isolation).

In other cases, when it is possible to derive the superordinate directly from the definition, frequently the term used is so general that it is difficult to extract detailed and useful information from it. For example, compare the way in which **frieze** is defined in these two dictionaries: Cobuild tells us that "A **frieze** is a long, narrow, decorative feature along the top of a wall". However, "feature" is such a general term, that all the semantic information concerning **frieze** must be derived from

the adjectives used to qualify it. Instead, OALDCE describes **frieze** as an "ornamental band or strip along a wall (usu at top), e.g. ...", where the concepts of "long" and "narrow" are carried by the use of band or strip.

These differences mean that the information that can be extracted from this side of the definition is somewhat different from that extractable from the dictionaries we have analysed previously in other projects and imply that different strategies must be studied if meaningful information is to be derived.

#### 3.3.2.1 Treating the genus information

For the moment we have simply performed a preliminary analysis of the genus term, postponing any more detailed examination of the rest of the RHS to a later date. We have thus focussed our attention on:

- making a first classification of the different types of semantic relations represented by the genus terms, e.g. hyperonymy, synonymy, set-of, part-of, etc.;
- developing procedures to recognize and process the different kinds of genuses automatically;
- reconstructing the genus term when in the Birmingham input data we find a lexically empty term tagged as superordinate or synonym;
- recognizing and tagging following particles so that they can be attached to the genus term if necessary; in this way, we avoid losing potentially important information.

*Identifying Semantic Relations*

In our input data, on the RHS side of the definition, the values recognized by Birmingham as genus data were tagged as either 'superordinate' or 'synonym'. As a preliminary step, these values were rewritten, with only a simple preanalysis, as provisional values in our Intermediate Template. The only treatment was to divide the complex genus terms (identified by the presence of "or"). For example, for **act 9**, where the form of the headword given in the entry: is **acting up**, the Birmingham input gave as superordinate: are not behaving or working. On the IT, this data is divided as follows:

```
genus_info: prov_superordinate1: are not behaving
            prov_superordinate2: are not working
```

Our first task was thus to process the genus data in various ways in order to derive more significant values.

In fact, different types of semantic relationships between genus term and definiendum can be recognized, apart from the superordinate and synonym relationships already tagged explicitly by Birmingham. We have thus studied the definition data in order to identify other meaningful relationships. Our starting point for this kind of classification has been the detailed study on different kinds of taxonomic data already made for the ACQUILEX project (see Hagman (1991)). When the classification has been completely finished it will be interesting to see how far the results of this study can be applied to Cobuild data or if different types of superordinates will emerge from processing Cobuild definitions.

The strategy used by Birmingham to automatically parse the definitions is based on the recognition of particular words or sequences of words as delimiters between "chunks" of information. This strategy had to be carefully considered when we attempted to interpret this parsed output. For example, the method used by Birmingham to distinguish between superordinate or synonym depends on the presence or absence of a following discriminator, independently of the type of discriminator found. However, this can lead to apparent inconsistencies in the data. For instance, we found: **absorption**, sense 2, synonym: the action of absorbing something but **abstention** 1 superordinate: act of not voting. We had to standardise the treatment of this kind of data by recognizing particular patterns as equivalent and processing them in the same way. In many cases, our analysis meant that we recognize a synonym where Birmingham had tagged a superordinate. This is illustrated below in the section on expanding genus terms which appear as lexically empty words.

Following the technique described by Hagman (op cit), we searched the provisional genus data for the presence of certain trigger words which permitted us to recognize that the definiendum is in a specific semantic relationship, other than an is-a relationship, with its defining term.

We look for the following pattern: trigger word + of + string. Consider the following examples from the Birmingham definition data:

```
area, sense 1, superordinate: part of a city, country, or the
                              world
country, sense 1, superordinate: one of the political areas
administration, sense 2, superordinate: the range of
                                        activities
assembly, sense 3, synonym: process of fitting parts together
communication, sense 2, superordinate1: activity of giving
                                        information
                        superordinate2: process of giving
                                        information
```

In these cases, the presence of a trigger word (part, one, range, process, activity) means that we are able to recognize the following relationships: PARTOF, MEMBER OF, SETOF, ACTOF, PROCESSOF. It is the syntactic head of the noun phrase that denotes a particular semantic relation between the definiendum and the NP following the preposition "of". In this way, we resolve the kind of inconsistency noted above.

For example, if our parser finds the strings "activity of", "action of", "act of", then the ACTOF relationship is recognized. Thus, the following genuses:

```
synonym:  the action of absorbing something
superordinate: act of not voting.
superordinate:  activity of giving information
```

will be processed as follows:

```
entry_info: entry              : absorption
genus_info: prov_synonym       : the action of absorbing
                                 something
            actof              : actphr : action of
                               : act    : absorbing something

entry_info: entry              : abstention
genus_info: prov_superordinate : act of not voting
            actof              : actphr : act of
                               : act    : not voting

entry_info: entry              : communication
genus_info: prov_superordinate : activity of giving
                                 information
            actof              : actphr : activity of
                               : act    : giving information
```

where "actof" denotes the type of relationship; the value of "actphr" is the actual trigger string which allows us to recognise the relationship; the value of "act" is the current genus.

Examples of the different values recognized are given in the General Template of 3.3.1.1. In the Genus_Info group on the Template, the first attribute is obligatory and contains the values read from the Birmingham input, and the others are optional and contain the results of our analysis of the genus information.

*Deriving the Base Form*

As already stated, the values for the genus information received from Birmingham were written in the provisional superordinate/synonym field. However, if we are going to be able to reuse this data, we must derive its base form or lemma. We do this by matching the form given in the provisional attribute against the inflection fields for the relevant entry in the on-line dictionary (identified

normally using a masking and nearest match technique). When a match is found, the first form or lemma is extracted and written as the base form value under the appropriate genus_info attribute (see the Template in 3.3.3.1), or others that will be added as our study progresses. Here below, we give a brief idea of the other rules that have been written to derive the base form for the genus term.

For example, for verbs, phrasal verbs and verbal phrases, when the genus value consists of more than one item, then the first item is assumed to be a verb and the base form is searched as described above.If the first item is "is" or "are", we check to see whether the form used for the entry is passive (PASSIVE will be a value for the gram or the voice attributes). If so, then we eliminate "is/are" and search the base form of the verb. Thus, for example, "is sent" is rewritten as "send". Otherwise, "is", "are", and also "there is", "there are" are rewritten as "be", "can" is rewitten as "be able to", "can't" as "not be able to", and so on. Thus, for **apply 4** we have:

```
def_type      : 1
lemma         : apply
entry_info    : entry              : apply
genus_info    : prov_superordinate : is relevant
                is-a               : be relevant
```

For nouns, if the genus term ends in a preposition, this will be removed and attached to the immediately following discriminator which should be a noun phrase. For example, beam, sense 2, has as superordinate: line of. We attach the final preposition "of" to the following discriminator "light", and there will then be an improved symmetry between the two parts of the definition statement. Other rules are being implemented.

*Expanding Genus Terms which appear as Lexically Empty Words*

In the data received from Birmingham, we have found a relatively high number of verb definitions in which the value of the superordinate or synonym attribute appears to be a so- called lexically empty or support verb, in particular, "make". Other examples that have been tested are "get", "be", "go", "have", "take". We decided to process these definitions in order to derive more meaningful genus terms. We thus constructed a new value for the genus term by attaching to the base form (the lemma) of the support verb the first following discriminator. The new value was written in exp_superordinate or exp_synonym depending on the presence or absence of another following discriminator, as shown in the following examples. Thus, for **allow 4** we had the following piece of input from Birmingham:

```
    (superordinate
         '(makes)
    )
    (match2
         '(it)
    )
   (discriminator
        '(possible)
   )
)
```

which our parser processes as

```
genus_info:prov_superordinate:makes
           exp_synonym:make possible
```

as there is no longer a following discriminator and we thus derive a relation of synonymy. Whereas, the definition for **control 3** from Birmingham included

```
    (superordinate
         '(make)
    )
```

```
(match1
     '(it)
)
(discriminator
     '(work)
)
(discriminator
     'in
     .....
     .....
```

and we have processed as

```
genus_info:prov_superordinate:make
           exp_superordinate: make work
```

where the genus term remains as superordinate as, in this case, there is still a following discriminator. Other examples are:

```
cause, 3, genus_info:prov_superordinate:make
                    exp_synonym: make happen
exceed, 2, genus_info:prov_superordinate:go
                    exp_synonym: go beyond
improve, 1, genus_info:prov_superordinate:gets
                    exp_synonym: get better
operate, 3, genus_info:prov_superordinate:make
                    exp_synonym: make work
reduce, 3, genus_info:prov_superordinate:make
                    exp_synonym: make smaller
```

We think that this reconstruction is interesting, not only for NLP purposes but also for the lexicographer, as it highlights the existence of this kind of verbal phrase formed by a support verb and noun or adjectival phrase.

*Constructing Phrasal Verbs*

Examining the genus term data received as input from Birmingham, one of the things that we noted was that, very frequently, for verbs or verbal phrases the first following discriminator was a single preposition. As our analysis of the Cobuild definition is currently concentrating on a consideration of the contents of the LHS plus the genus term and the "matches" on the RHS, and not yet treating the discriminator data, we risked losing this important information. It had to be attached to the superordinate in some way.

We thus decided to attach these following prepositions to the genus term, writing the new reconstructed value in the exp_superordinate or exp_synonym attribute. The original value remains in the prov_superordinate or prov_synonym attribute. In this way, the genus term value becomes equivalent in form to a phrasal verb. At times, we find that this choice is confirmed by the dictionary as, under the relevant entry, this combination of verb + preposition is listed in fact as a phrasal verb, while, at times, the particular preposition is indicated in the grammar field as obligatory for one sense of the verb.

We could have chosen only to write the preposition directly with the verb in those cases where we were forming an already recognized phrasal verb and, in the other cases, write the preposition in a preferred or obligatory preposition attribute. However, as is known, phrasal verbs are still a critical point for the dictionary compiler; the criteria which can be used to distinguish them from a straightforward adverbial or prepositional construction have not yet been generally agreed and it is possible to find different solutions in different dictionaries. We thus think that this reconstruction could well be useful for the lexicographer. We may however reconsider this decision once we have had the opportunity to study more data, deciding that it is more correct to tag the preposition as such and write it in following attribute.

In any case, currently processing the test vocabulary in this way, we have the following results. (In a number of cases, the values of the exp_entry and exp_superordinate/synonym attributes are also a result of other analyses, described elsewhere in this report.)

```
apply, 3, entry: applies,
    prov_superordinate: is relevant, exp_superordinate: be
                                                relevant to

carry, 2, entry: carry on
    prov_superordinate: take part, exp_superordinate: take
                                                    part in

charge, 6, entry: charge, exp_entry: have charge
    prov_superordinate: responsibility,
    exp_superordinate: have responsibilty for

consideration, 3, entry: consideration:
exp_entry: take into consideration
    prov_superordinate: think, exp_superordinate: think about

improve, 2, entry: improve
    prov_superordinate: get, exp_superordinate: get better at

information, 1, entry: information, exp_entry: have
                                                information
    prov_superordinate: know something, exp_superordinate:
                                                know about

permit, 2, entry: permits
    prov_superordinate: makes it possible
    exp_superordinate: make it possible for

resolve, 3, entry: resolve
    prov_superordinate: deal, exp_superordinate: deal with
```

In any case, it must be remembered that this strategy cannot evidence all potential phrasal verbs. In cases where the preposition does not immediately follow the genus term, or when it is written together with a noun phrase, then we do not attach it to the verb. For example, for **protect** 1 the input from Birmingham gives us: superordinate: prevent, match1: them, discriminator: from being harmed or damaged and we are unable to derive :exp_superodinate: prevent from.

In a subsequent stage, it will be important to also extract and code this information. However, the attachment of the preposition in some way to the verb is also important for another reason. It helps us in the disambiguation of the sense of the genus term. A strategy we intend to adopt to identify the correct sense of the genus term for verbs is described below in detail in the final section of this report. However, it is important to note here that, in two of the above examples, the value we have constructed for the genus term helps us to disambiguate it.

For example, with the entry for **consideration 3**, where the headword is **take into consideration**, our reconstructed genus term is "think about". **Think** has 8 different senses divisions, all for verbs, + 5 for phrases, and 3 for phrasal verbs. **Think about** is not listed as a phrasal verb but the grammar of sense number 2 is "VB WITH 'about', When you **think** about something, you consider it", and that of sense number 7 is "VB WITH 'of' OR 'about', If you **are thinking** of doing something, you are considering doing it." Thus we can reduce our search for the right sense of the genus term "think about" for **take into consideration** to **think 2** or **think 7**. Note that these two senses have the same superordinate "consider" and there thus is a near circularity leading us back to our starting point **consideration**. Two questions arise: how different are these two sense of **think**?; should **think about** actually appear as a phrasal verb in it own right? Thus our analyses provide important feedback for lexicographers.

Similarly, with **resolve 3** our value for the genus term is now "deal with". The dictionary entry for **deal** has 5 senses and also lists 3 phrasal verbs. One of these is **deal with** which has 2 sense divisions. The correct sense can be selected automatically following the method described in 3.5.2.1.

### 3.3.3 The syntactic-semantic parser

So far, we have described the type of information we derive from the Cobuild definitions. We will now give an idea of the procedures used to extract this information.

#### 3.3.3.1 Implementation

To implement the Pisa Syntactic-Semantic Parser, we have developed a subset of functions so that the program is as self-explanatory as possible. For the sake of clarity, this set of functions can be subdivided as follows:

i. list manipulation functions
ii. string manipulation functions
iii. grammar functions
iv. input/output structure manipulation functions

where lists are implemented as character strings operating on blocks of consecutive characters delimited by spaces ('words'), and i/o structures are binary tree-like structures whose nodes are accessible by path specification. All the programs have been written in the C programming language on a SPARC10 workstation.

Our parser takes as input the syntactically parsed or "chunked" definition data provided by Birmingham. The general strategy adopted is to try to identify significant chunks within the definition on the basis of clues provided by strings or sequences of strings which have been identified as typical delimiters of meaningful items of information. It thus uses a series of complex pattern matching techniques.

Here below we give just some examples of how these techniques have been developed to treat the co-text data. The programming details for this part of the procedure can be found in Appendix 1 to Deliverable 4, in the section "Inside Low-level Functions", see functions "elabmatch_info".

#### 3.3.3.2 Analysing the Co-text Data

The co-text data is processed as follows. The co-text must first be read to see if it contains "or" and ",". Each item or group of items which has been divided by "," and/or "or" is considered separately for the analysis. The co-text is then analysed to see whether values for the SPECIFIC, TYPICAL or *_FEATURES attributes can be derived:

- If the first item contained in the co-text is contained in a List (List2 of Appendix 2 above), e.g. you, someone, something, etc., and if the co-text contains one of the "triggers" given in another List (List4 of Appendix 2) e.g. such as, for example, then the string, or sequence of strings, following the "trigger" will be analyzed to extract values for the attribute TYPICAL and values for *_FEATURES will be derived from the item which is listed in List2.
- If the first item contained in the co-text is contained in List2 (e.g. you, someone, something, etc.), and if the co-text does not contain any of the "triggers" given in List4, then values for *_FEATURES will be derived from the item which is listed in List2.
- Otherwise, if the first item contained in the co-text is not contained in List2, then the co-text will be analysed to derive values for SPECIFIC.

In the following, we will give an idea of how the co-text is processed to extract information on "SPECIFIC" arguments. In this case, each item or group of items in the co-text are read. For each group of items, the last one is presumed to be the head argument and those preceding are normally taken as its modifiers. When the modifier is attached to an argument in the co-text that is not the first argument, it is taken as modifying only its adjacent head. Thus, if in our input data we have:

```
'(a)
'(word)
'(or)
```

```
'(a)
'(musical)
'(note)
```

we derive "word" and "note" as values for SPECIFIC and "musical" as the value for PREMOD of "note". Whereas any string preceding the first argument in the co-text is taken as referring to each argument. Thus, if we have

```
'(someone's)
'(advice)
'(or)
'(suggestion)
```

we derive "advice" and "suggestion" as values for SPECIFIC, each with an attached PREMOD: someone's.

When we find cases of embedded "of" groups preceding an argument, this group is read as the value of the modifier of the head. For example, from

```
'(a)
'(piece)
'(of)
'(writing)
'(or)
'(speech)
```

we derive as SPECIFIC values: "writing" and "speech", and "piece of" as the value for the PREMOD of "writing". These classes of premodifiers may be further analyzed to derive other semantic properties of the lexical item being defined in a second stage. Whereas, if the "of" group follows the noun, the group is read as a postmodifier of the noun. So with

```
'(an)
'(attitude)
'(,)
'(position)
'(,)
'(or)
'(way)
'(of)
'(behaving)
```

we derive as SPECIFIC values: "attitude", "position", "way" and "of behaving" as the value for the POSTMOD of "way".

When the embedded "of" group includes one of the special items in List2 (e.g. someone, something) then it is taken as modifying each preceding argument. Thus,

```
'(the)
'(rate)
'(or)
'(speed)
'(of)
'(something)
```

gives as SPECIFIC values: "rate", "speed", each of which has "of something" as its POSTMOD.

The co-text is more difficult to handle when it is an unbroken sequence of strings, i.e. it contains no ",", "or", or embedded "of" groups and thus a first splitting is not possible. In this case, the string is acquired to see whether it contains one of the items in List2 (e.g. you, something, etc., but also including which, who, and that) but not in the first positions. If so, the string is divided at this point. For example, if we have

```
'(the)
'(food)
```

```
'(you)
'(are)
'(eating)
'(,)
```

we derive "food" as value for SPECIFIC and "you are eating" becomes a value for POSTMOD. This phrase may then be subjected to further analysis in a second stage of development of the parser.

As stated above, when the first item of the co-text contains one of the special items in List2 (e.g. someone, something) values for *_FEATURES are derived from this item. In this case, any data following this item is taken as a post modifier of it. Thus, for example,

```
'(something)
'(that)
'(you)
'(have)
'(been)
'(offered)
```

gives "-anim" as value for *_FEATURES and "that you have been offered" as value of POSTMOD.

If the string following the item in List2 contains "or" or "," then it is divided into separate groups as described above and the features derived for the List2 item are assigned to each of the post modifying values. To give an example, with the definition "If you admit to something bad, unpleasant, or embarassing"

```
'(something)
'(bad)
'(,)
'(unpleasant)
'(,)
'(or)
'(,)
'(embarassing)
```

is rewritten as "something bad", "something unpleasant", "something embarassing", and our parser produces the following output:

```
subj_info          : subj_features1      : +anim, +hum
obj_info           : obj_postmod1       : bad
                   : obj_features1       : -anim
                   : obj_postmod2       : unpleasant
                   : obj_features2       : -anim
                   : obj_postmod3       : unpleasant
                   : obj_features3       : -anim
```

*Deriving *_FEATURE Values*

As has already been stated, when the co-text data contains one of the special items listed in List2 in the first positions, then the appropriate values for *_FEATURES are read from this list. However, when the cotext gives us values for SPECIFIC arguments then we attempt to find feature values for these arguments. For instance, feature values are found for SUBJ_INFO data using the following procedure:

i. considering the articles included in the contents of 'cotext1' and the inflection of 'head1'. In this way, it is usually (although not always) possible to derive the value for COUNT.
   a. if co-text1 contains "a", "an" or "one", then the value +COUNT is derived for each value of SPECIFIC.
   b. if co-text1 contains no article and the form of the verb in 'head1' is in the third person singular, then the value -COUNT is derived.

c. if co-text1 contains no article and the form of the verb in 'head1' is in the third person plural, then the value +COUNT is derived.
d. Otherwise, no value for COUNT is derived.

ii. examining the strings contained in the relevant 'match' field in the RHS. If the content of this field is any of the strings contained in List3, the relevant values are taken and inserted in *_FEATURES

For example, for **abdicate1**, we have the following piece of input from Birmingham:

```
(lhs-1
      (co-text1
        (match1
           '(a)
           '(king)
           '(or)
           '(queen)
        )
      )
      (head1
           '(abdicates)
           '(,)
      )
 )
 (rhs-2
      (match1
           '(he or she)
```

where the presence of the article "a" gives us the value of +COUNT and where the RHS match1 "he or she" matches the LHS match1 "a king or queen". (When the RHS match contains an "or", the strings before and after the "or" are associated with the relevant strings of the LHS match, e.g. in this case we have "king" associated with "he", and "queen" with "she".) The output of the parser is as follows:

```
subj_info      : subj1           : specific: king
               : subj_features1  : +anim,+hum, +masc, +count
               : subj2           : specific: queen
               : subj_features2  : +anim,+hum, +fem, +count
```

## 3.4 Representation: from the Intermediate Template to Typed Feature Structures

This section illustrates the second and the third step (evaluation and representation) in the process of conversion of actual Cobuild lexical entries into formalized entries in the form of Typed Feature Structures. The general background of this conversion procedure is provided in section 3.4.1, whereas the two steps are illustrated in section 3.4.2 and 3.4.3 respectively. In section 3.4.2, the Intermediate Template is re-analysed from the conversion point of view, i.e. the different types of information extracted are evaluated with respect to their representability in TFS form and utility for NLP purposes. Section 3.4.3 illustrates in detail the Pisa TFS representation. Finally, section 3.4.4 raises the issue of the role of the extracted information in NLP, as "preferences" or as "constraints".

### 3.4.1 Conceptual background

Before discussing the evaluation and representation of the information contained in the Intermediate Template in terms of Typed Feature Structures, the conceptual coordinates of our work need to be provided. As already mentioned in the Introduction, the project was born as an answer to "the need to incorporate into formal grammars a representation of the actual usage of words" as testified by Cobuild dictionary entries (see Project Technical Annex, p.4).

The Intermediate Template, described in the previous section, represents the first step of this formalization process. However, the IT is not directly usable by NLP systems. First, it contains

intermediate results to which a definite semantic interpretation still has to be assigned. Second, and more important here, it represents the formalization of the Cobuild entry as such, in the sense that it mainly reflects the Cobuild lexicographic descriptive framework. In fact, the IT contains all the information clustered around each word sense defined by Cobuild, and there is no commitment to any other linguistic formalism. In this sense, the IT can be defined as theory-neutral, since the information contained in it could, in principle, be converted into different lexical representation formalisms for use within different theoretical frameworks.

However, in order to make the extracted information directly exploitable by NLP systems, a further step in this formalization process was required: the extracted information needed to be related to a formal theory of grammar and thus be represented in an appropriate lexical representation language. Hence, the conversion of IT entries into TFSs arises from the need to transform a formalized lexicographic description into a representation to be integrated within a formal grammar description.

The three stage approach adopted by Pisa for the formalization of Cobuild entries described in section 3.2 implies a change in perspective between the different steps of formalization. In the first step, i.e. that which results in the IT, the focus of the attention was on the identification, extraction and formalization of the information implicitly or explicitly contained in Cobuild entries. In the second step, that covering the evaluation and conversion stages and resulting in the TFS representation, the attention has been shifted to the integration of such information within a formal grammar framework.

Thus a change in perspective, however, does not mean that the IT is mainly data-driven, and the TFS representation is mainly theory-driven. In spite of the fact that the TFS representation has been conceived for integration in a theoretical framework, its adherence to the actual content of Cobuild lexical entries (in their turn reflecting the actual usage of words) remains a fundamental requirement to be met. Therefore, in the design and implementation of the TFS representation it has been necessary to find a balance between adherence to the actual data on the one hand, and their representability and exploitability in the framework of current formal grammars and/or NLP systems on the other hand. Clearly, the choice of representing information on the actual usage of words—such as typical usages, style, and preferential information on the arguments or, more generally, collocates of lexical items—obliged us to revise and integrate the adopted framework in order to include information not currently handled by formal grammars, even though it has not always been possible to finalise a reasonable formalization. Some of the types of information that we would have liked to formalize are not handled by current TFS formalisms, at least to our knowledge. In the following sections, we will illustrate how the selected formal framework has been revised in order to integrate the information relative to the actual usage of words.

Head Driven Phrase Structure Grammar—HPSG—(Pollard and Sag, 1987, 1993) has been chosen within the Project as the Common Interface (see Deliverable 3). Thus, our TFS entry mainly rests on the theoretical notions of HPSG, which we have enriched and adapted to include and represent the information we extracted from the Cobuild dictionary. The choice of HPSG as a formal and grammatical framework has implications at different levels.

Let us first consider how this choice relates to the Pisa's specific task within the project, i.e. the conversion of Cobuild dictionary entries into TFSs: it should be noted that, from the formalism point of view, HPSG and TFS are fully compatible as typed feature structures are the formal tools used to build the HPSG theory of natural language and semantics.

From a more general perspective, the choice of the HPSG formalism has a further advantage: in contrast to formalisms such as DATR, it is not restricted to lexical representation. This makes it much easier to integrate our TFS entries in NLP systems based on HPSG, for testing and use. Although this means that the lexical representation language is in a sense too general for our application, and thus requires revisions and integrations, on the other hand it ensures the testing and usability of the produced lexical entries by HPSG-like grammars (e.g. by CEC grammars, which are based on HPSG).

Moreover, as a grammar theory, HPSG shares with Cobuild the view that syntactic and semantic patterns are strictly interlocked, both contributing to the definition of linguistic objects. HPSG has

been designed as an integrated theory of natural language syntax and semantics, i.e. it integrates different linguistic dimensions in its description of linguistic objects: in fact, the HPSG description of linguistic signs includes specifications of their phonological, syntactic and semantic properties. Even if the perspective is different, a parallel assumption lies behind the Cobuild dictionary, where syntactic, lexical, semantic, and pragmatic properties are considered as interacting and contributing to the identification of each word sense: in fact, the distinction of different word senses in the Cobuild dictionary is determined by the combination of syntactic and semantic factors.

Due to this convergence of views between HPSG and Cobuild, it was possible to formalize one of the main assumptions behind the Cobuild dictionary, i.e. the interlocked dependency of syntactic, lexical and semantic properties in the definition of lexical items. This possibility led us to adopt an integrated representation of syntactic and semantic information within our TFS entries, rather than restricting ourselves to the representation of semantic information only, as had been the initial commitment within the project. However, together with these theoretical motivations, another more practical reason supported our decision to integrate all the information extracted from Cobuild lexical entries (i.e. morphological, syntactic, semantic and pragmatic data) within our TFS entries: we felt that this choice could help to guarantee the usability of our results by real world applications with only minor adaptations, and in particular without the need to add a lot of missing (i.e. mainly syntactic) information.

### 3.4.2 Evaluating and classifying the information contained in the Intermediate Template

A preliminary step in the design and implementation of the TFS representation of Cobuild lexical entries is represented by a careful evaluation of the information extracted from and stored in the Intermediate Template. This corresponds to the second step of our conversion procedure, whose final result is the selection of the information to be converted into TFS form. The Intermediate Template has thus been revisited from the conversion point of view. The information it contains has first been classified on the basis of whether:

i. it was immediately representable in TFS form;
ii. it needed further analysis.

The first group contains all the information extracted from Cobuild entries which could—in principle—be represented in TFS terms. By contrast, the second case refers to our so-called "intermediate results", that is to those cases for that we have been unable to assign a final and reliable semantic interpretation; in these cases, further analysis is needed before the information can be considered for the conversion into TFS form. This is the case, for instance, of all those attributes in the Intermediate Template whose name ends in *_PREMOD or *_POSTMOD (e.g. SUBJ_PREMOD and SUBJ_POSTMOD, OBJ_PREMOD and OBJ_POSTMOD, OBLIQUE_PREMOD and OBLIQUE_POSTMOD, etc.). For each argument of a verb, these attributes contain semantically uninterpreted modifiers appearing in the definition text. The distinction between PREMOD and POSTMOD is syntactically based since it reflects whether their values pre- or post-modify the argument under consideration. They will be ready for conversion only after further analysis directed at formulating any significant linguistic generalizations holding for classes of arguments across different verbs.

However, not all the information in i. has been selected for conversion. A selection within the i. group has been operated on the basis of whether the representable information could be:

a. used for access and retrieval within the TFS lexicon built starting from the Cobuild dictionary,
b. usefully exploited by NLP applications, and in particular by HPSG-like grammars.

Only information meeting the requirements in a. or in b. has been selected for conversion. On the basis of these two criteria, DEF_No. (the number univocally indentifying each definition) and DEF_TYPE (the code identifying the definition strategy used by the lexicographer in the definition) attributes have both been excluded from the conversion procedure. Despite their representability

in TFS terms, they represent "working" attributes that have been used to guide the extraction procedure (for instance, the definition type determined the extraction strategy which has been applied for the analysis of each definition), and for which we do not see any possible use for the relevant applications—NLP and Lexical Data Bases—considered in this context.

The remaining attributes selected for conversion can be classified on the basis of their role with respect to a. and b. above. Thus, LEMMA and SENSE_No., corresponding to the dictionary headword and to the number identifying its sense respectively, represent the dictionary coordinates of the entry being defined whose role within the TFS entry is mainly that of access keys.

All other remaining attributes could be, one way or another, of some use in NLP applications. Within this group, a last distinction can be drawn between constraining and preferential information, the former representing conditions for defining linguistic structures (i.e. constraints), and the latter measures for comparing already defined linguistic structures (i.e. preferences). This distinction, which will be discussed in more detail in a following section, is also crucial for our conversion purposes. In fact, there are circumstances under which, depending on the role assumed by the converted information in NLP applications (i.e. as a preference or as a constraint), a different representation strategy must be adopted. In what follows, we will briefly run through the remaining attributes to establish their status and role within the conversion procedure.

The ENTRY attribute, containing the actual lexical item currently being defined, is converted as the "headword" of the TFS entry. We have decided to take the value of ENTRY rather than that of LEMMA as the headword of the TFS entry given that most of the attributes in the template refer to the value of ENTRY and not to that of LEMMA. For instance, under the lemma **attention** the Cobuild dictionary has an entry whose headword is **pay attention**; as shown in section 3.3.1.4 above, most of the information recorded within the IT entry for **pay attention** refers to this expression as a whole (see, for instance, the preferential information specified about its arguments), rather than to the lemma under which it is recorded. This is the reason why LEMMA has been 'downgraded' to the role of a mere access key. However, upgrading the ENTRY attribute to the role of headword poses new problems in lexicon construction. In spite of the fact that we usually think of a lexicon as a list of individual words, the ENTRY attribute often contains complex lexical items or phrases to which a single meaning has been assigned: from this perspective, a revision of the distinction between phrasal and lexical signs as conceived in HPSG is needed (we will discuss this point in more detail in a following section).

The GRAM attribute of the Intermediate Template plays a crucial role in the conversion procedure: its value is used to select the appropriate typed feature structure used to represent the entry being converted. For instance, the grammar code VB WITH OBJ tells us that we are dealing with a transitive verb subcategorizing for two noun phrases, the one playing the function of subject and the other that of object. This implies that an IT entry with this grammar code is translated into the TFS for representing verbs, where the attributes for recording the subject and object information have been defined. Similarly, the grammar code UNCOUNT N translates into the TFS corresponding to nouns, where the value of the COUNT attribute has been set equal to *uncount* and restrictions have been defined with respect to the selection of the determiner. Details about the correspondence between grammar codes and TFSs are given in the following section. Here it is sufficient to point out the key role of the grammar code in the conversion process.

However, GRAM is not always helpful in selecting the relevant typed feature structure to be used to represent a given IT entry. When within the IT the value of GRAM is *phrase*, this information must be further integrated with the value of the attribute PHRASE_TYPE (which has been computed during the previous parsing stage). In fact, the grammar code *phrase* is too general, as it simply refers to a group of words that are used together, regardless of the kind of phrase (e.g. adverbial, verbal etc.); put in these terms, this specification is not sufficient to determine the selection of the appropriate TFS to represent the PHRASE entry, and thus needs to be integrated by the PHRASE_TYPE specification. This is the case, again, of **pay attention**, whose grammar code is PHRASE and for which it has been inferred that the PHRASE_TYPE is *verbal*.

There are also cases in which the value for GRAM derived from the Birmingham input data and recorded within the IT may be overridden. This happens, for example, in the case of those

nouns or adjectives which have actually been defined as part of a verbal expression (examples of this type of entry have been given in section 3.3 above). In this case, the value *verbal* has been assigned to PHRASE_TYPE and the corresponding TFS entry will have the structure of a verbal entry, rather than that of a noun or an adjective as could have been expected on the basis of the grammar code. This is the case, for instance, of the entry **have flexibility**, recorded under the lemma **flexibility**, where the GRAM attribute has the value of UNCOUNT N and PHRASE_TYPE is equal to *verbal*. Thus, the PHRASE_TYPE attribute is always used in combination with the grammar code information, either to integrate it (as in the case of **pay attention**) or to override it (as in the case of **have flexibility**).

The information contained in the SEC_GRAM attribute is also converted. As stated before, SEC_GRAM provides other grammar codes, to be combined with the main grammar code: all together, they hold concurrently with respect to the word sense being defined. Different strategies have been adopted for converting the SEC_GRAM information. Whenever possibile, the values of GRAM and SEC_GRAM are collapsed within the same TFS entry. For instance, in the IT entry for **abstention 1** the values of GRAM and SEC_GRAM are set to *count n* and *uncount n* respectively: in the corresponding TFS entry, the disjunction *count V uncount* has been assigned as value of the COUNT attribute. But this solution is not always possible: in the case of the verb **accept 1**, the grammar codes assigned as values of GRAM and SEC_GRAM are *VB with OBJ* and *VB* respectively: these codes correspond to two different typed feature structures, those for transitive and intransitive verbs. The strategy adopted in cases like this has been that of generating, during the conversion stage, the TFS entry corresponding to the main grammar code—the transitive reading in the case at hand—and then recording within the TFS the other possible reading, which will be generated by means of lexical rules applied to the transitive entry.

The information contained in the INFLECTION attribute has been converted for use by lexical rules. In fact, the TFS entry converted from the IT representation always refers to the "base form". However, this base lexical entry contains all possible inflected forms—both the regular and the irregular ones—of the lemma being defined. Contrary to HPSG, whose base lexical entries contain only information about irregular morphology (the regular inflected forms being predictable on the basis of morphological rules), we decided to provide this kind of information for all entries, given that this kind of information was regularly provided by the Cobuild dictionary, and—on the other hand—given the simple morphology of the English language. Although, at first glance, this choice makes the TFS entry heavier, on the other hand it avoids the need for a morphological component. This has been possible because during the previous extraction stage, the set of relevant inflected forms for each separate word-sense was isolated from the complete set attached to the headword; in fact, in the Cobuild dictionary the list of inflected forms of each headword included all possible forms, regardless of the part of speech they referred to.

So far, we have illustrated attributes playing the role of "constraints" in the construction of linguistic structures. The remaining attributes, illustrated below, mainly refer to "preferential information" to be used to select, from different constructions or interpretations, the preferable one. At this stage, it is important to know the role of this kind of information in the parsing or generation process only in the case of those attributes which, in a sense, duplicate information already present with the role of constraint. This is the case of the attributes VOICE and FORM, in the intermediate templates of verbs and nouns respectively.

The VOICE attribute (whose possible values are 'active', 'passive', 'progressive') is used for verbs and specifies the preferred voice or tense used in this sense of the verb; this kind of information has been inferred from the LHS of the definition. Here, before proceeding to the conversion in TFS form, we have had first to clarify the relationship between the grammar codes and the parallel grammatical information extracted from the definition. Of course, the value *passive vb* for GRAM is different from the value *passive* for VOICE: in the former case, the verb is only used in the passive voice, whereas in the latter the verb is usually but not always used in the passive (i.e. other possible usages cannot be excluded). Thus, the representation of these two cases should be differentiated. Following the HPSG specifications, the first case can be represented by restricting the set of values of VFORM to *pas* and by revising the subcategorization lists accordingly, i.e. by directly generating

the TFS entry for a passive verb. But this solution is not suitable for the second case, given that other usages could not be excluded. We have thus inserted a new attribute, PREFVFORM, to represent the preferential usage, without overriding the information given in the main grammar code.

Similar observations hold for the FORM attribute in the IT for nouns, specifying the typical form under which a given noun occurs. When the attribute FORM is assigned the value *plural*, this means that the countable noun under consideration is commonly used in the plural form, but this does not exclude its occurrence in the singular. This specification has a parallel in the grammar code PLURAL N, qualifying nouns used only in the plural, e.g. **clothes**. Whereas the second case can be represented in TFS form by directly generating the entry for a plural noun, with the value of the NUMB attribute restricted to *plur*, the first case needs to be dealt with by means of an ad hoc attribute, called PREFNUMB, specifying the preferantial usage for that noun. Hence, the information recorded as value of the PREFVFORM and PREFNUMB attributes will rather be used by a hypothetical module handling preferential information, especially for generation tasks.

The remaining attributes all refer to preferential information. However, given that their values do not interfere in any way with other attributes expressing constraints, their TFS representation at this stage does not pose particular problems: the problems with this kind of information are restricted to its exploitation by NLP applications, as discussed in section 3.5 below.

The INFERENCE attribute has been converted in TFS form, in spite of the fact that we do not know whether it could be usefully exploited within NLP systems. It refers to the kind of action expressed by the verb, e.g. possible, likely, inherent, negative/unlikely, collective, subjective: this seems to be rather a human oriented specification. In fact, neither HPSG nor any other grammar theory we know of seems to handle this kind of information.

The GENUS attribute has currently been only partially converted; for a complete, exhaustive and coherent representation in terms of TFSs, it would have been necessary to have extracted from the dictionary the whole set of semantic relations emerging from it, together with all possible values. A possible alternative would have been to operate on a semantically homogeneous dictionary subset; this is what has been done within the Acquilex project. But, as stated above, this was not the main focus of attention within the project: the criteria adopted here for the selection of the sample of entries to be formalized have been the representativity of the different parts of speech, and of the different defining strategies. From this, it follows that the results extracted as values of the genus information would need further integrations and analysis before an exhaustive formalization would be possible. For the time being, our TFS representation of the GENUS information should be seen as a general proposal whose details need to be refined.

The attributes SUBJ_INFO, OBJ_INFO, OBJ2_INFO and OBLIQ_INFO for verbs, PRECOLLOC_INFO and POSTCOLLOC_INFO for nouns, and COLLOC_INFO for adjectives have all been converted into TFS form. From the conversion point of view, their role has been two-fold. First, in a number of cases they have been used to integrate the syntactic information provided by the grammar code, and thus directly contributed to the general shaping of the TFS. This is the case, for instance, of optional complements appearing in the typical context of the word being defined, i.e. the LHS of the definition, but not specified by the grammar code (probably due to their optionality). Consider, for example, the entries for the verbs **acquiesce 1**, **act 4** or **adjudicate 1**, for which the grammar code did not include the information about the prepositional complement (introduced by the prepositions "to", "in" and "on" respectively) emerging from the LHS of the definition. However, the major contribution of these attributes to the TFS representation is from the semantic point of view, i.e. they provided useful information on the semantic preferences imposed by the word being defined on its arguments or, more generally, collocates. All these semantic preferences have been encoded in the TFS to be used as preferential information.

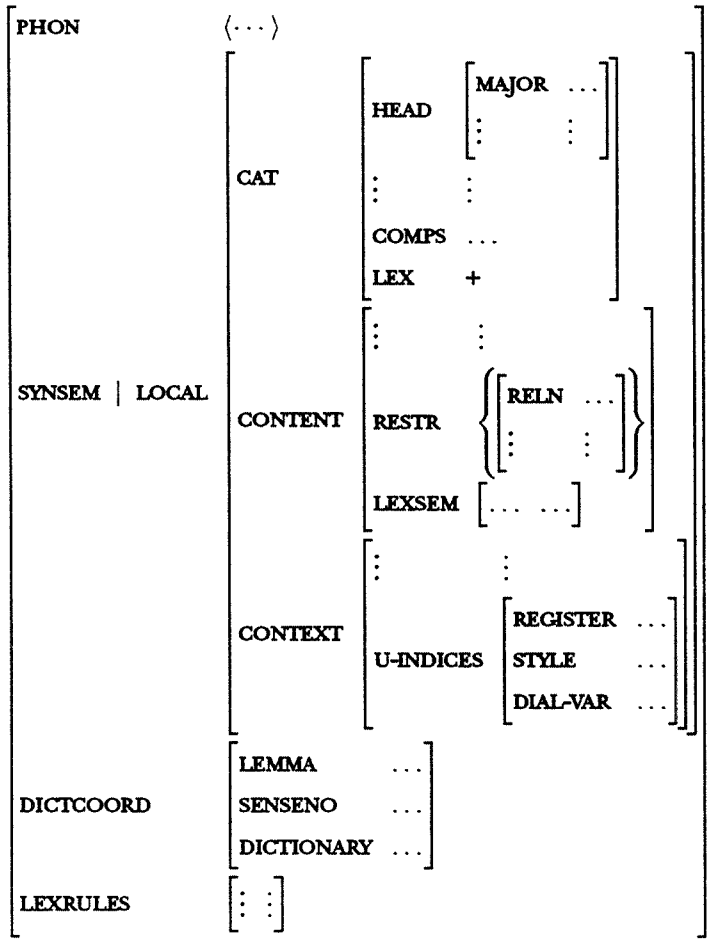
### 3.4.3 Conversion and Representation in TFSs

In the previous section, we selected the relevant information contained in the Intermediate Template to be converted into TFS form. What must be shown now is how the relevant information has been mapped onto the TFS representation format.

As stated above, HPSG has been chosen as the theoretical framework to which our entries

should conform. Thus, the illustration of our TFS entries will be limited to pointing out how the Pisa TFS representations differ with respect to canonical HPSG. Two different kinds of divergences can be observed: first, HPSG features to which a different interpretation has been assigned; second, Cobuild-specific features which have been added.

Let us start from the skeleton common to all our lexical entries, which is represented by the following TFS:




As can be noted, this structure complies, to a large extent, with the general HPSG framework: it corresponds to the TFS associated with all linguistic *signs*, where orthographic (PHON), syntactic and semantic (SYNSEM) information is simultaneously represented. The main differences lie in the insertion of Cobuild-specific features such as DICTCOORD, LEXRULES, LEXSEM and U-INDICES, and in a different interpretation of the LEX and CONTEXT features.

Let us first consider the new features. The DICTCOORD feature encodes the coordinates locating a given entry within a dictionary: i.e. it contains information about the dictionary (DICTIONARY), the lemma (LEMMA) under which the entry being defined has been recorded, and its sense number (SENSENO). Within the LEXRULES attribute, we have recorded features whose values have to be used by lexical rules to produce morphologically related versions of the same word: for instance, the TFS entry for the plural form in the case of countable nouns, or the TFS entries for the third person singular or the past forms in the case of verbs. Hence, as we will see below, the value of the LEXRULES feature varies depending on the part of speech of the entry being defined. The LEXSEM feature, inserted among the attributes defining CONTENT, encodes the information extracted from the RHS of the definition, i.e. the genus information. Following the typology of relations illustrated above, different kinds of relations can be found as value of this

attribute. Again, in this case too, the possible value of this attribute mainly depends on the part of speech of the word being defined. Finally, the feature U-INDICES (i.e. usage indices), embedded as value of the CONTEXT attribute, is employed to characterize the word being defined with respect to its contexts of use, specified through the REGISTER, the STYLE and the English variant DIAL-VAR attributes.

Let us turn now to attributes for which our interpretation differs with respect to canonical HPSG, LEX and CONTEXT.

As can be noted, the path SYNSEM | LOCAL | CAT | LEX has been assigned the value +. In HPSG, this restriction on the possible values (i.e. + or -) of the attribute LEX is part of the definition of *lexical signs* which, together with *phrasal signs*, are subtypes of the more general type *sign*. The distinction between *lexical signs* and *phrasal signs* is based on the value assigned to the path SYNSEM | LOCAL | CAT | LEX, which is + and - respectively, and on the fact that the latter introduces a new attribute, DAUGHTERS which is typed to *constituent-structure*. From this, it follows that *lexical signs* do not have an internal structure. It is in this respect that our interpretation of LEX differs from HPSG: in fact, as stated in the previous section, our TFS lexicon includes entries for individual words as well as entries for phrases recorded within the Cobuild dictionary. For all of them, the value of LEX has been set to +. Thus, according to our interpretation, the feature LEX simply indicates the fact that the TFS entry is part of the lexicon, regardless of its internal structure.

According to HPSG, the CONTEXT value contains context-dependent linguistic information: following this general definition, we assigned as value of this feature the Cobuild-specific attribute called USAGE-INDICES (U-INDICES), which represents information about various indexical coordinates locating the word being defined with respect to the register, the style and the dialectal variety it relates to.

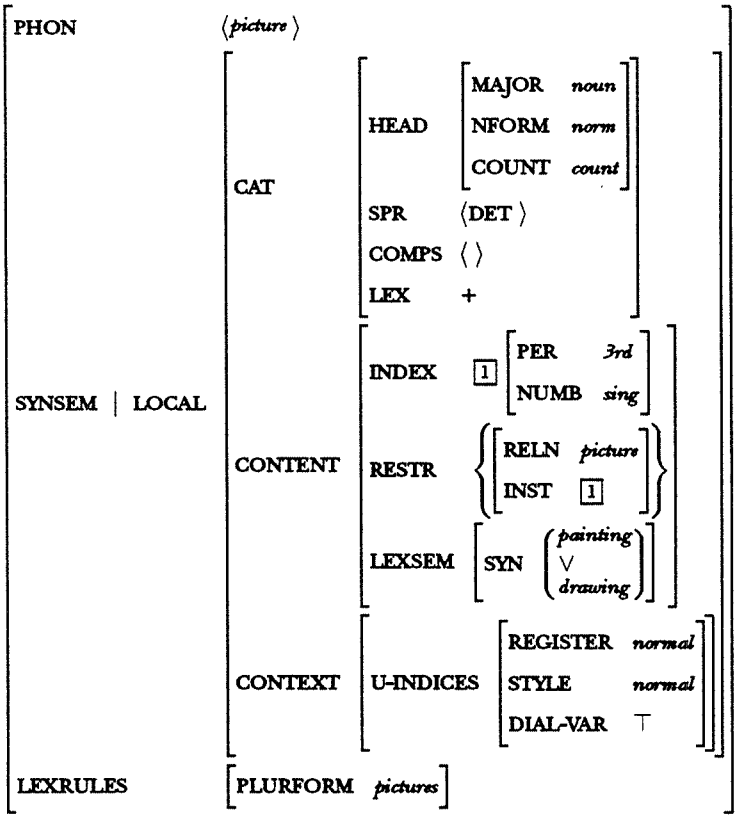
The figure above shows the skeleton to which all our TFS entries conform. In what follows, we will illustrate through a series of examples how this skeleton has been specialized to represent actual Cobuild entries. In this conversion process, the role played by the grammar code information is crucial. It is used to select the general type to which the individual entry refers: e.g. nouns, verbs, adjectives. Thus, the total information content of each individual entry is the result of unifying the idiosyncratic information particular to the entry being defined with the information inherited from the general types to which it relates. A selection of lexical entries for each part of speech is given below. Given that our main purpose here was that of illustrating our TFS representation, the TFS entries have been simplified to make them more readable; for instance, the attribute DICTCOORD has been systematically omitted. There are other omissions within the single sections: the first TFS entry of each section contains all the relevant information for the part of speech under analysis; the following ones mainly show the attributes under discussion.

#### 3.4.3.1 Nouns

According to the Cobuild grammar codes, nouns are divided into the following classes: COUNT (able) N(ouns), UNCOUNT(able) N(ouns), COLL(ective) N(ouns), MASS N(ouns), PLURAL N(ouns), SING(ular) N(ouns), PROPER N(ouns). A selection of entries illustrates how the Cobuild grammatical specifications have been converted into TFS/HPSG terms. It should be noted that the design of these representations has been made on the basis of the grammar notes in the Introduction of the Cobuild Student's Dictionary.

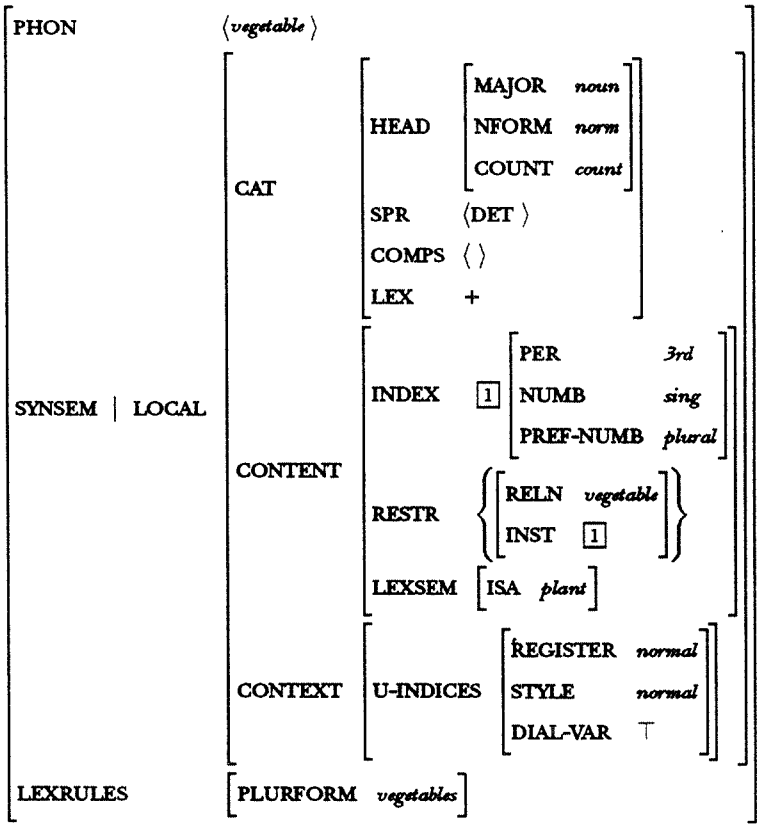
**COUNT N**

Following Cobuild, countable nouns have both a singular and a plural form, the determiner is obligatory when used in the singular, whereas it is optional when used in the plural. Let us consider how this information has been encoded in TFS form, as exemplified by the entry for **picture** 1 given below:



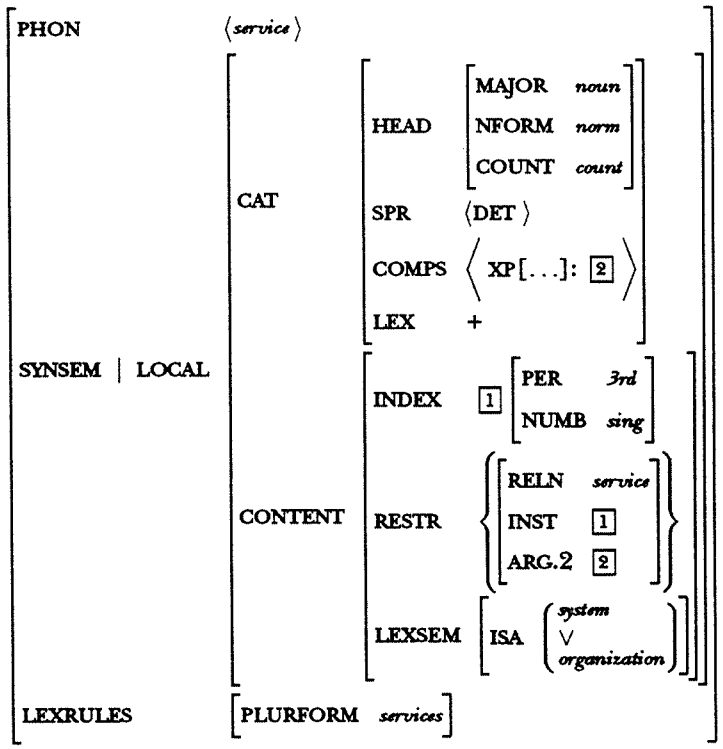

The TFS entry corresponds to the base form, which in this case is the singular form. Thus the specifications contained in it relate to the singular: the presence of a determiner is obligatory (see the value of the SPR attribute), the value of NUMB is set to *sing*. Moreover, the countability information is recorded as value of the path SYNSEM | LOCAL | CAT | HEAD | COUNT, which is set to *count*; it should be noted that the COUNT attribute has been added for our purposes. The plural form of the entry being defined is recorded as value of the attribute PLURFORM in LEXRULES. The lexical rule operating on this sign to produce the entry corresponding to the plural form will change the value of NUMB into *plur*, make the determiner optional, change the value of PHON to the plural form. As far as the genus information is concerned, in this case two synonyms (see the SYN attribute) have been derived.

Let us now consider another countable noun, **vegetable** 1, for which the Cobuild dictionary, in the LHS, has provided information about a preferential usage: **vegetable** is typically used in the plural form.




The information about the preferential usage has been encoded as value of another Cobuild-specific attribute, PREF-NUMB, which has been inserted among the attributes defining *index*, and whose value is equal to *plur*. In this case, a hyperonym (see the ISA attribute) has been provided as value of LEXSEM.

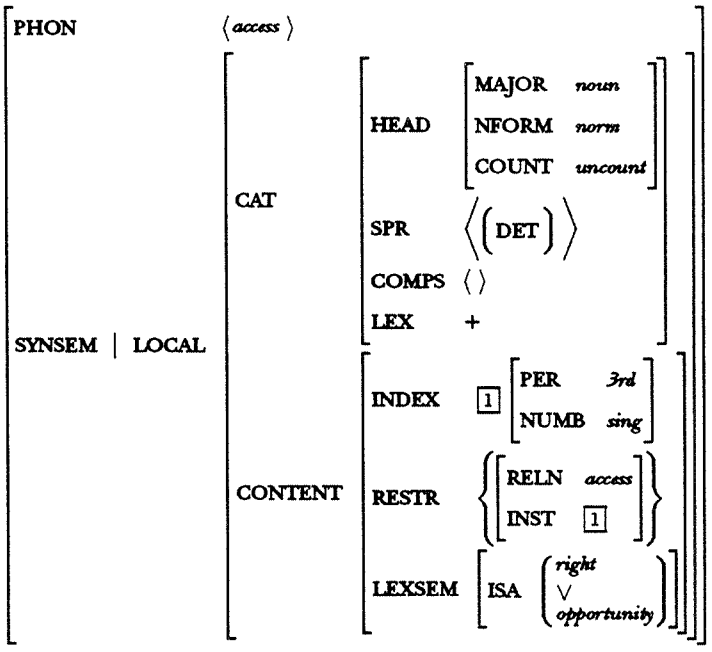
However, the above grammar codes are sometimes combined with other specifications. Let us consider the case of the complex code COUNT N WITH SUPP, which means that the corresponding countable noun is not usually used on its own and needs an adjective in front of it or a relative clause or a prepositional phrase after it. **service 1** exemplifies this case:




The above TFS represents this case, where the complements (COMPS) list contains an XP complement, unspecified with respect to its syntactic category, which is coindexed with the ARG.2 of the relation in the RESTR attribute. It should be noted, however, that, before they can be used, entries like this need major revisions: first, the syntactic category of the supporting word needs to be specified; second, and perhaps more importantly, a distinction should be made between the case in which the supporting word is a complement subcategorized for by the word under definition and the case in which it is simply an adjunct. Neither kind of specification is provided by the Cobuild dictionary, which has been conceived for the human user.

**UNCOUNT N**

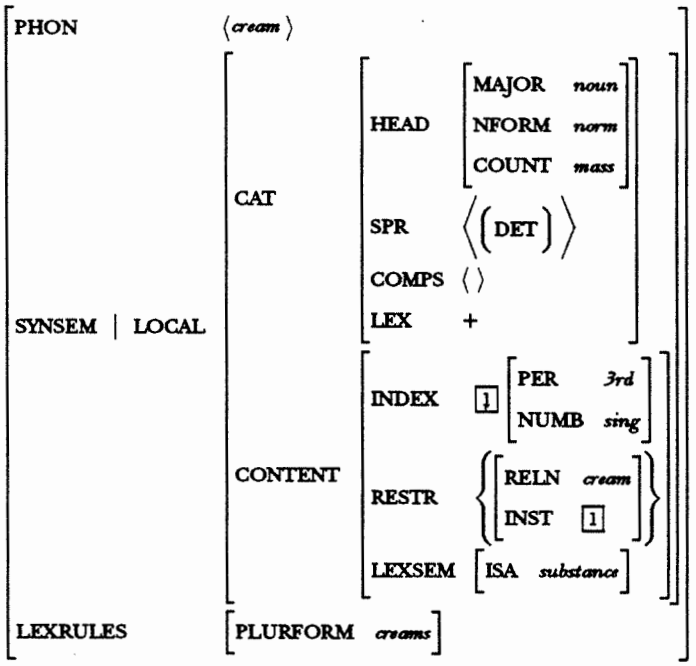
According to Cobuild grammatical information, uncountable nouns have only one form—the singular one—and are not used with a determiner in front of them, unless extra or specific information is given (for instance, when an adjective is used with them). Consider the entry for **access 2** below:




As can be noticed, the UNCOUNT N grammar code has been translated in TFS form by: i) setting the value for COUNT to *uncount*; ii) making the determiner optional (expressed by enclosing the value of the SPR attribute between round brackets); iii) omitting the specification of the plural form (this means that no lexical rule will operate on this sign).

**MASS N**

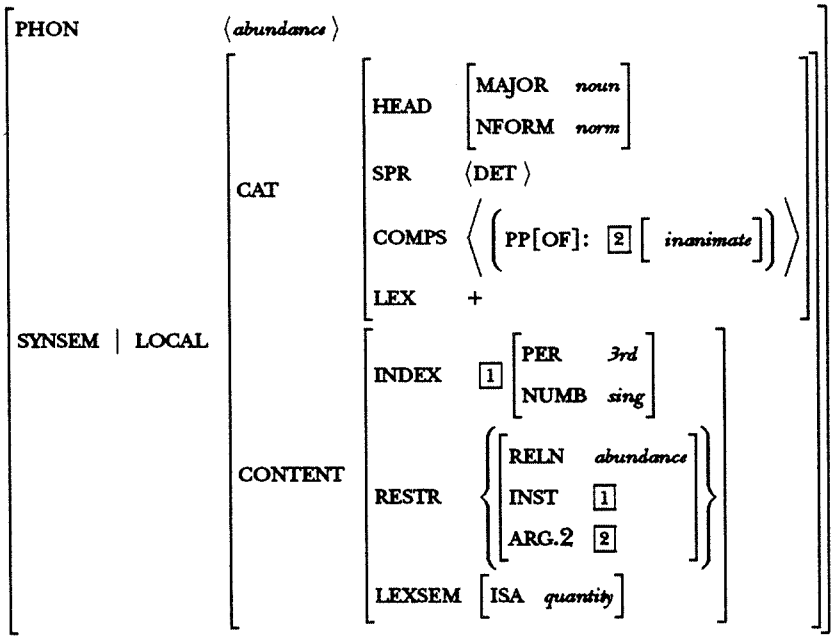
Mass nouns are usually used as uncountable nouns; however, when they refer to quantities or types of something, they can also be used as countable. On the basis of this definition, the TFS corresponding to mass nouns results from the integration of features of both countable and uncountable nouns. The entry for **cream 3** represents the case in point:




Here, the value for COUNT has been set to *mass*. Like uncountable nouns, the determiner has been encoded as optional; like countable nouns, it has a plural form (the lexical rule operating on this sign will be similar to that operating on countable nouns, except for the fact that the determiner is already optional).

**SING N**

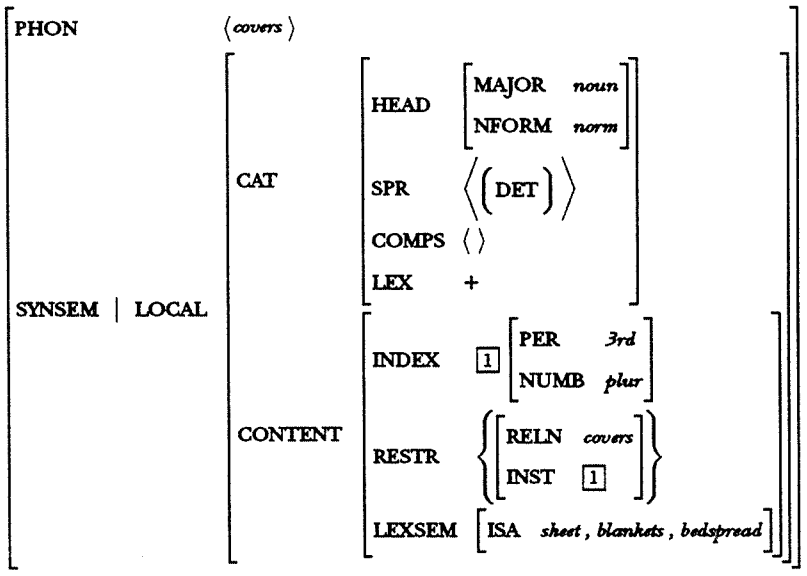
As explicitly stated in the code name, singular nouns are used only in the singular; moreover, they must have a determiner in front of them. Consider the entry for **abundance 1**:




where the determiner is obligatory, the NUMB is *sing*, and no LEXRULES attribute has been specified. The TFS above shows another kind of information which emerged from the definition text, but is not stated within the grammar code: **abundance 1** subcategorizes for a prepositional phrase headed by the preposition "of", and whose noun complement should be inanimate. This specification has been encoded as value of the COMPS list (for the encoding of semantic preferences on arguments see the section on verbs). The fact that this kind of information has not been given within the grammar code has been interpreted as a reflection of the fact that it corresponds to a typical usage, rather than an obligatory one. The representation of this kind of information followed from this interpretation; thus, the presence of this complement has been encoded as optional (see the round brackets enclosing it). Except for the fact that here the syntactic category of the complement is known, all the other observations made with respect to the WITH SUPP specification hold in this case as well.

**PLURAL N**

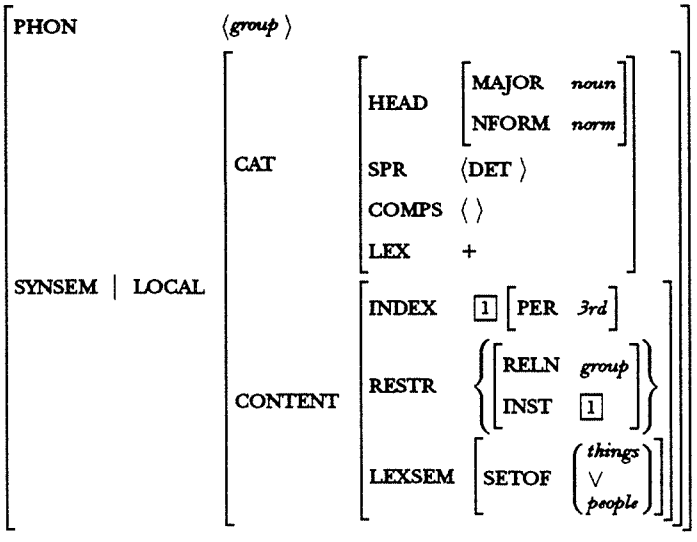
Plural nouns are only used in the plural and can have a determiner in front of them. This can be observed in the TFS for **cover 11**:




where everything refers to the plural form (see the values of PHON and RELN attributes), the NUMB value is restricted to *plur*, and the determiner is encoded as optional.

**COLL N**

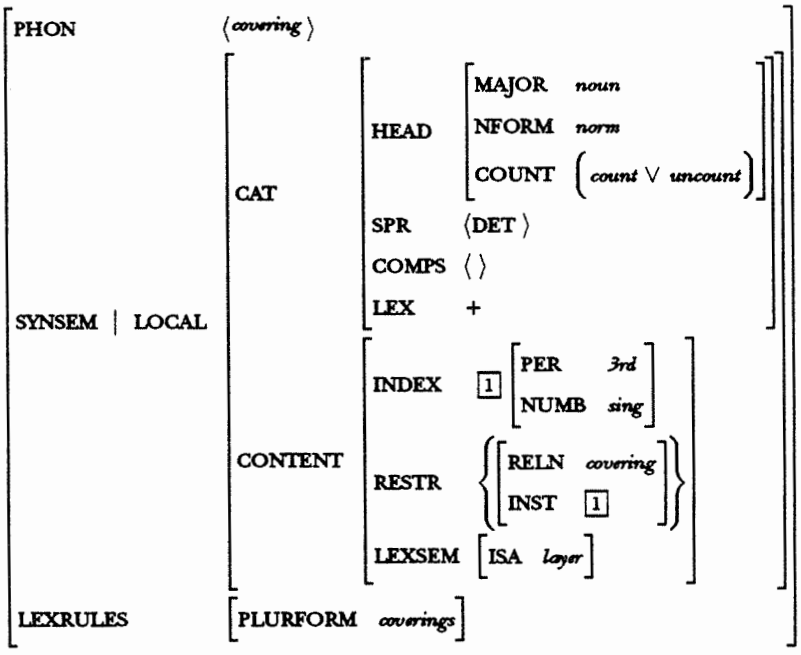
Collective nouns refer to groups of people or things. In the singular, they take a determiner; either the singular or the plural forms of verbs and pronouns can be used with them. Consider the TFS entry for **group 1**:




As can be noticed, the NUMB attribute is unspecified and for this reason it does not appear within the TFS (in fact, unspecified attributes have often been omitted for sake of readibility): this means that it can assume both *sing* and *plur* values, depending on the context. Concerning the LEXSEM attribute, another kind of semantic relation can be observed as its value, i.e. SETOF.

**COMPLEX GRAMMAR CODES**

In the previous section, we have seen that the main grammar code is sometimes combined with other grammar codes, and that the encoding of this combination varies from case to case. The general strategy we followed for encoding combination of grammar codes has been that of collapsing them, whenever possible, within the same structure. **covering 1** exemplifies one of the possible combinations: it is defined as COUNT N or UNCOUNT N. The TFS below shows how this combination of codes has been formalized:

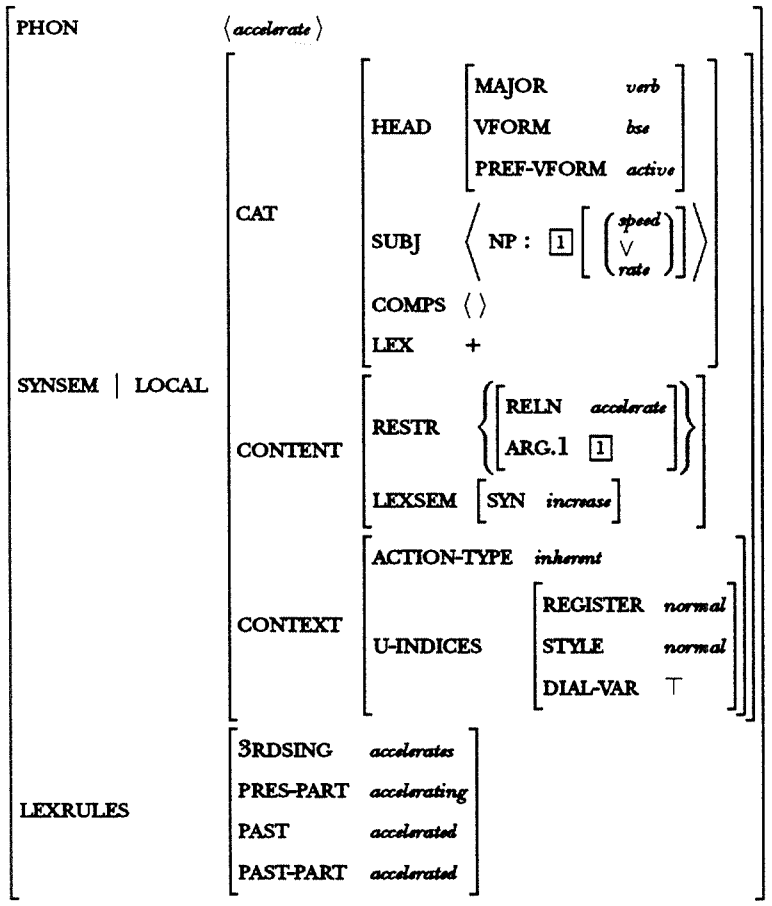



The disjunction *count V uncount*, assigned as value of the COUNT attribute, shows that this TFS results from a combination of codes: the order of the elements of the disjunction is relevant, since the first one encodes the main grammar code, and the second one the secondary one (the first grammar code is expected to reflect the most typical use of the word being defined). On the basis of this fact, in cases like this, the lexical entry corresponding to the main grammar code has been produced. Thus, the TFS above, except for the value of the COUNT attribute, is that of countable nouns. In the case of entries such as **structure 1**, where the main grammar code is UNCOUNT N and the secondary one is COUNT N, the disjunction given as value of the COUNT attribute will be different, i.e. *uncount V count*. Accordingly, the TFS produced will by that of the uncountable reading. In both cases, the other reading should be generated by lexical rules whose application is triggered by the value of COUNT.

### 3.4.3.2 Verbs

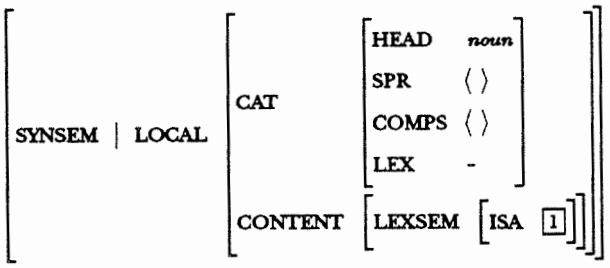
Let us now go through the most important types of lexical entries for verbs. In this case, the criteria which guided the selection of TFS entries were different from those for the nouns: rather than the representativity of the different grammar codes, here the selection aimed at showing how different kinds of information have been integrated within the TFS.

Consider the TFS representation of an intransitive verb such as **accelerate 11**:




First, the divergences of this TFS with respect to HPSG should be pointed out: PREF-VFORM and ACTION-TYPE are the verb attributes which have been inserted to represent Cobuild-specific information. PREF-VFORM, in a certain sense, duplicates part of the information contained in VFORM, with the main difference that whereas the latter encodes all the possible forms the verb can take, the former expresses only the preferred one (inferred from the definition text). ACTION-TYPE is part of the contextual information, and encodes what in the IT was called INFERENCE. Moreover, it should be noted that in the case of verbs the value of the LEXRULES attribute is different from nouns: it is defined by the features 3RDSING, containing the inflected form for the third person singular (the -s form), PRES-PART, containing the present participle (the -ing form), PAST, containing the past tense (the -ed form), and the PAST- PART, containing the past participle (the -ed or -en form). Other attributes which can be found as value of the LEXRULES attribute will be discussed below.

What remains to be seen is our idiosyncratic interpretation of the the subcategorization list, which—following the most recent version of HPSG (Pollard and Sag 1993)—has been jointly represented by the SUBJ and COMPS attributes. What changes with respect to HPSG is not the function of these attributes, but the specifications associated with them. In fact, the semantic preferences imposed by the verb on its arguments have been encoded as specifications on the elements of these lists. This has been possible thanks to the fact that in both cases the value assigned to these attributes is represented by lists of elements of type *synsem*, where *synsem* (as we have seen above) includes among its possible attributes the LEXSEM one. Let us turn back to the TFS above and see how the specification in SUBJ should be expanded:

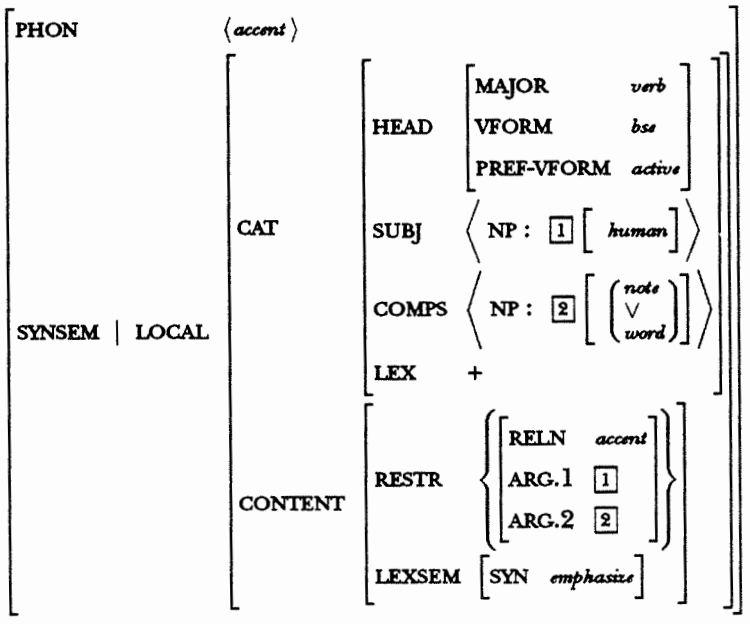



From this, we can see that NP denotes the SYNSEM value of a saturated nominal sign, and that the indexed value corresponds to the value of the relation which is in its turn the value of LEXSEM (ISA in the case at hand). Hence, the typical subject of **accelerate** is *speed* or *rate*, where *speed* and *rate* are semantic types rather than actual words. The fact that they are semantic types means that some of the specifications reported within the IT are inherent in this. If we compare this TFS with the IT from which it has been derived, which is reported below,

```
subj_info          : subj_postmods2      : of something
                     subj2               : specific: speed
                     subj_features2      : inanimate
                     subj_postmods1      : of something
                     subj1               : specific: rate
                     subj_features1      : inanimate
```

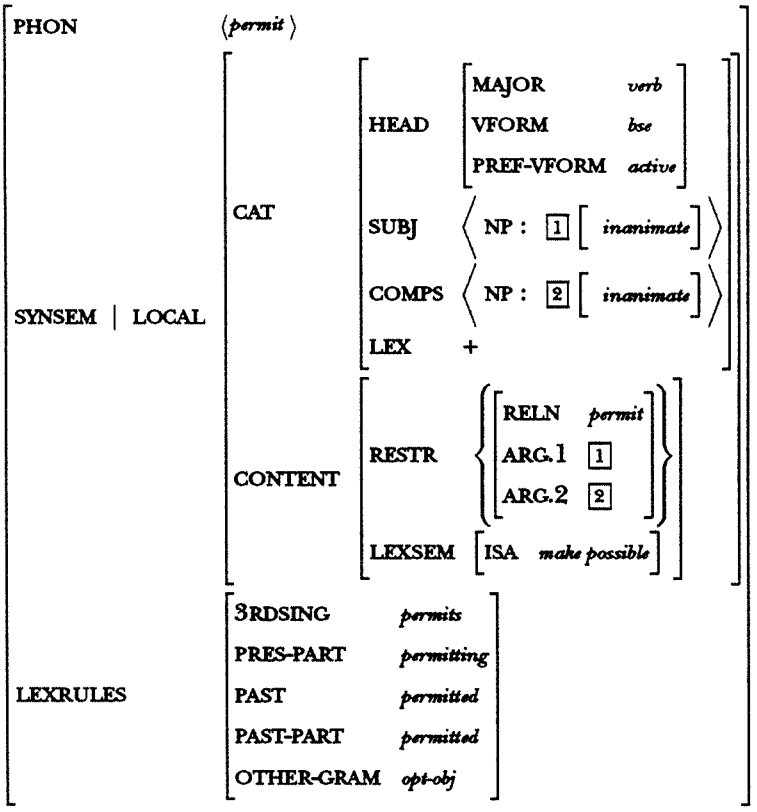
it can be noted that 'inanimate' subject feature does not appear any more as a restriction on the possible subjects of **accelerate** within the TFS representation; this follows from the fact that here *speed* and *rate* refer to types which are part of a semantic ontology, and their being 'inanimate' is implied by their definition as subtypes of a more general type 'inanimate'. This shows how the TFS entry contains, in the end, much more information than that actually shown in it, represented by information contained in the type system behind it.

Consider now the TFS representation for a transitive verb, exemplified by **accent 4**:



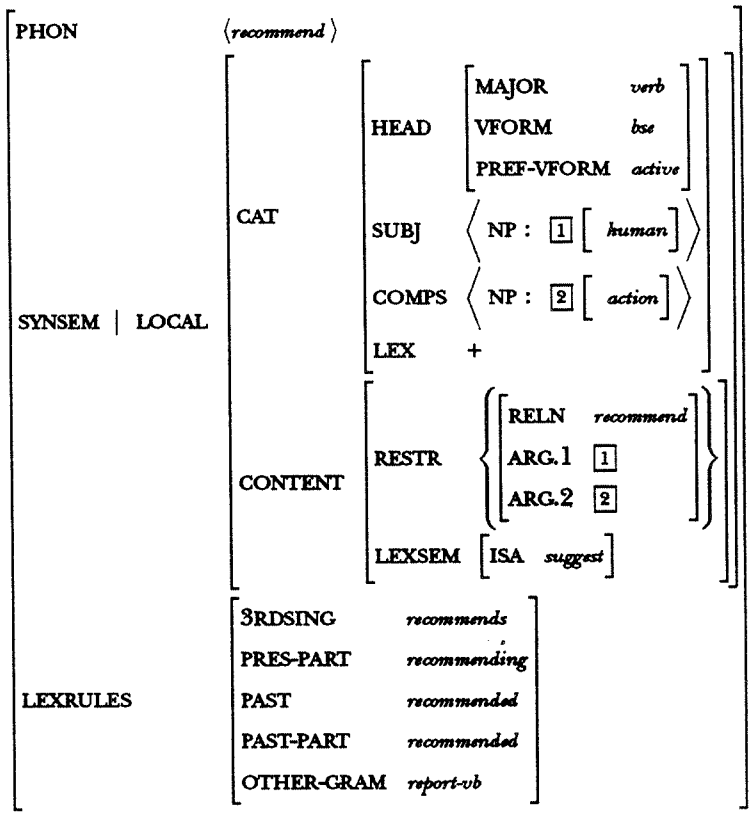

The main difference to be noted with respect to the TFS representation of intransitive verbs lies in the COMPS list, which is not empty.

The TFS entry below, corresponding to **permit 2**, represents another case of transitive verb, whose object is only optionally expressed; this information was originally expressed by means of the code VB WITH OR WITHOUT OBJECT.




As can be noticed, the TFS represents the transitive reading, whereas the intransitive one should be derived by means of a lexical rule: the attribute OTHER-GRAM in LEXRULES specifies the condition for this lexical rule of valency reduction to be applied.

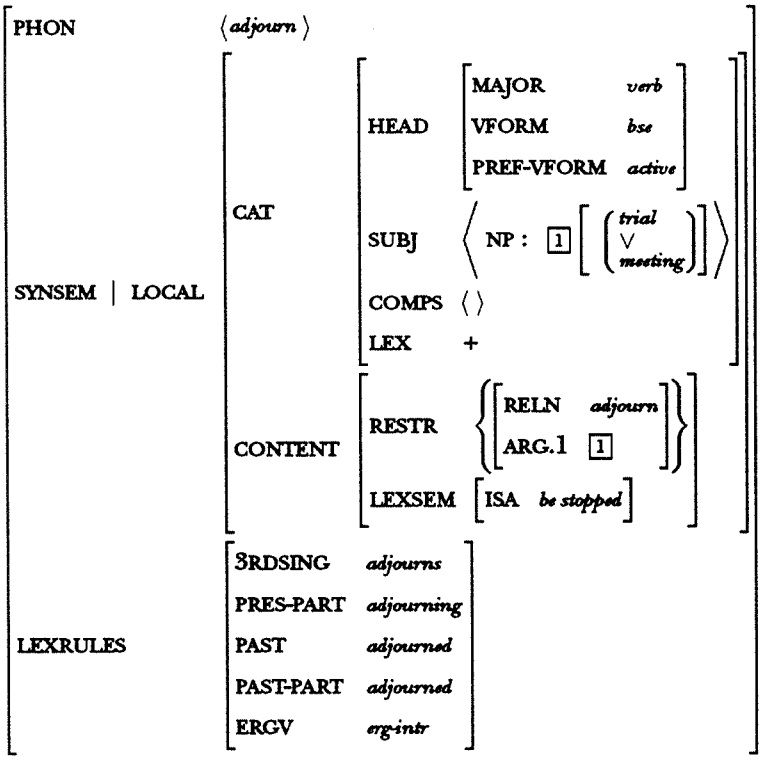
A similar case is represented by the TFS entry for **recommend 2**, where the Cobuild grammar code was VB WITH OBJ OR REPORT VB.




Again, the TFS represents the main grammar code, corresponding to the reading with a nominal object: the other reading of report verb should be derived by means of a lexical rule: the attribute OTHER-GRAM in LEXRULES specifies the condition for this lexical rule to be applied.

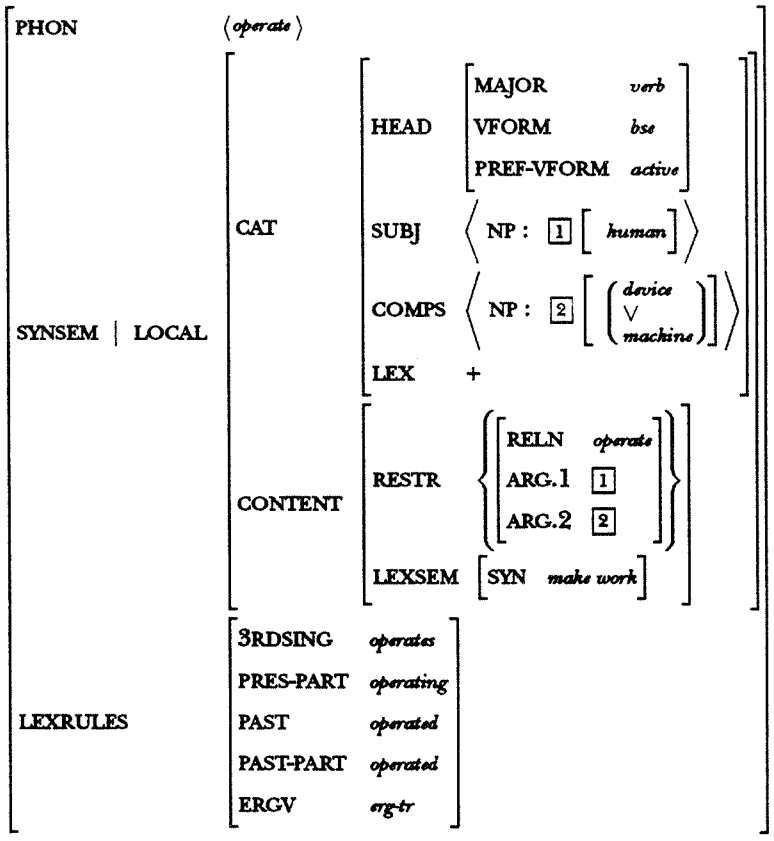
Let us now consider the case of ergative verbs. According to the dictionary data, we distinguish two different types of ergative verbs: i) ergative verbs typically used intransitively, and ii) ergative verbs typically used transitively. This information on the typical usage has been inferred from the LHS of the definition, whereas the grammar code is the same in both cases, i.e. ERG VB. This distinction contrasts with the standard treatment of ergative verbs, for which it is generally assumed that the transitive entry is the basic one, from which the intransitive one is derived via the application of lexical rules. Therefore, in this case the TFS representation is selected by considering both the grammar code and the information about the typical usage inferred from the definition.

The first case is here exemplified by **adjourn 1**, whose TFS representation is shown below:




The TFS corresponds to that of intransitive verbs. The ergativity information is registered as an attribute, ERGV, in LEXRULES, whose value is set in this case to *erg-intr*, meaning that the verb is typically used intransitively. A lexical rule, whose application is triggered by this specification, will generate the transitive reading by moving the subject to the object position (i.e. as top of the COMPS list), and adding a subject specification for which, unfortunately, no semantic information is available.

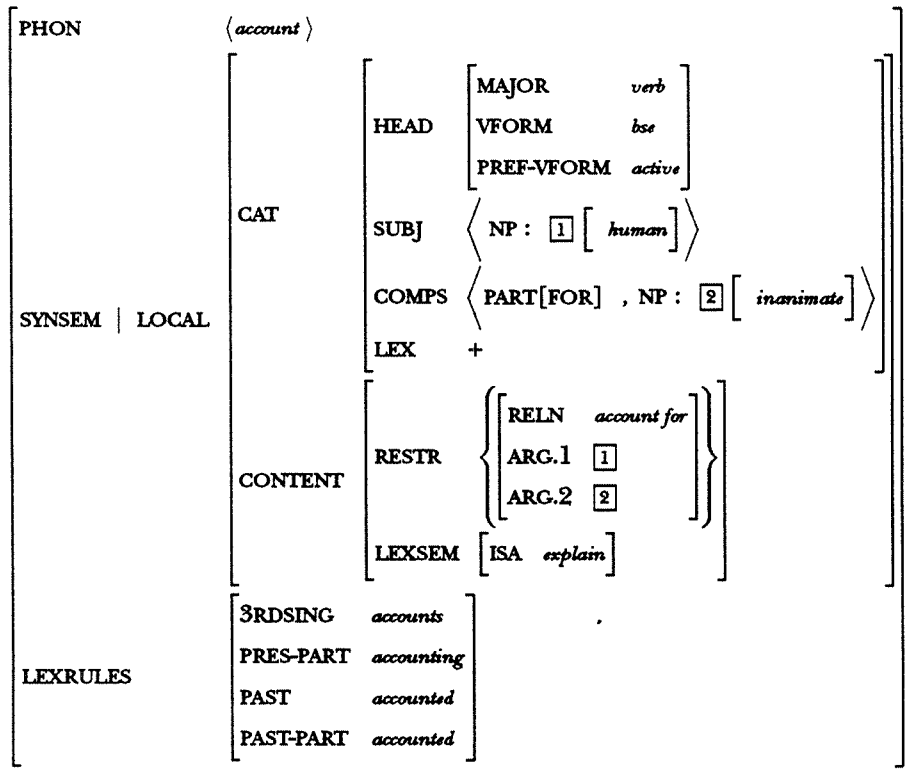
Case ii) is represented by **operate 3**:




In this case, given that the information concerning both subject and object is provided (the subject is preferably human, whereas the object is restricted to machines or devices), the basic TFS corresponds to that of transitive verbs. The attribute ERGV in LEXRULES has been assigned the value *erg-tr*: thus, the lexical rule operating on this sign will reduce the valency of the verb and at the same time will move the object, together with its semantic specifications, to the subject position.

At first glance, this treatment of ergativity might appear redundant: here we have two different entries for ergative verbs, and two different lexical rules operating on them, instead of just one basic representation and one lexical rule. But by unifying the representation of the two cases, the information about the typical usage would have been lost. So, behind this choice, there is the requirement, which has been constantly taken into consideration during this conversion and representation process, of reflecting the reality of the data.

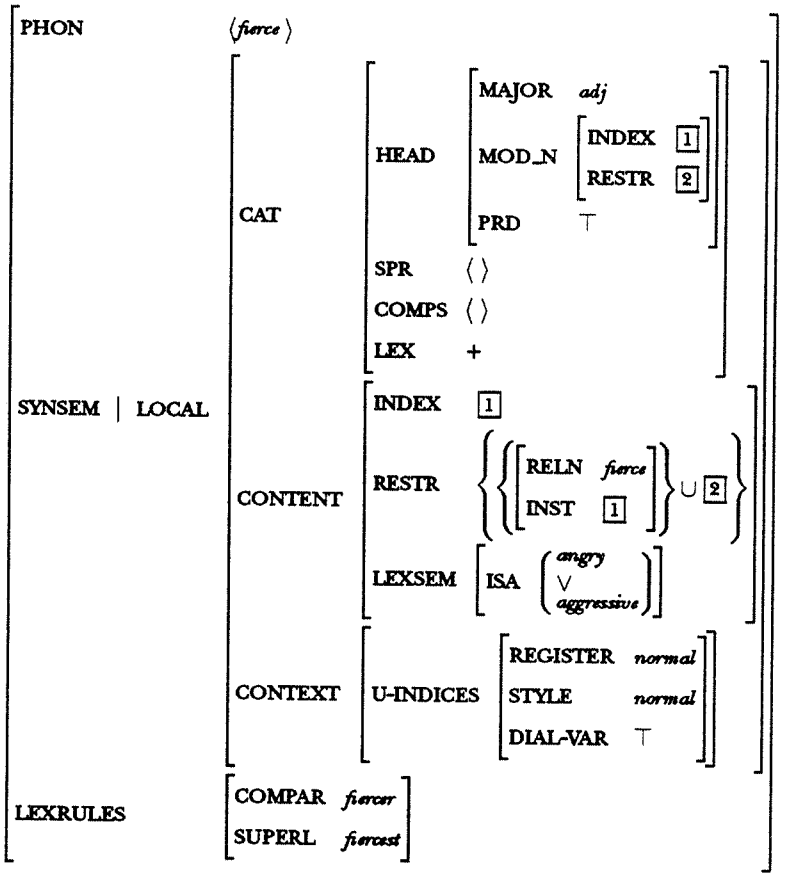
Let us conclude this brief survey of TFS verbal entries with the representation of phrasal verbs, exemplified below by the entry for **account for 1**:




In this case, the representation closely follows the standard HPSG treatment of this class of verbs. From the syntactic point of view, the *for* particle is subcategorized for by the verb, and thus appears within the COMPS list; this entails that the value of the attribute PHON does not include it. Semantically, this particle makes no direct contribution to the semantics: hence, what is defined from this point of view is the relation *account for* (see the value of the RELN attribute).

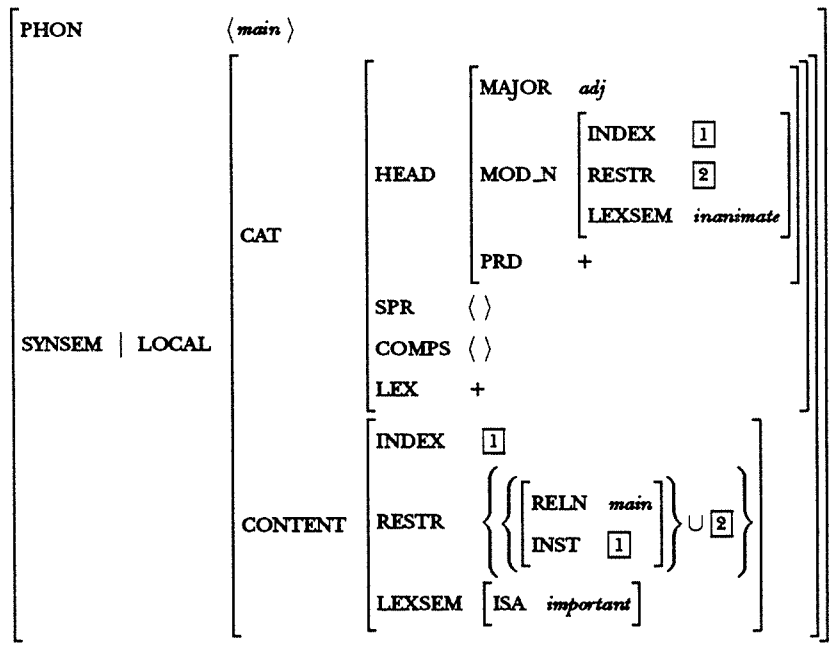
### 3.4.3.3 Adjectives

In what follows, a typology of TFS entries for adjectives is given. Consider the TFS entry for **fierce 1**:




In this representation, the only new Cobuild-specific attributes are represented by COMPAR and SUPERL given as value of LEXRULES: they are used to generate the comparative and superlative forms via lexical rules. The rest of this TFS conforms to HPSG for its general structure; the other Cobuild-specific attributes have been illustrated in the previous sections.

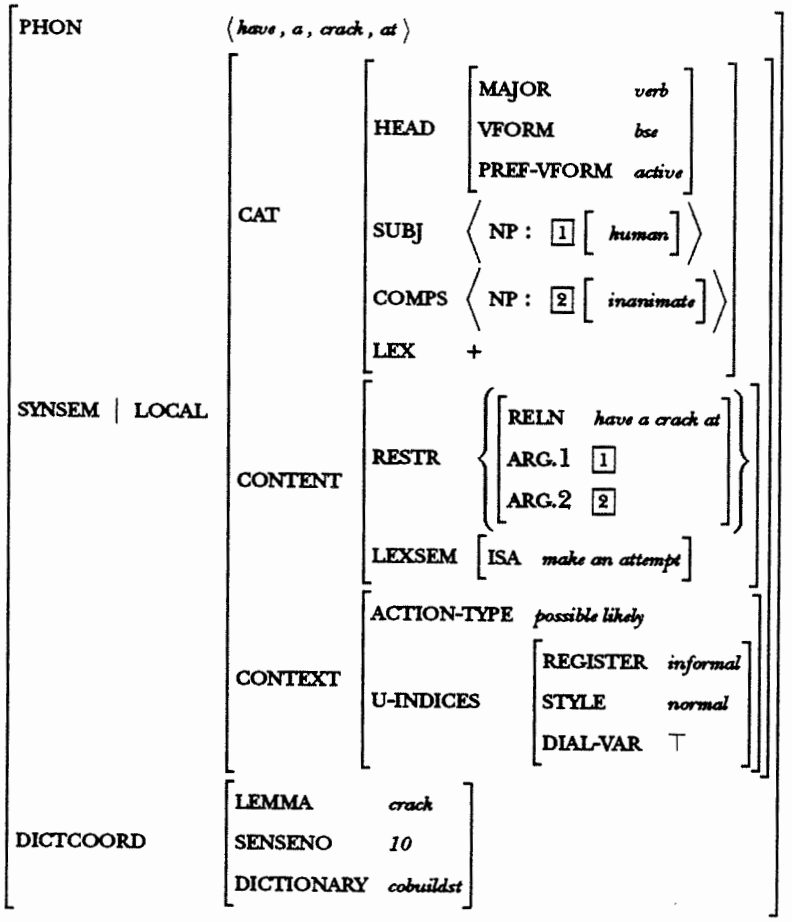
The TFS entry for **main 1** illustrates some other features used in the representation of adjectives:

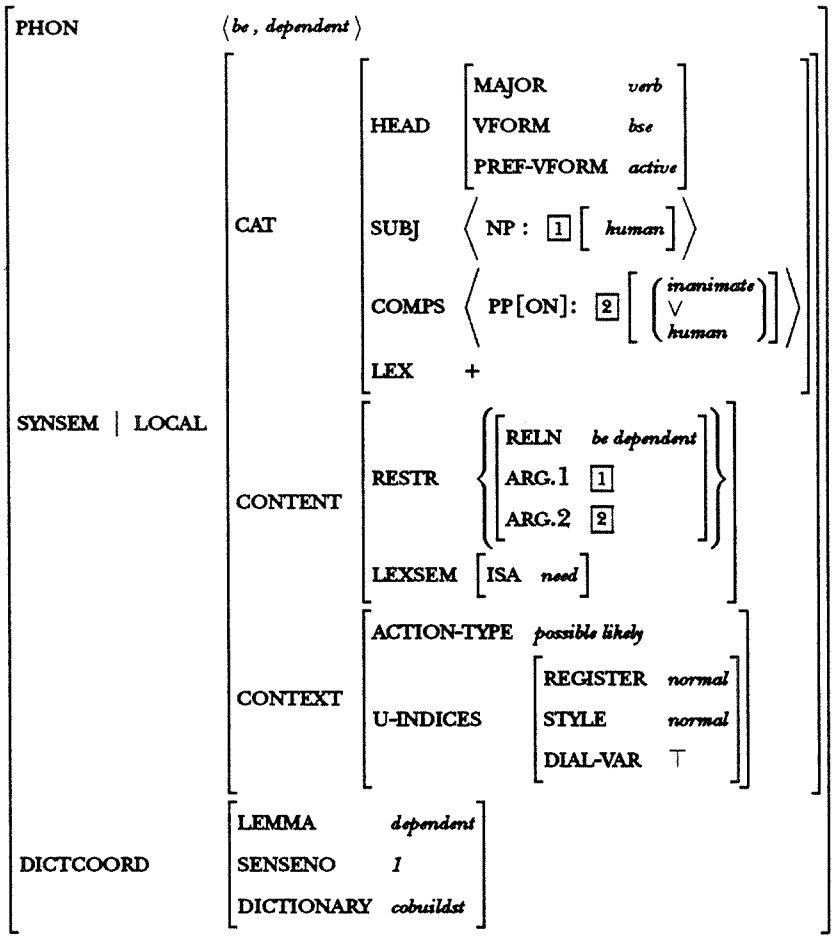



This TFS corresponds to the Cobuild grammar code ATTRIB ADJ, qualifying attributive adjectives. This information is represented by means of the attribute PRD, which is used in HPSG to distinguish predicative from attributive adjectives: in this case, this attribute has been assigned the value -. Another important feature of the TFS above is the representation of the semantic restrictions imposed by the adjective on the noun it modifies: such restrictions have been encoded within the value of the feature MOD_N, specifying the properties of the head noun selected by the adjective **main**. In particular, following the strategy adopted for the representation of semantic preferences on the complements of verbs and nouns, such restrictions have been encoded as value of the attribute LEXSEM; again, this is possible given that the value of the attribute MOD_N is of type *synsem*.

### 3.4.3.4 Representing nouns and adjectives as verbs

Two verbal entries, inferred from noun and adjective Cobuild entries, are given below. They represent in TFS terms the entries for the phrases **have a crack at** and **be dependent**, recorded respectively under the headwords **crack** and **dependent** respectively.






These entries are compact representations of phrasal expressions: they have been converted following the same strategy adopted in the representation of individual words. It should be noted that whereas PHON and SYNSEM attributes encode information about these expressions as a whole, the DICTCOORD attribute refers to the entry these phrases relate to. However, for these entries to be ready for use by NLP systems, a further conversion stage is required, during which an internal structure has to be assigned.

#### 3.4.3.5 The type system

Behind the TFS representations illustrated above lies a type system to which the individual lexical entries refer. In fact, the first step to be taken before representing any linguistic object in the form of TFS is the declaration of the types. The basic building blocks out of which feature structures, corresponding to linguistic objects, are constructed must be defined.

The construction of the type system behind our TFS lexical representations has been guided both by the empirical results of the knowledge extraction process from Cobuild entries, and by theoretical hypotheses concerning lexical information common to classes of words. Whereas the general structure of the lexical entry and the syntactic categories it refers to have been largely inspired by the HPSG theory, the lexical-semantic types have been extracted by the dictionary data.

The semantic preferences on complements and collocates of lexical items as well as the lexical semantic part of the lexical entry refer to semantic types which have been inferred by the analysis of dictionary data. In particular, we used the correspondences between the headword and its hyperonym, as well as other correspondences between parts of the left- and right-hand side of Cobuild definitions to build the semantic hierarchy to refer to from our TFS lexical entries. However, due to the sample of entries we have been dealing with (semantically heterogeneous and numerically

restricted), the semantic ontology we have built so far on the basis of the extracted data is far from being representative.

### 3.4.4 Constraints or preferences?

The choice to represent as fully as possible all the syntactic and semantic information that we derive from the COBUILD dictionary entries raises the problem of how such information could be used by a computational grammar. Is all the information extracted from a human oriented dictionary useful for NLP applications? We think that this question is crucial as far as the evaluation of our results is concerned. In particular, although we think that the semantic information associated with the complements and adjuncts of verbs and nouns, and the adjective collocates should be very useful for NLP purposes, so far no grammar formalism handles it in an exhaustive way. This kind of information should not be treated as absolute constraints whose violation makes a sentence totally unacceptable, but rather as preferences to be used, for instance, for the resolution of ambiguities (structural as well as lexical ones). In the context of this project, our proposal is restricted to the representation issue. How a grammar should treat this kind of information is far beyond the goals of the project.

Having decided that semantic information associated with the complements and collocates of lexical items must be represented, though it should not be treated as a constraint, we have had to propose a way of representing preferential information using a non-standard extension to standard HPSG. This has been a crucial point of our work. But this means that, at least for the present, the parts of our entry containing preferential information cannot be used as such by a standard, i.e. constraint-based, HPSG grammar. The question of the representation and use of preferential information is currently the topic of discussion among groups working in this area; our proposal concentrates on the representation side, but clearly also raises the problem of the exploitation of preferential information by NLP systems. Therefore, we hope that the preferential information represented within our entries can be used at a later stage of processing by a preference component operating on the results of the constraint-based grammar.

## 3.5 Using the Syntactic-Semantic Information

In this section, we will discuss various ways in which our lexical representations can be used. In the first part we list a range of possible applications for different types of users; in the second we go into more detail to specifically show how the syntactic-semantic information which has been derived for each lexical item can be used in an automatic procedure for dictionary sense disambiguation; finally we extend this proposal to suggest how the same kind of information could help in textual sense disambiguation, giving examples taken from the ITU corpus.

### 3.5.1 Human and machine oriented applications

As described in section 3.2 above, our results have been produced at two different levels: intermediate results; final results in the form of TFS entries. In the following, we will discuss briefly the possible applications of these different results for the three user types recognized by the project:

i. Human user
ii. Human user, assisted by the machine
iii. The machine

Obviously, the discussion here below refers entirely to the results that would be obtained once the parser has been applied to the whole dictionary.

#### 3.5.1.1 Intermediate results

At the end of the first stage of analysis, all the information that it has been possible to derive from the definitions is mapped onto an Intermediate Template. For each entry, the IT contains tagged, detailed and explicit orthographic, phonetic, morpho-syntactic, syntactic and semantic information. In particular, information on syntagmatically and paradigmatically related lexical items (e.g. complements and collocates on the one hand, and superordinates, synonyms on the other hand) has been derived by our analyses for each headword. Thus, the IT presents *explicitly* much information which is only contained *implicitly* in the printed dictionary. Examples of the format of our results at this stage can be seen in Section 3.3.

*Human user*

We feel that there is some scope for an application of these results by the non-computer user. The lexicographer, for instance, might find it convenient to examine the structure of his entries in this format on a printout or the screen. For example, a basic template has been defined for each grammatical category; the results could easily be sorted category by category and compared. In this way, inconsistencies and/or missing information are readily evidenced and can be marked for later correction on the machine. However, we feel that, similarly to other potential users, the lexicographer would be able to exploit the IT results far more usefully with the assistance of the machine.

*Human user, assisted by the machine*

User friendly interfaces could be easily implemented to make the information mapped on the IT readily available for different kinds of human users requiring detailed information on a lexical item, its semantic properties and its usage, e.g. translators, language learners, lexicographers again, etc. Different interfaces can be implemented to meet the needs of different types of users. This would be a very important application of our results: the dictionary user would have direct dynamic access to all the information contained in the lexical entries, wherever it has been stored, and in an interpreted form, instead of being bound by the restrictions imposed by the static alphabetical ordering of the printed volume.

As stated above, we feel that a particularly useful machine- assisted application of the IT results would be in lexicography. It is well known that the current trend for the representation of electronically generated lexical entries is to implement a TFS system. In general, computational linguists are very enthusiatic about the potential of such systems; however, they do not usually meet the same favour from the lexicographer, actually employed in day by day dictionary compilation. In fact, TFS entries are not easy for the human user to handle or analyse. The requirements of the formalism tends to make them "heavy" and they lack flexibility. The lexicographer generally is much more comfortable with a format that is readily interpretable without any need for any specialised training or expertise. We feel that our Intermediate Template meets this demand and would thus be an extremely useful tool in computer assisted lexicography. Furthermore, it can be used to represent information which so far is not handled by standard TFS formalisms. We would have no difficulty in implementing an interface so that the lexicographer could freely query, browse through,compare, merge and extract lexical information from the entries in the IT format.

*The machine*

Our Intermediate Template has been designed as the most convenient structure for a first computational model of the lexical information derived from the Cobuild definitions. Our TFS lexicon is then generated automatically by procedures which convert information from the IT into our type system. However, as the IT is entirely theory-neutral, automatic procedures can be developed to extract information and use it in the generation of any kind of computational lexicon.

#### 3.5.1.2 Final results

Examples of how the syntactic and semantic information extracted from the dictionary is represented in our TFS entries can be seen in Section 3.4.

*Human user*

We do not feel that there is a lot of scope for the human user to use our TFS entries in the printed form, especially the non-expert user. However, a printout could be useful for the linguist who wishes to examine the data in detail, perhaps in order to modify or add to the type system implemented.

*Human user, assisted by the machine*

On the other hand, we think that our TFS representation could be very useful for different kinds of computer-assisted human users, i.e. any user requiring a consistent formalized representation of the entry. In particular, they could be used by the overall dictionary editor (rather than by the general lexicographer responsible for the compilation of single entries) in a revision of the source or in the production of a new dictionary. The representation of the lexical entries in terms of types enforces a coherent structuring of the entire lexical system, and thus encourages coherence in design of new dictionaries and assists the correction of inconsistencies in the original, by evidencing clearly what is really pertinent in the definition of different kinds of entries.

*The machine*

We envisage different types of machine usage:

a. In the construction of computational lexicons for NLP applications: the information contained in our entries should be very useful for systems for the automatic analysis and generation of language. In fact, the information on usage of lexical items and their lexical and syntactic preferences encoded regularly in Cobuild definition statements and represented in our TFS entries is of great importance for NLP lexicons but is not easily derivable in other dictionaries.
b. In sense-disambiguation: we have already tested procedures which can be used for sense disambiguation within the source dictionary, i.e. disambiguation of the superordinates and of lexical preferences (see the next section); we have now begun to study the possibility of applying the information on the syntactic and lexical preferences of items for sense disambiguation in texts. For a first discussion, see section 3.5.3 below.

### 3.5.2 A strategy for sense disambiguation in the dictionary

Once our parser has been applied throughout the whole dictionary, over wide classes of definitions, we hope to implement a procedure now being experimented which exploits the syntactic-semantic information extracted for each lexical item to create, where possible, disambiguated, direct links on both (i) the paradigmatic axis and (ii) the syntagmatic axis, i.e. in the first case between the item and the correct sense of the relevant dictionary entry for the genus term, in the second case between the lexical item and the correct sense of its different arguments and modifiers.

#### 3.5.2.1 Genus term disambiguation

The paradigmatic links such as synonymy, hyponymy, hyperonymy, as well as meronymy, are those usually extracted and formalized from MRD definitions and exploited in the construction of a hierarchical lexicon (as in Acquilex and other similar projects). A major problem has always been to connect the entry item to the correct sense of the genus. In their description of the Cobuild definition statements, Allport et al. (1993) point out that the two parts of the definition (the LHS and the RHS) are in a relationship of equivalence: "The left part sets up matches which the right part must match or otherwise take account of". In fact, we have seen that preferences that have

been extracted as holding for the word being defined are shared by the genus term and can thus be used to disambiguate it, i.e. the combination of the information that has been extracted from the LHS and the RHS of one definition can then be projected onto the LHS's or RHS's of all the definitions listed by the dictionary for the lexical item corresponding to the genus term, searching for matches that will allow us to identify the right sense. For instance, if we refer back to the example of **apply 4** in the previous section:

**4** VB WITH OBJ
If you **apply** a rule, system, or skill, you use it in a situation or activity

we can assign the subject (+hum) and the object preferences (obj_specific: rule, system or skill, obj_features:-anim, +count) attached to **apply** in this sense to the potential arguments of its superordinate **use** in order to attempt to identify which sense of **use** is implied here.

The Student's dictionary entry for **use** has 7 different senses and 3 of them are for verbs:

**1** VB WITH OBJ
If you **use** a particular thing, .....

**2** VB WITH OBJ
If you **use** a particular word or expression, .....

**3** VB WITH OBJ
If you **use** people, .....

In each case, the subj_features (+hum) which we could derive from these definitions would match with those that we have potentially assigned to **use** as superordinate of **apply 4**. However, the best match as far as the object preferences are concerned would be with **use 1**. Sense 3 is immediately excluded as the obj_features that we derive from people include +hum, whereas in sense 2, while the subj_features would match, the values for obj:specific would clash (rule, system or skill vs. word or expression).

Another example of this kind of genus disambiguation strategy is shown by **function 2**:

**2** VB
If a machine or system **functions**, it works.

In this case, **work** is recognised as the genus term, and our parser tags it as a synonym. The semantic preferences attached to **function**, and thus assigned to **work**, are represented on the IT as follows:

```
subj_info      : subj1             : specific  : machine
                 subj_features1    : -anim, +count
                 subj2             : specific  : system
                 subj_features2    : -anim, +count
```

If we look at the dictionary entry for **work**, we find 16 sense divisions; 9 of these are for verbs. However, it will be seen that the best match for **work** with the above preferences is clearly sense no. 8: VB "If a machine or piece of equipment **works**, ....", where we again find machine as subject.

Our last example is that of **resolve, 3** mentioned above in 3.3.2.1 in the discussion on recognizing phrasal verbs as genus terms.

**3** VB WITH OBJ
To **resolve** a problem, argument or difficulty means to deal with it successfully.

Our parser gives **deal with** as the value of exp_superordinate. The preferences attached in this case to **resolve** and thus assigned to **deal with** are:

```
obj_info       : obj1              : specific  : problem
                 obj_features1     : -anim, +count
```

```
obj2              : specific  : argument
obj_features2      : -anim, +count
obj3              : specific  : difficulty
obj_features3      : -anim, +count
```

Of the two senses given for the phrasal verb **deal with**:

1 PHR VB
When you **deal with** a situation or problem, ...

2 PHR VB
If a book, speech or film **deal with** a subject, ...

the first sense is chosen as being the correct sense of the genus term under examination. This strategy appears to work well in disambiguating the genus term for verbs, although at times it may only help by reducing the possibilities rather than identifying a unique sense. Other values, in addition to argument selection features, that could be used to find the best sense match are those for the grammar and the inference attributes.

We are now evaluating to what extent it is feasible to use a similar strategy to disambiguate the genus term for nouns, again using the equivalences established between the two sides of the definition to assign the features that have been attached to the entry item also to the genus term. However, our first results are less encouraging than when working with verbs. The main problem is that the Cobuild definitions for nouns tend to be less generous with collocational information on the LHS. Thus, it is not generally likely that this information will be available or sufficient in itself to permit us to disambiguate the genus term.

Let us consider a few examples. The definition for **power 4** in the dictionary is:

4 UNCOUNT NOUN WITH SUPP
The **power** of something is its physical strength

where, in our input data, **strength** is tagged as superordinate of **power** and "of something" on the LHS is matched with "its" on the RHS. Our parser gives the following output on the IT for this sense of **power**:

```
sense_no       : 4
gram           : UNCOUNT N with SUPP
entry_info     : entry              : Power
genus_info     : prov_superordinate : strength
                 is-a               : strength
colloc_info    : colloc_prep        : preferred   : of
                 colloc_features    : -anim
```

The entry for **strength** in the dictionary gives 8 definitions, all refer to nouns, but the best match with the preferred collocational information that we have transferred to **strength** in this definition on the basis of its relationship with **power** is with sense 5 UNCOUNT N "The **strength** of an object is its ability to withstand rough treatment or heavy weights". Cfr. 1. "Your **strength** is ...", 2."**Strength** is also courage or determination", 4. "Your **strengths** are ..."6. "The **strength** of a feeling or opinion is ...", 7. "The **strength** of an opinion, argument or story is ...", 8. "The **strength** of a relationship is ...". The only other sense for which a match could be recognized is 3, UNCOUNT N. "You can also refer to power or influence as **strength** ..." where the potential matching would be between the headword of the definition we are examining (**power**) and the genus term. However, **power** in this definition seems to refer to sense 1 of the dictionary and not to sense 4. In fact, if our procedure were extended to take into account the example given with this sense in the dictionary, "the enormous strength ... of the unions" it would perhaps exclude it as a possible match.

This particular example gives a good result using the proposed method but, in general, it appears that the strategy needed for sense disambiguation is more complex for nouns than for verbs and that other kinds of information normally play a role. Consider the following definition

for **function** 1, COUNT N, The **function** of something or someone is its purpose or role, where "purpose" and "role" are tagged as synonyms of **function** by our parser. The entry for purpose gives 3 senses:

1 COUNT N
The **purpose** of something is the reason ...

2 COUNT N
Your **purpose** is the thing that ...

3 UNCOUNT N
**Purpose** is the feeling ...

where we find that the information we extract on collocational preferences for **function** (colloc_features: -anim, +hum) would allow us to match its genus **purpose** against both dictionary senses 1 and 2 (the purpose of something, your purpose = the purpose of someone), and the grammar information also matches in these two senses. Thus our range of choice is reduced but not exhausted.

Whereas, if we look at the following dictionary definitions for **role**, we find that the genus term for sense 1 actually includes the word "function" (a typical example of (semi- )circularity in the definitions which lexicographers try to avoid although at times it appears to be inevitable) and it is this that will permit our procedure to select the correct sense, rather than the collocational information.

1 COUNT N WITH SUPP
Your **role** is your position and function ....
(genus: position and function)

2 COUNT N
A **role** is one of the characters that ....
(genus: characters)

#### 3.5.2.2 Argument disambiguation

We have also examined methods to create disambiguated syntagmatic links, i.e. to be able to recognize the correct sense of the words used as arguments of the headword in the definitions. This disambiguation will be driven by both the syntactic and semantic features automatically extracted from the definitions. For instance, the Cobuild definition for **pick up 4** gives us "If you **pick up** a skill or idea ...". From this, we have extracted **skill** as one of the preferred arguments for **pick up** in this sense and, on the basis of the indefinite article, we have assigned the feature "+count" to **skill**. The dictionary entry for **skill** gives two senses: count and uncount. We can thus automatically select the right sense of **skill** when it is an argument of **pick up**. Again, using this strategy, it is not always possible to immediately identify the correct sense of an item, often we can just reduce the possibility of choice.

In any case, we think that a second strategy based on the semantic features that our analysis attributes to a given argument is more interesting. For example, with **abdicate 1**, "If a king or queen **abdicates**, he or she resigns" we identify as specific subjects "king" and "queen" for which, on the basis of correspondences between LHS and RHS, the features +hum, +male, and +hum, +female, respectively, have been inferred. This will permit us to map directly to the first sense of the dictionary entry for **king** which has "man" as superordinate (in the other senses of **king**, the superordinates are (chess) piece, and playing card), and also to the first sense of **queen** with superordinate "woman" (the other senses have (chess) piece, playing card, and also bee as their superordinates).

Indeed, connecting the verb arguments or, more generally, the words appearing in co-text1 and co-text2 with the corresponding lexical entries will add a further dimension to our lexical network, that of syntagmatic relations. In this way, the lexical network resulting from our formalization of the Cobuild dictionary will include two dimensions: that of paradigmatic and that of syntagmatic links.

#### 3.5.2.3 Towards the lexicon as a set of relations

We thus move from the dictionary considered as a list of distinct entries towards the construction of a series of combinatorial links, so that lexical items no longer exist in isolation but are continuously associated with all the others that can enter into relationship with them (see Calzolari (1990) for a discussion on the lexicon as a complex set of relations). Two aspects must be stressed: (i) the dictionary itself frequently provides the means to construct automatically disambiguated links, both in the paradigmatic and syntagmatic directions; (ii) the extraction of syntagmatic or phrasal links is a peculiar feature of this project and allows us to encode in the formal lexical entries that we are creating both their syntactic environment and the semantic preferences on their neighbours (whether arguments, modifiers, governors, etc.). From a theoretical perspective, these results, which blurr the familiar decoupling of lexis and syntax, could also be seen as an attempt to formalize the linguistic hypotheses behind the Cobuild dictionaries.

### 3.5.3 Testing the Pisa representations on the ITU Corpus

In this part of the report, we will discuss how we could test the TFS representations of the lexical entries, that we have derived from the parsed definitions received from Birmingham, on the ITU corpus. We are, however, restricted in our range of testing as we have just the approximately 400 analysed definitions of the test vocabulary available and in only a very few cases do we have a set of all (or most) of the definitions for the different senses of a single headword. Therefore, at this stage, we are unable to perform an exhaustive testing of our results against the Corpus and our discussion must be, to a large extent, hypothetical.

As was to be expected, a number of difficulties are encountered when attempting to test lexical entries that have been derived from an all-purpose learners' dictionary, constructed on the basis of the evidence provided by a general corpus, against a technical vocabulary. First of all, it should be remembered that the distinction between a general and a technical corpus is not restricted merely to the choice of vocabulary used but includes important differences in the style and structures employed, and these condition considerably the characterization of the lexicon. In the specific case of the ITU corpus we can mention, for example, the considerable use of the impersonal or passive constructions and the frequency of occurrence of nouns used attributively with other nouns, e.g. earth station, message source, telephone channel, etc., etc.. For these reasons, we feel that more valid results could probably be obtained by testing our entries against a general language rather than a technical corpus of this type.

One of the main tenets of the Cobuild theory is that words only acquire sense in context and it is possible to distinguish between the different senses of a lexical item on the basis of its syntactic structure and of the words with which it co-occurs. This is one of the main reasons that led Cobuild lexicographers to systematically specify, for each word sense, the lexical preferences exerted by the lexical item on its arguments, complements or collocates. The evidence provided by the corpus on word usage is thus encoded regularly in the definition statements and represented in our entries. We are convinced that this is very important information for the lexical component of many NLP applications and is needed by systems for both analysis and generation.

Our aim here is to test to what extent this preference information can be used to distinguish between different word senses in the ITU corpus, i.e. between cases of lexical and grammatical homography.

In the first part of this discussion, we have concentrated our attention primarily on verbs for two reasons:

i. Much of our initial work on the analysis of the Cobuild definitions focussed on this word class as it was richer in lexical and syntactic preferences, and thus our verb entries tended to be more fully characterized with respect to usage;
ii. It is known that in a technical language, it is mainly the nouns that bear the weight of topic specificity, i.e. the technical message; the verbs tend to be of more general sense. Therefore, the nouns in the ITU corpus often have senses which are not represented in the Students'

dictionary and/or are used in just one sense; they thus tend to be of little interest from the sense disambiguation viewpoint.

The Cobuild dictionary is human-oriented and very frequently our verbs are characterized as having a preference for a human subject. A technical corpus obviously reveals a different usage; as stated by the Birmingham analysis, the words are generally used in an impersonal context. The first rule to be assumed, therefore, when testing our entries on the corpus was that a +Human preference on the subject of a verb was to be ignored.

However, again, one of the problems facing us was that the verbs contained in the ITU corpus and included in our vocabulary subset tended to be used in just one sense, or in a sense not contained in our set of entries. For instance, if we examine the last example given by Birmingham above for **address**, we find that the dictionary gives three definitions for **address** as a verb:

**2** VB WITH OBJ
If a letter **is addressed** to you, your name and address are written on it.

**4** VB WITH OBJ
If you **address** a group of people, you give a speech to them.

**5** VB WITH OBJ
To **address** a problem means to deal with it.

and only one of them, No. 4, is contained in our test vocabulary. Unfortunately, this is not the sense contained in the corpus listing:

```
CCIR has been meeting since 1983 to address issues related to ISDN performance
```

which clearly reflects sense no. 5.

The lexical entries which we would derive from the three verb definitions would each be characterized by a specific preference on the object. (In the case of sense 4, the human preference assigned to the subject would be cancelled by the fact of treating a technical corpus, senses 2 and 5 would not be assigned a subject preference by our procedures.) For convenience, in this hypothetical discussion - hypothetical partly because we have only the parsed definition statement for **address 4**, we refer to the entries as they would appear structured in our Intermediate Template as they are easier to read. Here below, we give first the full entry for **address 4**:

```
def_no              : 337
sense_no            : 4
def_type            : 1
lemma               : address
entry_info          : entry               : address
                      norm_entry          : address
genus_info          : prov_superordinate1 : give a speech
                      syn1                : give a speech
                      genus_prep1         : to
inflection          : address addresses addressing addressed
gram                : VB with OBJ
voice               : active
inference           : possible likely
subj_info           : subj_features1      : human
obj_info            : obj_postmods1       : of people
                      obj_features1       : human-coll
```

We can assume that the obj_info for **address 2** and **address 5** would be as follows:

```
address, 2:
obj_info            : obj_specific        : letter
                      obj_features        : -anim, +count
address, 5:
obj_info            : obj_specific        : problem
                      obj_features        : -anim, +count
```

The "letter" of sense 2 and "problem" of sense 4 would then be referred to the relevant semantic type, once the semantic hierarchy has been constructed from the data. Thus "letter" and "problem" should here be considered as if they were meta-linguistic terms, representing the set of lexical items (intended as word senses) that will be mapped into the relevant semantic sets. We now show how these representations of the three senses of address, VB WITH OBJ, can be matched against the Corpus listing.

In the concordance **address** is found in the same form (infinitive) as in the definition with the object "issues". If we look at the entry for **issue** in the dictionary, we find that the appropriate sense is defined as: "An **issue** is an important problem or subject that ....." where the superordinate is "problem | subject". In our semantic type hierarchy, this sense of **issue** would be linked to the semantic type of "problem" and a correct (automatic) sense matching for the corpus example against the lexical entry derived from definition 5 in the dictionary should be possible.

Let us now give another more complete example from the ITU corpus to show how it should be possible to use our entries to help to disambiguate between different senses of a lexical item: both verbs and nouns. The entry for **function** has 5 definitions in the dictionary:

1 COUNT N
The **function** of someone or something is their purpose or role.

2 VB
If a machine or system **functions**, it works.

3 VB WITH 'AS'
If someone or something **functions** as a particular thing, they do the work or fulfil the purpose of that thing.

4 COUNT N
If one thing is a **function** of another, its amount or nature depends on the other thing.

5 COUNT N
A **function** is also a large formal dinner or party.

We have analysed the first three of these definitions and they are represented by our Intermediate Template as follows:

```
def_no              : 11122
sense_no            : 1
def_type            : 3
gram                : COUNT N
lemma               : function
entry_info          : entry              : function
                      norm_entry         : function
genus_info          : prov_synonym2      : role
                      syn2               : role
                      prov_synonym1      : purpose
                      syn1               : purpose
inflection          : function functions
art_info            : preferred: def
postcolloc_info     : colloc_prep        : preferred: of
                      colloc_features2   : human
                      colloc_features1   : inanimate

def_no              : 11123
sense_no            : 2
def_type            : 1
lemma               : function
entry_info          : entry              : functions
                      norm_entry         : function
genus_info          : prov_synonym1      : works
                      syn1               : work
inflection          : function functions functioning functioned
gram                : VB
voice               : active
inference           : possible
```

```
subj_info          : subj2              : specific: system
                     subj_features2     : inanimate,+count
                     subj1              : specific: machine
                     subj_features1     : inanimate,+count

def_no             : 11124
sense_no           : 3
def_type           : 1
lemma              : function
entry_info         : entry              : functions
                     norm_entry         : function
genus_info         : prov_synonym2      : fulfil the purpose of
                     syn2               : fulfil the purpose of
                     prov_synonym1      : do the work
                     syn1               : do the work
inflection         : function functions functioning functioned
gram               : VB with 'as'
voice              : active
inference          : possible
subj_info          : subj_features2     : inanimate
                     subj_features1     : human
clause_info        : clause             : as a particular thing
```

In the corpus, we found 147 listings of the form "function(s)". For space reasons we list just a small subset below:

```
  1)satellite is the core of the network and performs the
*function* of radio-relay in the sky using
  2)In the international area, global systems such as
INTELSAT *function* as international common carriers of
traffic between major land
  3)for which the CCIR criteria are given, as a *function* of
the radio-frequency carrier-to-
  4)4 -Antenna gain and 3 dB beamwidth as a *function* of
antenna diameter for practical antenna efficiencies
  5)is shown in Fig. 2.6 as a *function* of frequency for a
distance of 36 000 km
  6)terms of a change in antenna gain as a *function* of the
off-boresight angle
  7)r.p. emitted by the satellite is a *function* of the
level Cu of the signal received at
  8)/T)p can be expressed as a *function* of separate
interference contributions from adjacent frequency satellite
channels
  9)12 gives the satellite elevation angle and distance as a
*function* of the earth station's latitude and longitude
 10)angle R: distance of the satellite as a *function* of: -
relative station longitude,
.....
 13) can be conveniently approximated by a Bessel *function*
series expansion as: In the above
 14) approximation, J1 denotes the first-order Bessel
*function* of the first kind, bp the complex coefficients
.....
 22)Digital Circuit Multiplication Equipment (DCME) The
*function* of the digital circuit multiplication equipment is
.....
 80)the multidestination mode. In this case, the
multidestination *function*  is carried out by the DCME. In
consequence
```

As we can see here (Concordances 3–10), and as is confirmed by a complete scanning of the concordances, the vast majority of occurrences of **function** in this technical vocabulary reflect the Cobuild sense 4, which in the dictionary has been indicated as a formal use. It is interesting to note how a sense which is infrequent in a general corpus (Cobuild definitions are ordered by frequency of use) is the most common in a technical corpus. Another quite frequent sense is that attested by Concordances 13 and 14: **function** in the sense of mathematical correspondence; this sense is not

given at all in the dictionary. The other senses shown here are sense no. 1 (Concordances 1, 22, 80) and sense no. 3 (Concordance 2). In the entire concordance listing, this is the only occurrence of sense 3 and we found no evidence of senses 2 and 5. While this was to be expected for sense 5, we were a little surprised to find no example of sense 2.

Using the methodology described previously for **address**, it should be possible to use both the syntactic and semantic information we have derived for the different senses of **function** listed above to help us to distinguish between senses 1, 3 and 4 in the concordances. With reference to Concordance 2, the important clue is given by the grammar code VB WITH 'as' and this should be sufficient to indicate the sense. For the first noun sense, from our analysis we have derived that this sense of **function** preferably takes a definite article, is followed by a prepositional phrase with "of" + collocate (inaminate | hum), and has as superordinate purpose | role. Without going into a detailed discussion, it can be seen intuitively that the 3 concordances (1, 22, 80) for this sense respect this pattern.

For dictionary sense 4, we would derive that this sense is followed by a prepositional phrase with "of" + inaminate collocate. The important aspect of the definition for this sense (If one thing is a **function** of another, its amount or nature depends on the other thing) is that the two "things" cited are clearly equivalent and the superordinate amount | nature refers to them and not to the headword **function**. Our analysis would reflect this and indicate that function in this sense is preferably followed by a PP consisting of "of" + an inanimate item belonging to the "amount" or "nature" semantic classes.

This pattern is revealed in all the cases of sense 4 in our concordances. At times the correspondence can be traced directly by following the taxonomic links through the dictionary, e.g:

```
Concordance 4: diameter -> length -> amount
Concordance 10: latitude | longitude -> distance -> amount
```

where "length" is the superordinate for the appropriate sense of "diameter" in the dictionary, "amount" is the superordinate for the appropriate sense of "length", and so on. At times the correspondence will only be evident when a complete semantic type system has been constructed for the lexicon. On the contrary, in our few examples of **function** 1, when present, the preposition "of" always governs a concrete item, e.g. station, module, equipment, etc.

To sum up, these two noun senses of **function** (1 and 4) in our concordances are distinguished by both syntactic information (the use of the definitive article for sense 1) and semantic information (the presence of a word belonging to the semantic class "amount" in the PP following function in sense 4).

We think that this example gives a realistic idea of how our lexical entries and, in particular, how the information derived on the syntactic and semantic preferences of a lexical item could be an important part of a set of procedures for text sense disambiguation. Of course, such information is not usually sufficient in itself for sense disambiguation and will have to be combined with other types of syntactic and semantic analyses performed on the text (e.g. POS tagging, listing of possible superordinates for major word clases). It is our intention to continue in the future with experiments in this direction.

# 4 Logical Aspects of the Dictionary

## 4.1 The logical structure of Cobuild definitions

Unlike other dictionaries, Cobuild exploits the inferential capacity of human language in a strikingly explicit way. The language of its definitions directly reflects a crucial theoretical assumption underlying its design. Consider, for instance, the statement below (which of course analogously holds for all of the remaining definitions in Cobuild),

> If you use the word **boy**, you can expect to be presumed to be talking about a male child.
> (Hanks 1987:135)

which nicely encapsulates the legitimate basis for our logical reading[3] of the explanations in the dictionary. Our mapping of[4]

1 COUNT N
A **boy** is a male child.

to a logical expression of the (here quantifier-free) expression of an approach we have termed Bochum Logical Form (henceforth BLF)

$$boy.x \rightarrow male.x \wedge child.x$$

serves to make this inferential approach in lexicography formally explicit and to incorporate the analysis in an integrative linguistic framework.

Furthermore, our reading of the definitions as conditional statements facilitates the analysis of the dictionary as an ordered self-contained whole and not as a huge disjunction of entries merely ordered on an ergonomic—not theoretical—basis, i.e. alphabetically. The inherent logic of the Cobuild dictionary constitutes a subtle and extremely complex natural order which can now be expressed by the formal means of BLF. Given the adequate representation of such an inherent logical structure, the mapping of Cobuild's natural language definition text to inheritance-based formal linguistic theories is feasible.

HPSG supplies a rich, integrated linguistic apparatus involving multiple inheritance mechanisms which meet the logical requirements of the dictionary. Also, this framework involves a situation semantics component that is in concordance with basic assumptions underlying the Cobuild philosophy and BLF. The relationship between the so-called CONTENT and CONTEXT features of HPSG which we will employ reflects the relational theory of meaning in situation semantics. It encodes the correlation between an utterance situation and a described situation, or, viewed from a different perspective, fulfils the implicit lexicographic requirement contained in the following statement:

> All statements about word meaning are statements about word use.
> (Hanks 1987:135)

which, again, puts a considerable part of the Cobuild (and our) philosophy in a nutshell.

The attribute-value system employed for modelling linguistic objects in HPSG makes the theory suitable for at least partial mapping to ALEP syntax. This ensures the applicability of the results in NLP.

---

[3] Cp. Deliverable 1.

[4] This example, like all the others in the remaining sections, was taken from *Collins Cobuild Student's Dictionary*.

The crucial goal of the present approach then is to facilitate the integration of four theories:

- Cobuild
- Bochum Logical Form
- Head-driven Phrase Structure Grammar
- Situation Semantics

which should supply the answers to a number of questions:

- What are the common assumptions underlying these theories?
- What are their respective positions to each other?
- How much of the information representable in each theory can be mapped to the others?
- Will the presumed mapping process necessitate essential adaptations to the theories?
- How much information will get lost in the process?

Though in many respects different, the theories have much in common. The most important link between them is of course the handling of information by a strictly inferential approach. In fact, the impact of this common feature is so strong that it annihilates the—only apparent—discrepancy between the practically oriented approach of Cobuild and the mostly theoretically motivated views of the remaining theories.

Viewed from a Cobuild perspective, several feature complexes of HPSG could receive a much more profound treatment than could be achieved so far. Also, the bulk of necessary information for that can be transferred by linking Cobuild to HPSG and Situation Semantics via representation in BLF.

In short, the theories Cobuild, BLF, and Situation Semantics remain untouched by the present approach, whereas HPSG will be enriched by changes totally in concordance with its philosophy. The major aspects are contextual information, semantic types, semantic roles and selectional preferences.

Let us recapitulate now, in the briefest terms, the different steps of formalisation within BLF that were suggested in (Schnelle 1992). We will start with the example

**2** COUNT N
An **orange** is a round orange fruit that is juicy and sweet.

Note that there are two occurences of the term **orange**, one denoting the object being defined and one specifying a certain property, i.e. color, of it.[5] Now, this doesn't create a problem for a human user—he or she will be able to tell the semantic information of the first one from that of the second one in a split second. For a formal system, the situation is not so obvious. Also, a round object is defined in the dictionary as being shaped like a ball or a circle. An orange, however, does not have the shape of a disc, but that of a sphere. Again, this doesn't matter in the human user, but might startle a formal system, in which this disjunction of two informational items would have to be resolved. Hence one needs a

- Representation in disambiguated dictionary English:

  *An orange.2 is a round.1.1 orange.1 fruit.1 that is juicy.1 and sweet.1.*

Here, the indices of the annotated expressions correspond to the prefixed reading number in the Collins Cobuild Student's Dictionary explanations.

[5] It is fairly obvious that the distinction between the two expressions is not only of semantic but also of syntactic nature. Hence, checking the grammatical information given by the respective Cobuild types immediately following the reading numbers would certainly resolve a large number of ambiguities.

In order to arrive at a rigorously structured format of dictionary definitions that also captures the inferential power of natural language expressions, all explanations are transferred to *if ... then* statements, which are yielded by the

- Representation in logically regularised dictionary English:

  *If something.x is an orange.2*
  *then something.x is a round.1.1 orange.1 fruit.1*
  *that is juicy.1 and sweet.1*

These statements can then be represented in a normalised conjunctive form like in the expression given below:

*If sth.x is an orange.2*
*then sth.x is round.1.1*
*and sth.x is orange.1*
*and sth.x is a fruit.1*
*and sth.x is juicy.1*
*and sth.x is sweet.1*

Let us assume that an orange is defined in the dictionary as a specific citrus fruit[6], and that the definition of a citrus fruit tells us that it is a certain kind of fruit. By the transitivity of hypothetical syllogism[7], we then can safely assume that, if an orange is a citrus fruit, and a citrus fruit is a fruit, then an orange is also a fruit. As the final step of formalisation, the logically re-analysed entries are converted to a

- Representation in predicate logic:

$$\begin{aligned}&\forall x[\mathit{orange.2}(x) \rightarrow\\ &\quad[\mathit{round.1.1}(x)\\ &\quad\wedge\ \mathit{orange.1}(x)\\ &\quad\wedge\ \mathit{fruit.1}(x)\\ &\quad\wedge\ \mathit{juicy.1}(x)\\ &\quad\wedge\ \mathit{sweet.1}(x)]]\end{aligned}$$

At this point, let us digress a little. First, recall that by the conditional and distributive laws, it holds that

$$\begin{aligned}&(\mathit{orange.2}(x) \rightarrow\\ &(\mathit{round.1.1}(x) \wedge \mathit{orange.1}(x) \wedge \mathit{fruit.1}(x) \wedge \mathit{juicy.1}(x) \wedge \mathit{sweet.1}(x)))\\ &\Longleftrightarrow\\ &((\mathit{orange.2}(x) \rightarrow \mathit{round.1.1}(x)) \wedge\\ &(\mathit{orange.2}(x) \rightarrow \mathit{orange.1}(x)) \wedge\\ &(\mathit{orange.2}(x) \rightarrow \mathit{fruit.1}(x)) \wedge\\ &(\mathit{orange.2}(x) \rightarrow \mathit{juicy.1}(x)) \wedge\\ &(\mathit{orange.2}(x) \rightarrow \mathit{sweet.1}(x)))\end{aligned}$$

Now, if we think of the above predicates in terms of typed feature structures, where an expression like $\mathit{orange.2}(x)$ roughly corresponds to an—at this moment, purely hypothetical—type

[6] This information is implicitly contained in Cobuild in that the dictionary cites an orange as an instance of a citrus fruit.

[7] $((p \rightarrow q) \wedge (q \rightarrow r)) \Rightarrow (p \rightarrow r)$

$$orange.2\left[\ \right]$$

we could make a few assumptions about the informational content of such types. Take e.g. the types *orange.2* and *fruit.1*: since **fruit** serves as a superordinate of **orange**, the information contained in the former type is more specific than the information denoted by the latter one, i.e. the first type extends the second. This can be expressed by the statement

$$orange.2\left[\ \right] \sqsupseteq fruit.1\left[\ \right]$$

where the binary extension relation '$\sqsupseteq$' is the dual of the subsumption relation '$\sqsubseteq$', which will normally be employed. If we replace the predicate expressions and the logical connectives in the conditional statement

$$\begin{aligned}&((orange.2(x) \rightarrow round.1.1(x)) \wedge\\ &(orange.2(x) \rightarrow orange.1(x)) \wedge\\ &(orange.2(x) \rightarrow fruit.1(x)) \wedge\\ &(orange.2(x) \rightarrow juicy.1(x)) \wedge\\ &(orange.2(x) \rightarrow sweet.1(x)))\end{aligned}$$

by the corresponding counterparts in the feature structure notation, we get

$$\begin{aligned}orange.2\left[\ \right] &\sqsupseteq round.1.1\left[\ \right]\\ orange.2\left[\ \right] &\sqsupseteq orange.1\left[\ \right]\\ orange.2\left[\ \right] &\sqsupseteq fruit.1\left[\ \right]\\ orange.2\left[\ \right] &\sqsupseteq juicy.1\left[\ \right]\\ orange.2\left[\ \right] &\sqsupseteq sweet.1\left[\ \right]\end{aligned}$$

What is expressed here, is that the identification of 'being an orange' comprises the information of 'being round', of 'being orange', of 'being a fruit', and so on. Of course, it would be nonsensical to claim that these types carry information other than semantic, for we would get informational clashes or unification failures if we matched "substantival" and e.g. "adjectival" types.

Now, the binary extension relation, or subsumption relation for that matter, is a weak partial ordering on feature structures, i.e. it has the properties of being reflexive, antisymmetric, and—this is essential here—of being transitive as well. Taking up our 'citrus fruit' example again, we could establish a type hierarchy by these properties of the extension relation:

$$orange.2\left[\ \right] \sqsupseteq citrus\_fruit.1\left[\ \right] \sqsupseteq fruit.1\left[\ \right]$$

i.e. the first type inherits all the information contained in the second and third types. But by stating that the type *orange.2* is a subtype of five supertypes we have of course a case of multiple inheritance, and this is a formal device that is of course supported by HPSG (not! by the ALEP system). End of digression.

In the remainder of ths chapter we will outline our approach of mapping this logically structured dictionary information to HPSG.

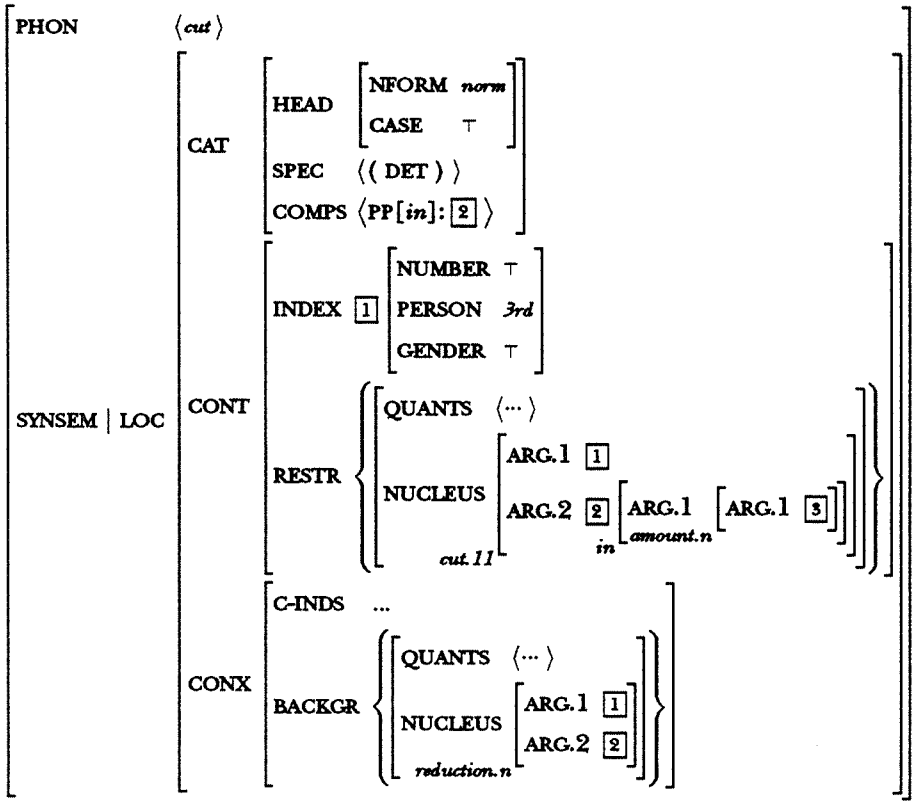
## 4.2 The structure of HPSG lexical entries

### 4.2.1 Standard representations for subtypes of "word"

In this section, we will briefly sketch a few basis assumptions underlying lexical entries in HPSG, and will point out several modifications suggested by the present approach which will then be explained in the following sections. Let us begin with the Cobuild definition for one of the senses of **cut**:

**11** COUNT N
A **cut** in an amount is a reduction in it.

Our logical analysis yields the following attribute-value matrix (henceforth "AVM") in a slightly modified HPSG format.




Given the complexity of the structure, a few general remarks on notation are appropriate here. The angle brackets in the value range of the SPEC and COMPS attributes denote list values, i.e. ordered $n$-tuples that can of course be of length one, as is the case above, or even zero. Unlike in other unification-based formalisms, like Functional Unification Grammar (FUG) or PATR, braces do not denote disjunctions but set values, over which the regular set-theoretic operations like union are defined. Tags like 1 express token- not type-identity of the so annotated substructures. Path information is given by the vertical slash, ' | ', as in SYNSEM | LOC, where the attribute to the left of the symbol is taken to expand to at least one further value which is not relevant in the specific context. Typed feature structures are denoted by AVMs with a left subscript in italics giving the type name. Names of attributes are capitalised, names of atomic values, which coincide with atomic sorts, are given in lower case italics. (DET) means that, syntactically, this argument does not always need to be realised, and hence is optional. In general, parentheses enclosing items in lists denote the optionality of these elements, i.e. they have to be interpreted as, given $n$ optional components, denoting a disjunction of $2^n$ lists. Thus we generally have

$$\langle \varphi, (\psi) \rangle \iff (\langle \varphi \rangle \vee \langle \varphi, \psi \rangle)$$

and in our present case, we arrive at the following equivalence:

$$\langle\,(\text{DET})\,\rangle \iff (\langle\,\text{DET}\,\rangle \vee \langle\ \ \rangle)$$

Also unlike in other formalisms, there are no colons as separators between attributes and their values, except in rare and well-defined cases (Cp. below). Note also that AVMs are not identical with feature structures, but only denote them, i.e. they are feature structure descriptions.

Now, how do we "read" this AVM? SYNSEM supplies syntactic as well as semantic information. It takes complex feature structures of the types *local* and *nonlocal* as values. The attribute NONLOCAL is not of concern here, it is employed for giving binding information in, for example, constructions containing reflexive pronouns. Following the path ... | CAT | ...[8] we get syntactic information, as can be seen by looking at the HEAD information: here, a type *noun* is identified by specification of the CASE and NFORM features, the latter of which could also be specified as *it* or *there*—this would be the case in the respective expletives. The list-valued attribute COMP[9] defines the subcategorisation list of the so specified signs; in the case of nouns, this list is usually empty. With transitive and ditransitive verbs we find the complements they take here. Note that these are syntactic arguments then, whose semantic content is co-indexed with the semantic argument positions of the verbs.

With respect to the attributes that matter most in the semantic description, RESTR and CONX, the present approach will diverge from the standard HPSG notions. The motivation for the mapping of dictionary information to CONX will be given the next section.

Let us turn to verbs now, and consider the entry for **adore 4**:

**4** VB WITH OBJ
If you **absorb** information, you learn and understand it.

$$\left[\begin{array}{ll}
\text{PHON} & \langle \textit{absorb} \rangle \\
\text{SYNSEM} \mid \text{LOC} & \left[\begin{array}{ll}
\text{CAT} & \left[\begin{array}{ll}
\text{HEAD} & \left[\begin{array}{ll}\text{VFORM} & \top\\ \text{AUX} & -\\ \text{INV} & \top\end{array}\right]\\
\text{SUBJ} & \langle \text{NP}[\textit{nom}]\cdot \boxed{3} \rangle\\
\text{COMPS} & \langle \text{NP}[\textit{acc}]\cdot \boxed{4} \rangle
\end{array}\right]\\
\text{CONT} & \left[\begin{array}{ll}
\text{QUANTS} & \langle \cdots \rangle\\
\text{NUCLEUS} & \left[\begin{array}{ll}\text{ARG.1} & \boxed{3}\ \underset{\textit{you}}{\left[\text{ARG.1}\ \boxed{1}\right]}\\ \text{ARG.2} & \boxed{4}\ \underset{\textit{information.n}}{\left[\text{ARG.1}\ \boxed{2}\right]}\end{array}\right]_{\textit{absorb.4}}
\end{array}\right]\\
\text{CONX} & \left[\begin{array}{ll}
\text{C-INDS} & \ldots\\
\text{BACKGR} & \left\{\left[\begin{array}{ll}\text{QUANTS} & \langle \cdots \rangle\\ \text{NUCLEUS} & \left[\begin{array}{l}\text{ARG.1}\ \boxed{3}\\ \text{ARG.2}\ \boxed{4}\end{array}\right]\\ \textit{learn.n} \wedge \textit{understand.n} & \end{array}\right]\right\}
\end{array}\right]
\end{array}\right]
\end{array}\right]$$

In standard HPSG treatments of verbs, we normally find $\text{NP}[\textit{nom}]\boxed{1}$ instead of our $\text{NP}[\textit{nom}]\cdot\boxed{3}$ in the subcategorisation lists (the values of SUBJ and COMPS). The former expression denotes a

[8] This is the position where the grammatical information given in the Cobuild definitions will be encoded.
[9] See Pollard & Sag (1993) for a discussion of this attribute in opposition to SUBCAT.

SYNSEM | LOC structure with a path leading to INDEX information, the latter we introduced in order to indicate a SYNSEM | LOC structure with a path leading to NUCLEUS information instead. INDEX information in HPSG theory is restricted to the use in the representation of semantic nominal information, as stated in Pollard & Sag (1993, Chapter 1)[10]:

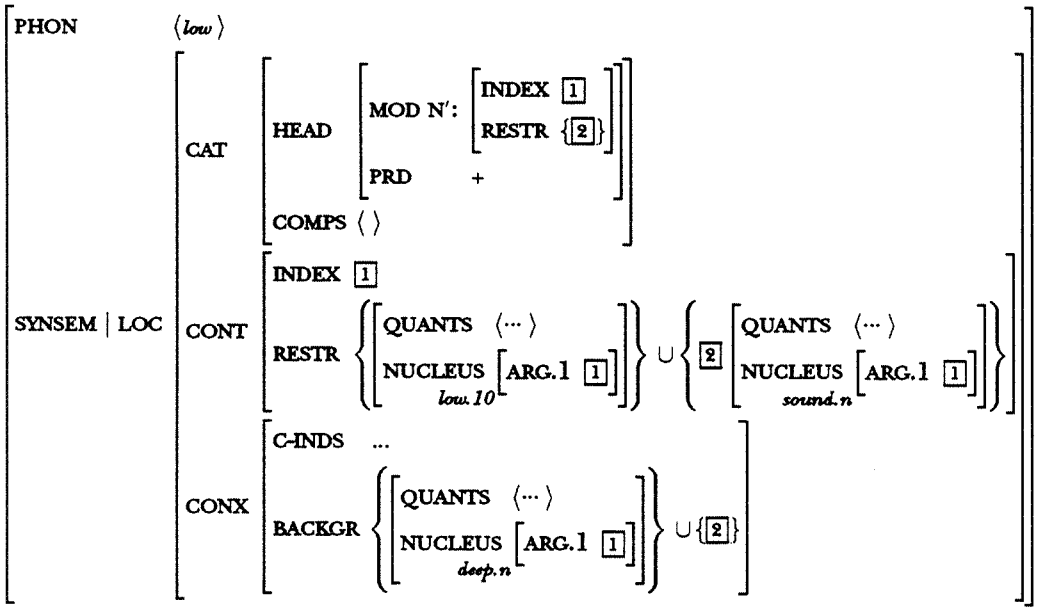
> For the CONTENT value of nominals (i.e. lexical nouns and their phrasal projections), we will employ a feature structure sort called *nominal object* (*nom-obj*), which in turn bears an attribute called INDEX (IND). The INDEX value, a structure of sort *index*, should be thought of as the HPSG analog of a reference marker in Discourse Representation Theory [...] or of a parameter introduced by an NP use in situation semantics. As we will soon see, it is indices to which semantic (or thematic) roles are assigned.

Our notation, then implies a shift in the encoding of semantic roles which will be explained in the followind sections. It should already be noted that the above abbreviation is not to be confused with expressions like XP[...]: 2, where the colon denotes a path to the immediate value range of CONTENT.

Let us finally look at the definition of **low 10**

**10** ADJ
A **low** sound is deep.

and its representation in HPSG:



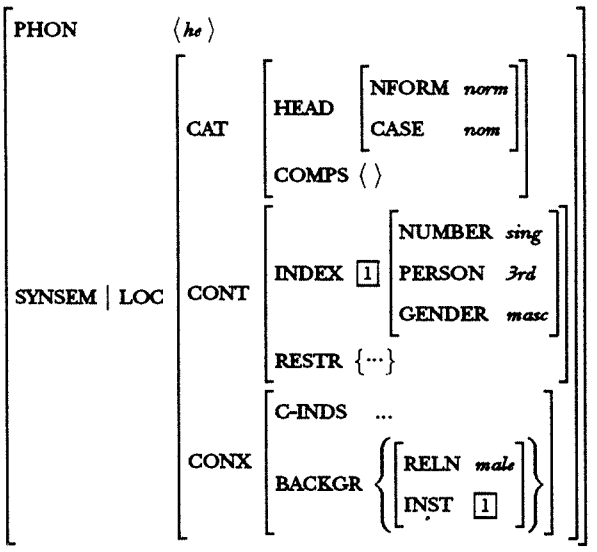

Note the syntactic cross-selection of the modified noun with a path leading to its immediate CONT value, expressed by "MOD': " and that the operator for set union "∪" denotes the coinciding properties of *low* and *sound* in one instance, which explains the differentiation in the meaning of **low** when used with **sound**. This will become clearer in Section 4.3.1.3.

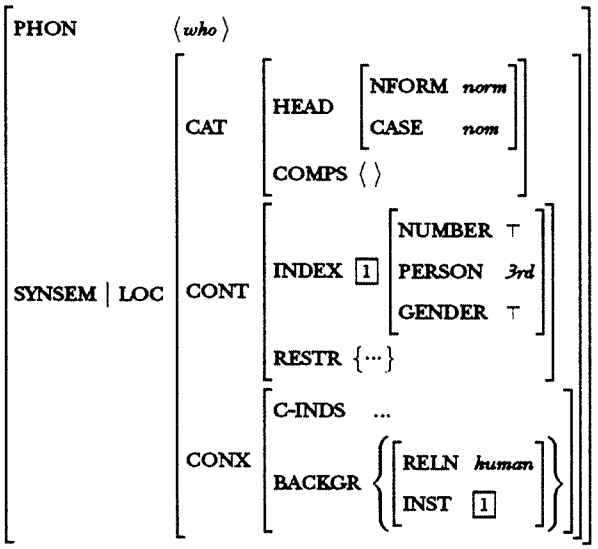
### 4.2.2 Representation of "contextual" information

In this section, we will claim that while CONTEXT information has not been treated in sufficient detail in standard HPSG yet, the RHSs of Cobuild definitions supply extremely rich information for precisely this feature. We will try to support this claim by a brief sketch of our HPSG treatment

[10] Unfortunately, the page numbering is not consistent in the manuscript. Hence just the chapter number here and in the following sections.

of common nouns under the aspect of pronominal reference and several collective nouns in connection with aggregate and non-aggregate determiners. Consider for example the standard HPSG entries for "he" and "who" below:




Oversimplifying, one might say that the pronoun "he" actually doesn't mean anything on its own. Only when you refer to someone by it does it get a certain meaning. That is why the CONT | RESTR range is empty in the HPSG entry. On the other hand, though, you can only refer to someone by "he" when this person is male—and this boils down to *male*(*x*)—or you'll deviate from normal usage. This is expressed by the CONX | BACKGR value, which states that the normal usage context or utterance situation always requires a male referent for "he". Analogously, the same holds for "who", where normal usage requires a human referent. So far so good, but how can the entry for **boy** 'tell' that the referent of this expression is perfectly suitable for being referred to by both "he" and "who". It lies at hand that the relevant information in **boy** should also be stored in the CONX value range.

Normally, though, HPSG treatments show little interest as regards the CONX value range of regular nouns, which is hence mostly mentioned only implicitly or ignored altogether. Sadly so, since it also constitutes the much asked-for interface to world knowledge. Apart from that, the CONX value range states a lot of 'linguistically relevant' information. And this is, where a dictionary like

Cobuild comes in very handy. As pointed out above, it states precisely what we want to have available for an NLP system, i.e., once again:

1 COUNT N
A **boy** is a male child.

This enables us to encode the predicates *male*(*x*) and *child*(*x*) in the CONX range of **boy**—just like before with *male*(*x*) and *human*(*x*) in the cases of "he" and "who", respectively.

$$\left[\begin{array}{ll} \text{PHON} & \langle \textit{boy} \rangle \\ \text{SYNSEM} \mid \text{LOC} & \left[\begin{array}{ll} \text{CAT} & \left[\begin{array}{ll} \text{HEAD} & \left[\begin{array}{ll} \text{NFORM} & \textit{norm} \\ \text{CASE} & \top \end{array}\right] \\ \text{SPEC} & \langle (\text{DET}) \rangle \\ \text{COMPS} & \langle\,\rangle \end{array}\right] \\ \text{CONT} & \left[\begin{array}{ll} \text{INDEX} \boxed{1} & \left[\begin{array}{ll} \text{NUMBER} & \top \\ \text{PERSON} & \textit{3rd} \\ \text{GENDER} & \textit{masc} \end{array}\right] \\ \text{RESTR} & \left\{\left[\begin{array}{ll} \text{QUANTS} & \langle \cdots \rangle \\ \text{NUCLEUS} & \underset{\textit{boy}}{\left[\text{ARG.1} \boxed{1}\right]} \end{array}\right]\right\} \end{array}\right] \\ \text{CONX} & \left[\begin{array}{ll} \text{C-INDS} & \ldots \\ \text{BACKGR} & \left\{\underset{\textit{male}}{\left[\text{ARG.1} \boxed{1}\right]}, \underset{\textit{child}}{\left[\text{ARG.1} \boxed{1}\right]}\right\} \end{array}\right] \end{array}\right] \end{array}\right]$$

Via a grammatical principle which checks the consistency of the CONX values and types the unification of e.g. *male* and *female* would result in a failure, $\perp$, or whatever you prefer. Encoding information of Cobuild in this fashion, however, makes additional revisions of the CONX complex necessary: the principle of contextual consistency will have to be formulated more restrictively.

It will only work, though, if the consistency of the types is mentioned explicitly elsewhere, which necessitates statements of type subsumption relations for each entry. In the case of **boy**, we need to state in feature terms (see also below) that

$$\textit{boy}(x) \rightarrow \textit{male}(x) \wedge \textit{child}(x)$$

or that the type *boy* is an extension of both the types *child* and *male*.

Consider the Cobuild entry for **child**: "A **child** is a human being ...". We get an LHS (or CONT)—i.e. *child*(*x*)—which implies the RHS (or *conx*)—*human*(*x*)—resulting in the corresponding type subsumption statement. This guarantees that in the case of "who" and **boy** their respective CONX values (*human* and *child*) unify. Once tied to a "boyish" referent, "who" now carries the CONT semantic information of **boy**, so to speak. The procedure with "he" works analogously.

The question now is: why at all encode this type of information in the CONX value range, if a type subsumption statement is required in any case? Consider the definition below:

1 ADJ
**Fierce** means very aggressive or angry.

Whereas in the case of **boy** it is reasonable to assume that it inherits information both from **male** and **child**, it is certainly non-sensical to claim that **fierce** inherits information from both **very** and

**aggressive** among others.[11] **Fierce** does, however, place a very specific restriction on its supertype that cannot be expressed in a hierarchy. Storing this information in the BACKGROUND instead has the nice effect to predict that "slightly fierce" sounds slightly odd.

Next consider Pollard & Sag's (1993:2.4.3) treatment of aggregate versus non-aggregate determiners. We repeat their data here:

(...)
a. Every faculty is/*are homogenous.
b. Every faculty meets/*meet on a monthly basis.
c. All faculty *is/are required to submit midterm grades by March 12.
d. All faculties *meets/meet on a monthly basis.

Their analysis of the non-aggregate determiner **every** and the aggregate determiner **all** predicts the data above. They further claim in a footnote that the set of nouns showing the same behaviour as **faculty** includes **staff**, **clergy**, and **nobility**, and that nouns like **family**, **committee**, and **government** "disallow aggregate quantification with the singular forms." (2.4.3) Now compare the following entries from Cobuild:

**1** COLL N OR UNCOUNT N
A **family** is a group of people who are related to each other, especially parents and their children.

**1** COLL N
A **committee** is a group of people who represent a larger group or organization and make decisions for them.

**1** COLL N
A **government** is the group of people who are responsible for governing a country or state.

**1** COLL N
The **staff** of an organization are the people who work for it.

**1** PLURAL N
The **clergy** are the religious leaders of a Christian church.

**2** COLL N
The **nobility** of a society are all the people who have titles and high social rank.

The correspondence is striking. In all "faculty" cases the definitions contain the superordinate "a group of people", which in our analysis would hence be encoded as BACKGROUND information triggering precisely the same predictions as in Pollard & Sag (1993). BACKGROUND information corresponding to **people** would then disallow the reading in c.

### 4.2.3 Representation of hierarchical type information

It has been pointed out in the preceding sections that we intend to couch our logical analysis of the dictionary in the situation semantics module assumed in HPSG. It lies at hand that this strategy employs a translation of our disambiguated predicates into complex feature structures. For exemplification, consider the following standard HPSG AVMs now:

$$\begin{bmatrix} \text{RELN} & \textit{orange} \\ \text{INST} & \boxed{1} \end{bmatrix} \qquad \begin{bmatrix} \text{RELN} & \textit{pink} \\ \text{ARG} & \boxed{1} \end{bmatrix} \qquad \begin{bmatrix} \text{RELN} & \textit{walk} \\ \text{WALKER} & \boxed{1} \end{bmatrix}$$

$$\begin{bmatrix} \text{RELN} & \textit{love} \\ \text{LOVER} & \boxed{1} \\ \text{LOVED} & \boxed{2} \end{bmatrix} \qquad \begin{bmatrix} \text{RELN} & \textit{give} \\ \text{GIVER} & \boxed{1} \\ \text{GIVEN} & \boxed{2} \\ \text{GIVEN-TO} & \boxed{3} \end{bmatrix}$$

[11] The disjunction is, admittedly, a totally different story. Strictly speaking, one could argue that this necessitates two separate entries, unless one finds an appropriate treatment of disjunctive compound infons.

There are five different $n$-place relations, which obviously correlate with predicates in the logical sense. Each of them also introduces differently labelled argument positions as further attributes which correspond to rather specific semantic roles. As a first step of the present re-interpretation, the argument positions are generalised to ARG.1 to ARG.$n$

$$\begin{bmatrix} \text{RELN} & \textit{orange} \\ \text{ARG.1} & \boxed{x} \end{bmatrix} \qquad \begin{bmatrix} \text{RELN} & \textit{pink} \\ \text{ARG.1} & \boxed{x} \end{bmatrix} \qquad \begin{bmatrix} \text{RELN} & \textit{walk} \\ \text{ARG.1} & \boxed{x} \end{bmatrix} \qquad \begin{bmatrix} \text{RELN} & \textit{love} \\ \text{ARG.1} & \boxed{x} \\ \text{ARG.2} & \boxed{y} \end{bmatrix} \qquad \begin{bmatrix} \text{RELN} & \textit{give} \\ \text{ARG.1} & \boxed{x} \\ \text{ARG.2} & \boxed{y} \\ \text{ARG.3} & \boxed{z} \end{bmatrix}$$

since we can assume that the argument positions of the disambiguated predicates we employ are identified precisely enough to allow avoiding the further specification of semantic roles like the ones introduced in the original HPSG feature structures. This, however, goes along with a different analysis of the value ranges of ARG.$x$. Take, for example, the lexical entry for **snore**, where in a standard treatment one would assume a semantic role like SNORER or perhaps a more general role like AGENT or EXPERIENCER. Consider now the following Cobuild definition:

1 VB
When someone who is asleep **snores**, they make a loud noise each time they breathe.

Given our analysis, we must now provide a representational means to express the specific type of arguments the relation *snore* can take. The dictionary defines this by placing *someone* in the subject position of the verb. Recall that *someone* is to be treated as a term of the metalanguage of dictionary definitions and carries much more meaning than the corresponding pronoun (Cp. Section 1). Hence we are justified in treating this expression as another predicate, or rather, relation in the situation theoretic sense. We express this by defining an infon with the one-place relation *s-o*:

$$\ll \text{s-o}, \dot{x}, 1 \gg$$

In concordance with standard HPSG approaches, we have assumed throughout this report that following correspondence holds

$$pred(a_1, \ldots, a_n) \quad \approx \quad \ll \text{reln}, a_1, \ldots, a_n, 1 \gg$$

and have adopted the notational means below

$$\ll \text{reln}, a_1, \ldots, a_n, 1 \gg \quad \approx \quad \begin{bmatrix} \text{RELN} & \textit{reln} \\ \text{ARG.1} & \boxed{a_1} \\ \vdots & \vdots \\ \text{ARG.}n & \boxed{a_n} \end{bmatrix}$$

Provided these mappings we can now state that **snore** places a condition on the parameter of its argument that corresponds to the type *s-o*:

$$\begin{bmatrix} \text{RELN} & \textit{snore.1} \\ \text{ARG.1} & \boxed{2} \begin{bmatrix} \text{RELN} & \textit{s-o} \\ \text{ARG.1} & \boxed{1} \end{bmatrix} \end{bmatrix}$$

Additionally allowing conjunctive compound infons, one could even further restrict the argument role and follow the dictionary by claiming

$$\begin{bmatrix} \text{RELN} & \textit{snore.1} \\ \text{ARG.1} & \boxed{2}\begin{bmatrix} \text{RELN} & \textit{s-o} \wedge \textit{asleep.1} \\ \text{ARG.1} & \boxed{1} \end{bmatrix} \end{bmatrix}$$

The present approach naturally involves the formulation of a hierarchy of the above semantic types. Consider these AVMs in Pollard & Sag (1993:8.5.1)

$$\textit{psoa}\begin{bmatrix} \text{RELATION} & \textit{love} \\ \text{LOVER} & \textit{ref}\begin{bmatrix} \end{bmatrix} \\ \text{LOVED} & \textit{ref}\begin{bmatrix} \end{bmatrix} \end{bmatrix} \qquad \textit{psoa}\begin{bmatrix} \text{RELATION} & \textit{love} \\ \text{INFLUENCE} & \textit{ref}\begin{bmatrix} \end{bmatrix} \\ \text{INFLUENCED} & \textit{ref}\begin{bmatrix} \end{bmatrix} \\ \text{SOA-ARG} & \begin{bmatrix} \end{bmatrix} \end{bmatrix}$$

The problem with these feature structures is that they are not totally well-typed, i.e. it is not the case that they bear all the attributes introduced by a specific type and only those. This is a restriction on typed feature logics that is commonly assumed. Pollard & Sag (1993:8.5.1) then, suggest to replace the above AVMs by matrices in which the function of the atomic values in the RELN range is now performed by corresponding sorts, as for example, in:

$$\textit{love}\begin{bmatrix} \text{LOVER} & \textit{ref}\begin{bmatrix} \end{bmatrix} \\ \text{LOVED} & \textit{ref}\begin{bmatrix} \end{bmatrix} \end{bmatrix} \qquad \textit{persuade}\begin{bmatrix} \text{INFLUENCE} & \textit{ref}\begin{bmatrix} \end{bmatrix} \\ \text{INFLUENCED} & \textit{ref}\begin{bmatrix} \end{bmatrix} \\ \text{SOA-ARG} & \begin{bmatrix} \end{bmatrix} \end{bmatrix}$$

We follow this approach and adopt the convention expressed below:

$$\begin{bmatrix} \text{RELN} & \textit{reln} \\ \text{ARG.1} & \boxed{a_1} \\ \vdots & \vdots \\ \text{ARG.}n & \boxed{a_n} \end{bmatrix} \approx \textit{reln}\begin{bmatrix} \text{ARG.1} & \boxed{a_1} \\ \vdots & \vdots \\ \text{ARG.}n & \boxed{a_n} \end{bmatrix}$$

Pollard & Sag (1993:8.5.1), however, additionally assume that the relation between the above atomic sorts is captured by a sortal hierarchy where the following subsumption relations hold:

$$\begin{array}{ccccc} \textit{psoa} & \sqsubseteq & \textit{influence} & \sqsubseteq & \textit{persuade} \\ \textit{psoa} & \sqsubseteq & \textit{feeling} & \sqsubseteq & \textit{love} \end{array}$$

The necessity for this constellation arose from their analysis of control verbs, which we will not repeat here. It is crucial for our approach, though, that the formalism allows that types do not introduce any additional features locally, as implied by the analysis in Pollard & Sag (1993:8.5.1).[12] Here, obviously, the type *persuades* does not introduce any features locally, since both INFLUENCER and INFLUENCED are inherited from *influence*.

Let us now recapitulate the strategy for the conversion of our predicates to typed feature structures. The correlation, exemplified by the NUCLEUS value of **orange**, is the following:

[12] Incidentally, this is permissible in ALEP, too.

**First order:** $orange.2(x)$

**Infonic:** $\ll$orange.2, $\dot{x}$, 1$\gg$

**AVM:**

$$\begin{bmatrix} \text{RELN} & orange.2 \\ \text{ARG.1} & \boxed{x} \end{bmatrix}$$

**AVM:**

$$orange.2\left[\text{ARG.1}\ \boxed{x}\right]$$

Provided the predicate logic analysis in Section 4.1,

$$\begin{aligned}&((orange.2(x) \rightarrow round.1.1(x)) \wedge\\ &(orange.2(x) \rightarrow orange.1(x)) \wedge\\ &(orange.2(x) \rightarrow fruit.1(x)) \wedge\\ &(orange.2(x) \rightarrow juicy.1(x)) \wedge\\ &(orange.2(x) \rightarrow sweet.1(x)))\end{aligned}$$

we can now represent the information on **orange** as types in a hierarchy with multiple inheritance:

$$orange.2\left[\text{ARG.1}\ \boxed{x}\right] \sqsupseteq round1.1\left[\text{ARG.1}\ \boxed{x}\right]$$

$$orange.2\left[\text{ARG.1}\ \boxed{x}\right] \sqsupseteq orange.1\left[\text{ARG.1}\ \boxed{x}\right]$$

$$orange.2\left[\text{ARG.1}\ \boxed{x}\right] \sqsupseteq fruit.1\left[\text{ARG.1}\ \boxed{x}\right]$$

$$orange.2\left[\text{ARG.1}\ \boxed{x}\right] \sqsupseteq juicy.1\left[\text{ARG.1}\ \boxed{x}\right]$$

$$orange.2\left[\text{ARG.1}\ \boxed{x}\right] \sqsupseteq sweet.1\left[\text{ARG.1}\ \boxed{x}\right]$$

where " $\sqsupseteq$ " denotes the dual of the subsumption relation " $\sqsubseteq$ ". Hence we assume the following to hold for the lexicon encoded in typed (complex) feature structures:

- it is a weak partial order under the subsumption relation
- it forms a semi-lattice with $n$ maximal elements, but no greatest
- it forms a hierarchy of semantic types

### 4.2.4 Representation of selectional preferences

As has also been pointed out by the Birmingham and Pisa teams in the preceding sections, selectional preferences are expressed in plain English in the LHS of Cobuild definitions. Consider, for example, the following entries:

1 VB
If a king or queen **abdicates**, he or she resigns.

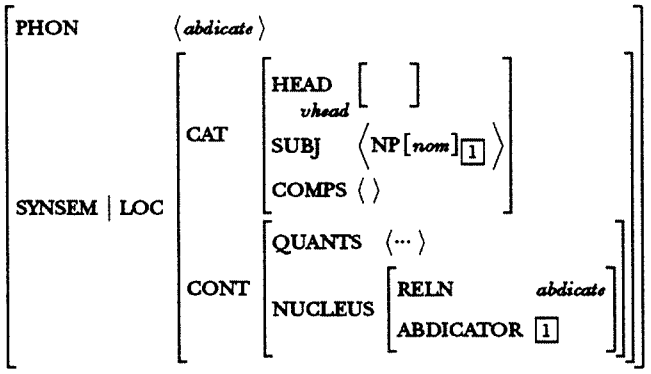
1 VB
When a donkey **brays**, it makes the loud, harsh sound that donkeys make.

1 VB WITH OBJ
When a woman **breast-feeds** her baby, she feeds it with milk from her breasts, rather than from a bottle.

2 VB WITH OBJ
When you **butter** bread or toast, you spread butter on it.

1 VB
When a horse **gallops**, it runs very fast so that all four legs are off the ground at the same time.

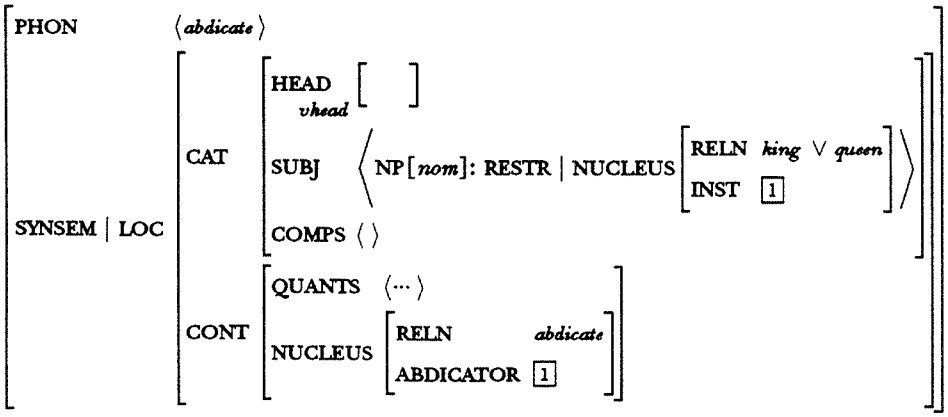
Preferences have so far not been treated in the HPSG framework. The practically only means the theory provides for covering issues like these are the subcategorisation principle and semantic roles. The AVM below depicts the standard HPSG lexical entry for **abdicate**:




With only fairly specific semantic roles like ABDICATOR and a subject NP with co-indexed INDEX information, there are no explicitly formulated principles that predict other grammaticality judgements than the following:

(1)
a. The queen abdicated.
b. The general abdicated.
c. The donkey abdicated.
d. * The king abdicated the queen.
e. * The queen abdicated the general.
f. * The queen abdicated the donkey.

Sentences (1) d.–f. do not obey the subcategorisation principle, since the verb phrase contains an additional noun phrase not indicated on the COMPS list of the head daughter. Hence they are regarded as being ungrammatical in the same sense as the sentence *"Abdicated." lacking a subject noun phrase. Also, there is nowhere an explicit statement that prevents the filling of an ABDICATOR role by *donkey*. A tentative solution could be to encode the preference information on the SUBJ list of the lexical entry:



This strategy fits into a number of approaches within the wider HPSG framework (and also NLP), and hence is practicable. In the strict sense of the theory, however, it predicts the following judgments on the well-formedness of the above sentences:

(2)

a. The queen abdicated.
b. * The general abdicated.
c. * The donkey abdicated.
d. * The king abdicated the queen.
e. * The queen abdicated the general.
f. * The queen abdicated the donkey.

which is not the intended result. Ideally, the grammar should express a number of different aspects in this set of sentences. First, b. might be "ungrammatical" when, for example, uttered by a native speaker of German who simply assumes, that the same selectional preferences hold for ***abdanken*** and **abdicate**—in German, high-ranking officer can "abdicate", so to speak. The same sentence uttered by a native speaker of English will probably carry a slightly different meaning, i.e. it would insinuate that the general perceives himself as a royalty, or something similar. In c., an analogous reading would be difficult. Sentence d. is not ungrammatical, but could very well be meant ironic, if we, for example, anchor the king under discussion to Henry VIII. It would in this case mean that the queen "was abdicated" (and probably beheaded). This trick can be performed with a wide range of transitive verbs, though, and hence is not an idiosyncratic behaviour of **abdicate**. In e., the ironic reading of d. might be obtained if we think of a queen like Elisabeth I "abdicating" a prominent figure like Sir Francis Drake, but f. can only with difficulty be interpreted in this way.

This constellation might be expressed by a differentiation of the several well-formedness constraints involved in the data under discussion, perhaps very roughly along the following lines:

(3)

a. The queen abdicated.
b. $?_{cont}$ The general abdicated.
c. $??_{cont}$ The donkey abdicated.
d. $??_{cont/comps}$ The king abdicated the queen.
e. $??_{cont/comps}$ The queen abdicated the general.
f. $???_{cont/comps}$ The queen abdicated the donkey.

First, this is to be understood, that sentences b.–f. are not considered ungrammatical in the absolute sense of being hardly understandable, but as in some way deviating from "normal" usage. Admittedly, the HPSG framework is currently not designed to cover the various degrees of severity of lexical selection ranging from mere preferences to restrictions hardly to be overruled. For instance, you can of course say without any problem that somebody **butters** a croissant instead of bread or toast as indicated in the dictionary, but in order to find a context where someone correctly states that a woman **breastfeeds** somebody or something other than a baby, it takes a very moving scene in Steinbeck's *Grapes of Wrath*. We can, however, distinguish in HPSG between different types of information, as indicated by *cont* and *comps* in the above list. Consider the AVM below, where these expressions correspond to the features CONT and COMPS:

$$\left[\begin{array}{ll} \text{PHON} & \langle \textit{abdicate} \rangle \\ \text{SYNSEM} \mid \text{LOC} & \left[\begin{array}{ll} \text{CAT} & \left[\begin{array}{l} \text{HEAD}\ \underset{\textit{vhead}}{\left[\ \right]} \\ \text{SUBJ}\ \langle \text{NP}[\textit{nom}]\boxed{1} \rangle \\ \text{COMPS}\ \langle\ \rangle \end{array}\right] \\ \text{CONT} & \left[\begin{array}{ll} \text{QUANTS} & \langle \cdots \rangle \\ \text{NUCLEUS} & \left[\begin{array}{ll} \text{RELN} & \textit{abdicate} \\ \text{ARG.1} & \boxed{2}\left[\begin{array}{ll} \text{RELN} & \textit{king} \vee \textit{queen} \\ \text{ARG.1} & \boxed{1} \end{array}\right] \end{array}\right] \end{array}\right] \end{array}\right] \end{array}\right]$$

This notation expresses the view that selectional preferences constitute an inherent part of the meaning of the word. In the case of **abdicate** it is the relation *abdicate* itself which determines the restriction on the parameter in the value range of its single argument. In terms of situation semantics we would express this fact by the infon

$$\ll \text{abdicate.1}, \dot{x}, 1 \gg$$

where $\dot{x}$ denotes a restricted parameter for which the following condition holds

$$\dot{x}_1 = \mathit{IND}_1 \restriction \ll \text{king.1}, \mathit{IND}_1, 1 \gg \vee \ll \text{queen.1}, \mathit{IND}_1, 1 \gg$$

which, intuitively, means that the restricted parameter $\dot{x}$ filling the argument role of *abdicate* can only be anchored to a certain individual (a parameter of the type *IND*) if this individual meets the condition introduced by " $\restriction$ " namely that this individual must either be a king or a queen. This condition is denoted by the disjunctive compound infon above. This strategy, however, necessitates a complex re-formulation of the semantics principle since only the INDEX information is co-indexed on the COMPS list. A perhaps easier compromise is illustrated by the AVM below:

$$\left[\begin{array}{ll} \text{PHON} & \langle \textit{abdicate} \rangle \\ \text{SYNSEM} \mid \text{LOC} & \left[\begin{array}{ll} \text{CAT} & \left[\begin{array}{l} \text{HEAD}\ \left[\begin{array}{ll} \text{VFORM} & \top \\ \text{AUX} & - \\ \text{INV} & \top \end{array}\right] \\ \text{SUBJ}\ \langle \text{NP}[\textit{nom}] \cdot \boxed{2} \rangle \\ \text{COMPS}\ \langle\ \rangle \end{array}\right] \\ \text{CONT} & \left[\begin{array}{ll} \text{QUANTS} & \langle \cdots \rangle \\ \text{NUCLEUS} & \underset{\textit{abdicates}}{\left[\text{ARG.1}\ \underset{\textit{king} \vee \textit{queen}}{\boxed{2}}\left[\text{ARG.1}\ \boxed{1}\right]\right]} \end{array}\right] \end{array}\right] \end{array}\right]$$

where RELN has been replaced by the corresponding types, and where the substructure

$$\underset{\textit{abdicate}}{\left[\text{ARG.1}\ \underset{\textit{king} \vee \textit{queen}}{\boxed{2}}\left[\text{ARG.1}\ \boxed{1}\right]\right]}$$

expands to

$$\textit{abdicate}\left[\begin{array}{l}\text{ARG.1}\ \boxed{2} \\ \left(\begin{array}{l}\textit{king}\left[\text{ARG.1}\ \boxed{1}\right] \\ \vee \\ \textit{queen}\left[\text{ARG.1}\ \boxed{1}\right]\end{array}\right)\end{array}\right]$$

Recall that NP[*nom*]· 2 abbreviates a path leading to NUCLEUS information. The re-formulation of the subcategorisation principle (and Immediate Dominance Schemata) to a set of bi-partite constraints, one co-indexing INDEX information, the other NUCLEUS information, could be a tractable strategy for the needed differentiation, provided analyses like the above.

It must be noted here that the preceding AVM is incomplete—it lacks the necessary specification of the CONX value range. The BACKGROUND information within this range has been motivated before, but the present approach also necessitates the incorporation of contextual indices in the semantic information of phrases. Consider the sentence "The man abdicated." which is perfectly acceptable in a context like "Edward VIII had insurmountable social problems concerning his wife. The man (Edward VIII) abdicated." Now, provided our treatment of hierarchical semantic types, *Edward VIII* would extend *king* and *king* would subsume *man*, hence we get a correct interpretation of our above sentence. The same sentence uttered in the context "The researcher had insurmountable scientific problems with his report. The man abdicated." is very odd unless the researcher happens to be a king. This fact can only be captured in a description, if it is possible to identify the contextual indices and check the type of the individual under discussion.

## 4.3 Automatic derivation of HPSG entries from Cobuild

### 4.3.1 The mapping strategy

We distinguish between two levels at which the Birmingham output is mapped to HPSG format

- the outer level
- the inner level

both of which concern the essential technical aspects of the conversion procedures employed by the Bochum "d21" program. Apart from the specific theoretical assumptions underlying this algorithm, the levels reflect more general tactics of breaking up the informational components of the dictionary definitions. For exemplification, let us now consider the Cobuild entry for **permit**,

1 VB WITH OBJ
If you **permit** something, you allow it.

and the corresponding output of the Birmingham parser:

```
(    (def_number     19842)
     (sense          1)
     (def_type       1)
     (lemma          '(permit permits
                       permitting permitted ))
     (grammar        '(VB with OBJ))
     (pre
          (co-text0
               '()
          )
     )
     (op-word
          (hinge
               '(if)
          )
     )
     (lhs-1
          (co-text1
            (match1
               '(you)
            )
          )
          (head1
               '(permit)
          )
          (co-text2
            (match2
               '(something)
               '(,)
            )
          )
     )
     (rhs-2
          (match1
               '(you)
          )
          (synonym
               '(allow)
          )
          (match2
               '(it)
          )
     )
     (post
          (note '(a formal use.)
          )
     )
)
```

The general template for the information content of the "outer level" analysis and representation comprises the storage of the Cobuild `grammar`, `lhs-1`, and `rhs-2` information in the value ranges of the HPSG features CAT, CONT, and CONX, respectively (See Section 4.2 on the theoretical basis for this correlation). In concordance with common HPSG practice, the PHON value range is specified by `lemma` information. A full-fledged HPSG analysis would of course entail the encoding of the phonetic information readily available in Cobuild (cp. Section 4.4 on the technical implications). Filling the general template below

$$\begin{bmatrix} \text{PHON} & \langle \textit{"lemma/head1"} \rangle \\ \text{SYNSEM} \mid \text{LOC} & \begin{bmatrix} \text{CAT} & \textit{"grammar"} \\ \text{CONT} & \textit{"lhs-1"} \\ \text{CONX} & \textit{"rhs-2"} \end{bmatrix} \end{bmatrix}$$

with specific information would, in the case of **permit**, yield the following pseudo-entry:

$$\begin{bmatrix} \text{PHON} & \langle \textit{permit} \rangle \\ \text{SYNSEM} \mid \text{LOC} & \begin{bmatrix} \text{CAT} & \textit{"VB with OBJ"} \\ \text{CONT} & \textit{you permit something} \\ \text{CONX} & \textit{you allow it} \end{bmatrix} \end{bmatrix}$$

In order to build the remaining information gradually up to more complex feature structures, we now have to proceed not only by further translating the meta expressions of the Birmingham team but also by inserting several feature complexes by default. Consider the general "inner level" template for transitive verbs below:

$$\left[\begin{array}{ll}
\text{PHON} & \langle \textit{"lemma/head1"} \rangle \\
\text{SYNSEM} \mid \text{LOC} & \left[\begin{array}{ll}
\text{CAT} & \left[\begin{array}{ll}
\text{HEAD} & \textit{"VB"} \\
\text{SUBJ} & \langle \text{NP}[\textit{nom}] \cdot \boxed{3} \rangle \\
\text{COMPS} & \langle \textit{"with OBJ"} \rangle
\end{array}\right] \\
\text{CONT} & \left[\begin{array}{ll}
\text{QUANTS} & \langle \cdots \rangle \\
\text{NUCLEUS} & \underset{\textit{"lemma/head1"} + . \textit{"sense"}}{\left[\begin{array}{ll}
\text{ARG.1} & \underset{\textit{"match1"}}{\boxed{3}}\left[\text{ARG.1}\ \boxed{1}\right] \\
\text{ARG.2} & \underset{\textit{"match2"}}{\boxed{4}}\left[\text{ARG.1}\ \boxed{2}\right]
\end{array}\right]}
\end{array}\right] \\
\text{CONX} & \left[\begin{array}{ll}
\text{C-INDS} & \ldots \\
\text{BACKGR} & \left\{\left[\begin{array}{ll}
\text{QUANTS} & \langle \cdots \rangle \\
\text{NUCLEUS} & \underset{\textit{"synonym"}}{\left[\begin{array}{ll}
\text{ARG.1} & \textit{"match1"} \\
\text{ARG.2} & \textit{"match2"}
\end{array}\right]}
\end{array}\right]\right\}
\end{array}\right]
\end{array}\right]
\end{array}\right]$$

In the value range of the feature CAT—which has not involved a complex value in the "outer level" description—the grammar information supplied by the Birmingham output is broken up into three different HPSG feature specifications: HEAD, SUBJ, and COMPS. The list value of SUBJ contains a default insertion of a defined HPSG abbreviation reflecting the assumption that transitive verbs lexically require a nominative NP subject syntactically realising the first semantic argument. The remaining feature paths are filled in a similar fashion.

Finally, by replacing the remaining variables in the general verb template we arrive at the HPSG-entry corresponding to the first sense of **permit**:

$$\left[\begin{array}{ll}
\text{PHON} & \langle \textit{permit} \rangle \\
\text{SYNSEM} \mid \text{LOC} & \left[\begin{array}{ll}
\text{CAT} & \left[\begin{array}{ll}
\text{HEAD} & \left[\begin{array}{ll}
\text{VFORM} & \textit{bse} \\
\text{AUX} & - \\
\text{INV} & -
\end{array}\right] \\
\text{SUBJ} & \langle \text{NP}[\textit{nom}] \cdot \boxed{3} \rangle \\
\text{COMPS} & \langle \text{XP}[\ldots] : \boxed{4} \rangle
\end{array}\right] \\
\text{CONT} & \left[\begin{array}{ll}
\text{QUANTS} & \langle \cdots \rangle \\
\text{NUCLEUS} & \underset{\textit{permit.1}}{\left[\begin{array}{ll}
\text{ARG.1} & \underset{\textit{you}}{\boxed{3}}\left[\text{ARG.1}\ \boxed{1}\right] \\
\text{ARG.2} & \underset{\textit{s-th}}{\boxed{4}}\left[\text{ARG.1}\ \boxed{2}\right]
\end{array}\right]}
\end{array}\right] \\
\text{COXT} \mid \text{BACKGR} & \left\{\underset{\textit{allow.n}}{\left[\begin{array}{ll}
\text{QUANTS} & \langle \cdots \rangle \\
\text{NUCLEUS} & \left[\begin{array}{ll}
\text{ARG.1} & \boxed{3} \\
\text{ARG.2} & \boxed{4}
\end{array}\right]
\end{array}\right]}\right\}
\end{array}\right]
\end{array}\right]$$

#### 4.3.1.1 Mapping of verbs

Let us start with a simple example of a Type 1 definition:

1 VB WITH OBJ
If you **acquire** something, you obtain it.

Recall that the (first-order predicate) logical form of this entry assumed in Section 4.1. requires the assignment of a two-place predicate to the above `head1`, which in the current example would yield

$$acquire(x, y)$$

In the automatic analysis, this is achieved by checking the values of `co-text1` and `co-text2`, whose constellation here clearly indicates the arity. In the case of verbs labelled 'VB WITH OBJ', the necessary information could in principle be derived from this grammar information, but definitions of control verbs like

1 VB WITH OBJ
If someone or something **persuades** you to do a particular thing, they cause you to do it by giving you a good reason for doing it.

carry the same grammar specification. Hence the simple mapping of this item to the transitive verb template would yield wrong results, at least in the case of equi verbs.

A further step in the transcription to HPSG employs disambiguation of the entries in order to identify a definite meaning, which is then indicated by the '.*n*' suffix to the predicate, i.e.

$$acquire.1(x, y)$$

in the present case. This step is trivial in the headword—it is of course not trivial in the remaining cases within the rest of the definition text—since the value of `sense` in the Birmingham output supplies the information directly. This will, however, only work in cases where the input for the Birmingham system is correct. Consider, for example, the definition of **take advantage of**:

• PHRASES
If you **take advantage of** someone, you treat them unfairly for your own benefit.

for which the Birmingham parser yields the following output:

```
(    (def_number     419)
     (sense          2)
     (def_type       1)
     (lemma          '(advantage advantages ))
     (grammar        '(PHRASE))
...
```

The sense number is obviously incorrect—it is identical with the number of the second meaning of the defined noun. We suspect that this is due to the fact that, in the dictionary, there are no sense numbers indicated for the PHRASES of a lemma, and that the output of the Birmingham parser consequently yields incorrect (and identical) sense numbers for all the PHRASES-senses. Moreover, this sense number seems to be in all the relevant cases the sense number of the last non-PHRASES-sense defined. In our present case, we have the following constellation:

i. **1** COUNT N An **advantage** is something that puts you in a better position ...

ii. **2** COUNT N An **advantage** is also a benefit that ...

iii. [**2**?] PHRASES • If something is **to** your **advantage**, ...

iv. [**2** ] PHRASES • If you **take advantage of** someone, ...

v. [2?] PHRASES • If you **take advantage of** something, ...

where digits in brackets indicate the sense number contained in the Birmingham output, and where the question marks in iii. and iv. indicate that we cannot prove our assumption in these cases—since these readings are not in the test vocabulary—but have a strong suspicion (Cp., for example, also the two senses labelled "2" in the output for definitions of "attention" in Deliverable 6, Section 1). The best solution would probably be to define new sense numbers within the dictionary. Until this is done, the representation of the values of `sense` is suppressed in the current version of the Bochum system. It should be clear, though, that this can be changed again with very little effort.

At this point we leave the predicate logic notation by stepwise transcribing it to the representation in complex feature structure format. Hence we map *acquire*(*x*, *y*) to

$$\underset{acquire.1}{\begin{bmatrix} \text{ARG.1} & [x] \\ \text{ARG.2} & [y] \end{bmatrix}}$$

where the specification of RELN—i.e. "relation" in the sense of situation semantics as assumed by standard HPSG—has already been replaced by the corresponding type, which represents our above disambiguated predicate.[13] The values of ARG.1 and ARG.2 stand for the two possible arguments, i.e. "*x*" for the subject and "*y*" for the object of the verb. `match1` of `co-text1` and `match2` of `co-text2` supply us with information as to the selectional preferences in the use of the headword, or restrictions on its arguments. These crucial statements are carried over to the HPSG representation by expanding the above variables to more specific and complex values:

$$\underset{acquire.1}{\begin{bmatrix} \text{ARG.1} & \underset{you}{[3]}\begin{bmatrix}\text{ARG.1} & [1]\end{bmatrix} \\ \text{ARG.2} & \underset{s\text{-}th}{[4]}\begin{bmatrix}\text{ARG.1} & [2]\end{bmatrix} \end{bmatrix}}$$

By association of the first argument of **acquire**(.1) with a certain type—*you*—this feature structure description reflects the constraint expressed in the dictionary[14] that the subject of a well-formed sentence where the word is used must conform to the restrictions formulated in `co-text1` of the LHS. The analogous statement holds for the second argument, that is, our type *s-th* contains all the information implied by the value of `co-text2` in the LHS.[15]

Tags [1] and [2] expand to INDEX, or agreement, information of involved nouns, i.e.

$$\begin{bmatrix} \text{PER} & \top \\ \text{NUM} & \top \\ \text{GEND} & \top \end{bmatrix}$$

where we use "⊤" (top) in a rather casual way to represent a disjunction over all the values of the defined value range. These features will, however, only rarely be specified in the verbal lexical entries we derive, since virtually all of them will correspond to base forms.

Let us turn to the RHS now. Recall that the value of `hinge` is in our adaptation to HPSG expressed by the inherent relation of the CONT | RESTR and COXT | BACKGROUND values as well as to hierarchic semantic information. `superordinate`, or `synonym` in our present example, tells us that the second two-place predicate involved, first has to be mapped to

[13] Cp. Section 4.1. and Pollard & Sag (1993:8.5.1.)

[14] Cp. Section 2 on the general assumptions in Cobuild underlying the use of "you" and similar meta-language elements.

[15] Cp. also Section 4.4.4.

$$obtain.n\begin{bmatrix}\text{ARG.1}\ \underset{you}{[3]}\begin{bmatrix}\text{ARG.1}\ [1]\end{bmatrix}\\ \text{ARG.2}\ \underset{s\text{-}th}{[4]}\begin{bmatrix}\text{ARG.1}\ [2]\end{bmatrix}\end{bmatrix}$$

"*n*" means that, at present, we cannot tell which sense of **obtain** is meant in the definition, from whose inherent information it cannot be deduced. The tags now carry the information given in `match1` and `match2` of the RHS. Note that we do not employ a type corresponding to "it" in `co-text2`, since "it" clearly refers back to, or actually means the same as, "something" in the LHS, and hence receives the same type as `match2` in the LHS.

Taking `hinge` into consideration now, we arrive at the following attribute-value matrix:[16]

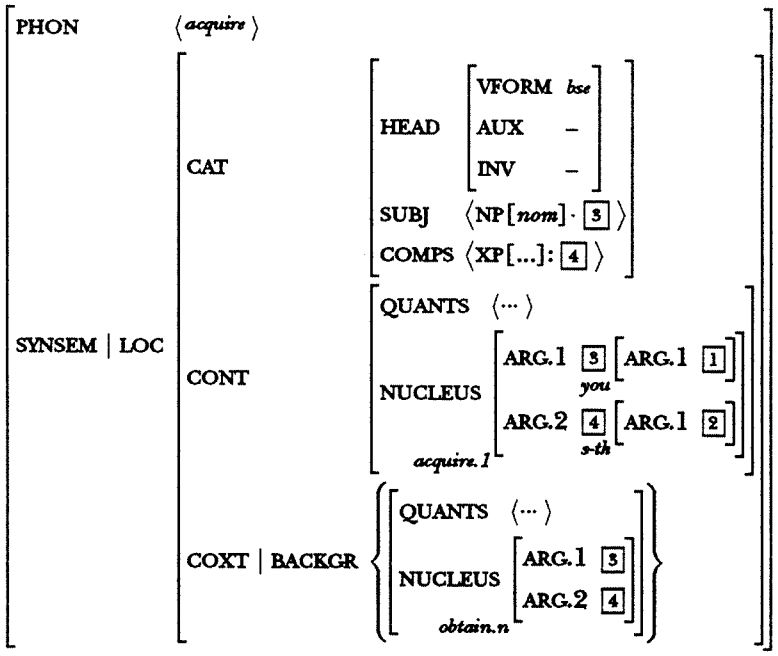
$$\begin{bmatrix}\text{CONT} & \begin{bmatrix}\text{QUANTS}\ \langle\cdots\rangle\\ \text{NUCLEUS}\ \underset{acquire.1}{\begin{bmatrix}\text{ARG.1}\ \underset{you}{[3]}\begin{bmatrix}\text{ARG.1}\ [1]\end{bmatrix}\\ \text{ARG.2}\ \underset{s\text{-}th}{[4]}\begin{bmatrix}\text{ARG.1}\ [2]\end{bmatrix}\end{bmatrix}}\end{bmatrix}\\ \text{COXT} \mid \text{BACKGR} & \left\{\underset{obtain.n}{\begin{bmatrix}\text{QUANTS}\ \langle\cdots\rangle\\ \text{NUCLEUS}\ \begin{bmatrix}\text{ARG.1}\ [3]\\ \text{ARG.2}\ [4]\end{bmatrix}\end{bmatrix}}\right\}\end{bmatrix}$$

This is in essence the bulk of the semantic information we extracted from the output of the Birmingham parser, which however, supplies far more information than that. Consider now the analysis of **acquire**(.1) in terms of the grammar assumed by the Birmingham team:

```
(    (def_number     241)
     (sense          1)
     (def_type       1)
     (lemma          '(acquire acquires
                       acquiring acquired ))
     (grammar        '(VB with OBJ))
     (pre
          (co-text0
               '()
          )
     )
     (op-word
          (hinge
               '(if)
          )
     )
     (lhs-1
          (co-text1
            (match1
               '(you)
            )
          )
          (head1
               '(acquire)
          )
          (co-text2
            (match2
               '(something)
               '(,)
            )
          )
     )
     (rhs-2
          (match1
               '(you)
          )
          (synonym
               '(obtain)
          )
          (match2
               '(it)
          )
     )
     (post
          (note '()
          )
     )
)
```

[16] Braces denote set values, not disjunctions.

The corresponding feature structure of HPSG yields the following representation:




Comparing this AVM with the one before shows one of the major advantages of HPSG, namely that semantic and syntactic information can straightforwardly be merged in a single framework. Consider now the value range of the CAT attribute, which yields the syntactic information specific to the verb **acquire**(.1). The values of HEAD—not to be confused with `head1` in the Birmingham grammar—state that it belongs to a certain type of words, here *verb*. Our parser of course received this information from the value of `grammar` in the Birmingham output. Implicitly, this also means that **acquire**(.1) must take a syntactic subject (SUBJ) in the sentence where it is used, and this syntactic argument, a nominative noun phrase, must carry the information contained in the first semantic argument of the verb, or `match1` of the LHS. Our analysis thus yields

$$\text{NP}[nom] \cdot \boxed{3}$$

As stated before, the centered dot in the above expression denotes a path leading not to the complete CONT information but to the NUCLEUS specification. Hence this is a deviance from common HPSG approaches, where only INDEX information is assumed to be identical in the subcategorisation list and the argument positions—expressed by

$$\text{NP}[nom]\boxed{1}$$

Recall also that the tags do not imply a duplication of information, but structure sharing or token, not type, identity. In graph-theoretic terms, our AVMs are so-called directed graphs. Hence the above tags simply denote re-entrant graphs, and thus the tag in the syntactic subject requirements of our verb just points at the semantic information; it does not copy it. The syntactic object, i.e. complement, requirements are expressed analogously. The value of COMPS (incidentally, another list value, indicating that there is an inherent order involved, e.g. in the case of ditransitive verbs), a syntactic category which currently is not specified further, receives the semantic value of `match2` in the LHS:

$$\text{XP}[...]: \boxed{4}$$

In most verbs labelled 'VB WITH OBJ' in the dictionary, this unspecified verb complement can be expanded to an accusative noun phrase,

$$\text{NP}[acc] \cdot \boxed{4}$$

Now, all the information we have dealt with so far, is of local syntactic-semantic nature, i.e. in terms of HPSG it belongs into the value range of SYNSEM | LOC. A linguistic sign also comprises phonological information, though. Since a more complex analysis of the phonological information is outside the scope of the present work, we simply take the first value of `lemma`, and after comparison of it with the value of `head1`, insert it in the string constituting the value of PHON:

$$\left[\text{PHON } \langle \textit{acquire} \rangle\right]$$

The same procedure is also employed in the isolation of the `head1` predicate, by the way. This checking of both constituents is made necessary by, for example, phrasal verbs, where we find cases like **pick** as the `lemma`, and then **pick up** or **pick at** as `head1`.

#### 4.3.1.2 Mapping of nouns

Before we get to more complicated cases, let us now consider an easy noun definition:

1 COUNT N
**A pitcher** is a jug.

This simple Type 3 definition is analysed by the Birmingham system as follows:

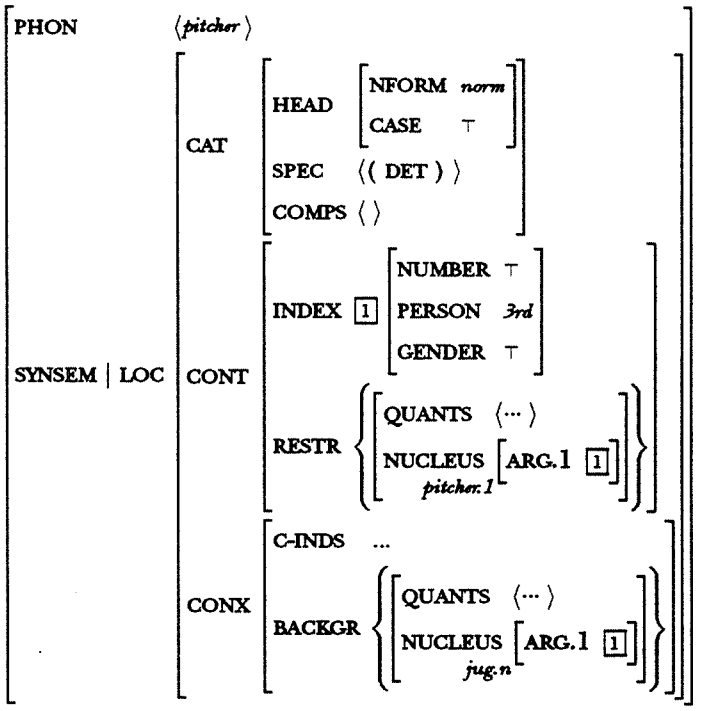
```
(    (def_number     20171)
     (sense          1)
     (def_type       3)
     (lemma          '(pitcher pitchers ))
     (grammar        '(COUNT N))

     (pre
          (co-text0
               '()
          )
     )
     (lhs-1
          (match_article
               '(a)
          )
          (head1
               '(pitcher)
          )
     )
     (link-word
          (hinge
               '(is)
          )
     )
     (rhs-2
          (match_article
               '(a)
          )
          (synonym
               '(jug)
          )
     )
     (post
          (note
               '(an American use.)
          )
     )
)
```

Recall that in a first step all definitions are implicitly mapped to a logically regimented language which also for nouns assumes the structure of logical implication. In the case of **pitcher** the mapping would yield

*If something(x) is a pitcher.1, then something(x) is a jug.1*

This step justifies employing the same CONT/COXT relation used in verbs, where the implicational relation is only more explicit in the Cobuild definitions in that Type 1 definitions include an "if" in the definition language. Thus the `hinge` value "is" of Type 3 definitions is mapped to the same structure as the `hinge` value "if" in Type 1 definitions.

Now compare the HPSG representation corresponding to the Birmingham output:




Let us look at the differences with respect to verbs, beginning with HEAD information. First, we have a CASE specification which is pretty obvious. Then we find




which indicates that, according to HPSG, **pitcher**(.1) belongs to the open class of *normal* nouns, in opposition to the closed class of nouns containing only the expletives "it" and "there", respectively. This analysis is straightforwardly taken from `grammar`. In the value range of SPEC, we find the specifier requirements of the noun, indicating whether it takes a determiner, or can be used without one. Frequently, this list value will be a disjunction containing an optional element. In the present case, the constellation

```
(lhs-1
     (match_article
          '(a)
     )
     (head1
          '(...)
     )
)
(link-word
     (hinge
          '(is)
     )
)
```

together with 'COUNT N' is an indication that a determiner is necessary when the noun is used in singular. The COMPS value, which will be mostly an empty list in nouns, is reserved for cases where nouns take complements, like the "picture of" class.

Unlike in verbs, the value range of CONT expands to the complex features RESTR and INDEX, which has already been mentioned. We presently can't supply the specification of gender here, since this information is not contained in the definition itself, but must be looked up elsewhere in the dictionary. Provided that the entire dictionary is available for such queries, the correct GENDER

value can be found quite easily by following the inferential chain via **jug** to **container** where we find the information that a container "is something ...".

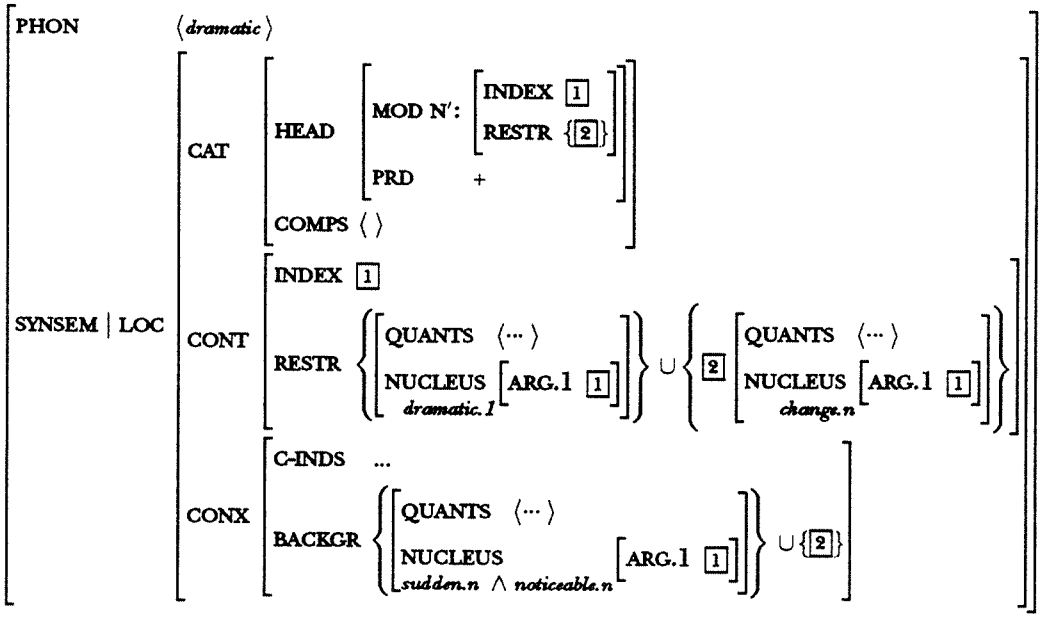
Please note that in opposition to standard HPSG approaches, where the first and almost always only semantic argument of nouns is called an "instance"—hence the corresponding feature INST—we employ the general feature ARG.1 to indicate this indentified argument position.

#### 4.3.1.3 Mapping of adjectives

Finally to adjectives, and this is where things get a bit more complicated. Our (this time not quite so) simple case is

1 ADJ
A **dramatic** change is sudden and noticeable.

Again, the `hinge` value, here "means", is expanded to the implicit implicational relation between the CONT and COXT values, and we get the following HPSG structure:




Let us start with the CONT value. Cobuild defines a very specific sense (1) of **dramatic**, namely the one the adjective has when used together with **change**, and hence our analysis also incorporates the `co-text2` value of the LHS, which specifies just this selectional restriction. It is expressed in our structure by the symbol for set-theoretic union "∪", denoting —in a very intuitive sense—the simultaneous presence (in one instance) of the properties of being a change and being dramatic.[17] Now, the same relation between the adjective and the modified noun (MOD N′) also holds in the RHS, or in terms of HPSG, in the value range of CONT | RESTR. Only, we find a conjunction in the `synonym`, which is expressed in our structure by the corresponding logical operator between the type names of the predicates (**sudden**, **noticeable**) in the superordinate space of the AVM. This superordinate or synonym space is the value in CONTEXT | BACKGROUND that does not carry the restriction tag of the modified noun, here 2.

In standard HPSG, the above logical notation would have to be interpreted as set-theoretic membership of both types—represented by specification of RELN—in the set of BACKGROUND infons, denoted by the following AVM:

[17] Cp. the discussion in Pollard & Sag (1993:8.3) on the possible logical treatment of these set-theoretic notions. This has also to be seen in conjunction with our operators over the type names corresponding to RELN values.

$$\left[\text{CONX} \mid \text{BACKGR} \left\{\left[\begin{array}{ll}\text{RELN} & \textit{sudden.n}\\ \text{INST} & \boxed{1}\end{array}\right], \left[\begin{array}{ll}\text{RELN} & \textit{noticeable.n}\\ \text{INST} & \boxed{1}\end{array}\right]\right\} \cup \{\boxed{2}\}\right]$$

Now that we have dealt not only with adjectives but also with a harmless case of conjunction, let us consider a definition of an adjective that includes disjunctions:

1 ADJ
A **lame** excuse, argument, or remark is poor or weak.

This is one of the tragic cases where we find a disjunction not only in the LHS but also in the RHS. The corresponding HPSG structure necessarily contains a number of logical operators:

$$\left[\begin{array}{ll}\text{PHON} & \langle \textit{lame} \rangle\\ \text{SYNSEM} \mid \text{LOC} & \left[\begin{array}{ll}\text{CAT} & \left[\begin{array}{ll}\text{HEAD} & \left[\begin{array}{ll}\text{MOD N':} & \left[\begin{array}{ll}\text{INDEX} & \boxed{1}\\ \text{RESTR} & \{\boxed{2}\}\end{array}\right]\\ \text{PRD} & +\end{array}\right]\\ \text{COMPS} & \langle\,\rangle\end{array}\right]\\ \text{CONT} & \left[\begin{array}{ll}\text{INDEX} & \boxed{1}\\ \text{RESTR} & \left\{\left[\begin{array}{ll}\text{QUANTS} & \langle \cdots \rangle\\ \text{NUCLEUS} & \underset{\textit{lame}}{\left[\text{ARG.1}\ \boxed{1}\right]}\end{array}\right]\right\} \cup \left\{\boxed{2}\left[\begin{array}{ll}\text{QUANTS} & \langle \cdots \rangle\\ \text{NUCLEUS} & \underset{\textit{excuse} \vee \textit{argument} \vee \textit{remark}}{\left[\text{ARG.1}\ \boxed{1}\right]}\end{array}\right]\right\}\end{array}\right]\\ \text{CONX} & \left[\begin{array}{ll}\text{C-INDS} & \ldots\\ \text{BACKGR} & \left\{\left[\begin{array}{ll}\text{QUANTS} & \langle \cdots \rangle\\ \text{NUCLEUS} & \underset{\textit{poor} \vee \textit{weak}}{\left[\text{ARG.1}\ \boxed{1}\right]}\end{array}\right]\right\} \cup \{\boxed{2}\}\end{array}\right]\end{array}\right]\end{array}\right]$$

The notation obviously needs some explanation. The general notational means in HPSG to represent disjunctions is parentheses with the logical operator enclosed, i.e.:

$$(\ldots \vee \ldots)$$

The above disjunctions are however to be understood as disjunctions over the entire set values, and not over the elements of the respective sets, of course. In standard notation, the above matrix would become far too large for representation. Consider, for example, the (almost) standard representation of the above RESTR value, which—admittedly—still is rather sloppy but already hints at the representational problem:

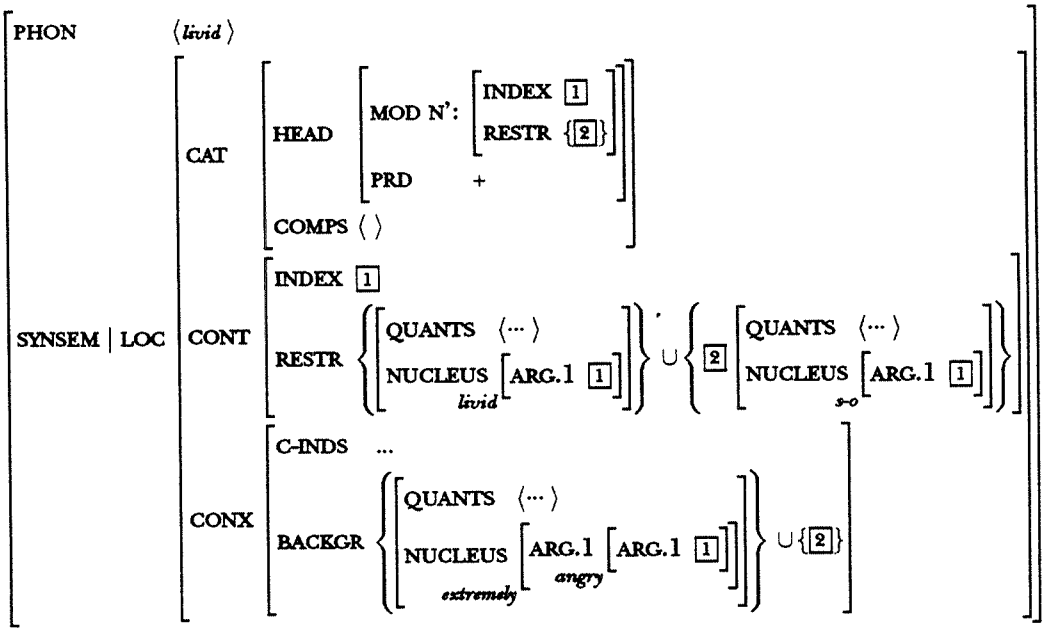
$$\left[\text{RESTR} \left\{\left[\begin{array}{ll}\text{RELN} & \textit{lame.2}\\ \text{INST} & \boxed{1}\end{array}\right]\right\} \cup \boxed{2}\left(\begin{array}{l}\left\{\left[\begin{array}{ll}\text{RELN} & \textit{excuse.n}\\ \text{INST} & \boxed{1}\end{array}\right]\right\}\\ \vee\\ \left\{\left[\begin{array}{ll}\text{RELN} & \textit{argument.n}\\ \text{INST} & \boxed{1}\end{array}\right]\right\}\\ \vee\\ \left\{\left[\begin{array}{ll}\text{RELN} & \textit{remark.n}\\ \text{INST} & \boxed{1}\end{array}\right]\right\}\end{array}\right)\right]$$

What is expressed here in any case is naturally the restriction of `co-text2` in the LHS of the definition. Since we find a disjunction there, it necessarily carries over to the `synonym` in the RHS, whose own disjunction then causes a a further multiplication of information.

Let us finally look at the definition of **livid**,

1 ADJ
Someone who is **livid** is extremely angry.

which we map to




Whereas in the case of **dramatic** we found a rather specific restriction on the modified noun, the definition of **livid** just implies that the involved noun must denote a person. This carries over to our analysis in that we include a type *s-o* in the description of the modified noun which expresses just this information of `co-text2` of the LHS. More important, however, is that the `discriminator` of the `superordinate` has been analysed as a specific type in the present example. The type *extremely* ensures that the expression is not interpreted as "being angry and being extreme at the same time", as was the case with **sudden** and **noticeable** in the definition of **dramatic**. In this respect it is formally treated similar to attributive adjectives of the "alleged" class.

#### 4.3.1.4 Mapping of the most general types

In opposition to the common practice in standard HPSG treatments, we do not assume representation of semantic roles by specific feature names, but by direct infonic encoding of the respective information as conditions on ARG.*n* values expressed by semantic types. This then also reflects the selectional restrictions adopted in Cobuild. In general, the following mappings hold:

- `(match:`$x$ `'(you))` $\Rightarrow$ $\underset{you}{[\text{ARG.1 } \boxed{x}]}$
- `(match:`$x$ `'(someone))` $\Rightarrow$ $\underset{s\text{-}o}{[\text{ARG.1 } \boxed{x}]}$
- `(match:`$x$ `'(something))` $\Rightarrow$ $\underset{s\text{-}th}{[\text{ARG.1 } \boxed{x}]}$
- `(match:`$x$ `'(someone or something))` $\Rightarrow$ $\underset{s\text{-}o \vee s\text{-}th}{[\text{ARG.1 } \boxed{x}]}$

Where $_{s\text{-}o \vee s\text{-}th}[\text{ARG.1 } \boxed{x}] = (\ _{s\text{-}o}[\text{ARG.1 } \boxed{x}] \ \vee \ _{s\text{-}th}[\text{ARG.1 } \boxed{x}])$

- (match:$x$ '(it)) $\Rightarrow$ $_{lhs\text{-}1\text{-}type}[\text{ARG.1 } \boxed{x}]$

Where ***lhs-1-type*** stands for the value the associated match:$x$ in the LHS carried, i.e. "(match:$x$ '(it))"of the RHS is in the HPSG representation assigned the type of the matched value in the LHS, namely (lhs-1 (match:$x$ '(...)). The most frequent type assigned will be *s-th*, and hence the typical mapping is:

- (match:$x$ '(it)) $\Rightarrow$ $_{s\text{-}th}[\text{ARG.1 } \boxed{x}]$

In the cases of he or she, they, and them, the general assignments will be:

- (match:$x$ '(he or she)) $\Rightarrow$ $_{lhs\text{-}1\text{-}type}[\text{ARG.1 } \boxed{x}]$
- (match:$x$ '(they)) $\Rightarrow$ $_{lhs\text{-}1\text{-}type}[\text{ARG.1 } \boxed{x}]$
- (match:$x$ '(them)) $\Rightarrow$ $_{lhs\text{-}1\text{-}type}[\text{ARG.1 } \boxed{x}]$

The 'default' assignment to types in HPSG will for they be:

- (match:$x$ '(they)) $\Rightarrow$ $_{s\text{-}o}[\text{ARG.1 } \boxed{x}]$
- (match:$x$ '(they)) $\Rightarrow$ $_{s\text{-}o \vee s\text{-}th}[\text{ARG.1 } \boxed{x}]$

As an example of how the general expressions in the Cobuild definition text are expanded to HPSG types consider the analysis of

1 VB
If a king or queen **abdicates**, he or she resigns.

Here, the Birmingham output he or she is in the CONX value range expanded to the more specific information of the CONT range—or LHS:

$$\left[\begin{array}{ll}
\text{PHON} & \langle \textit{abdicate} \rangle \\
\text{SYNSEM} \mid \text{LOC} & \left[\begin{array}{ll}
\text{CAT} & \left[\begin{array}{ll}
\text{HEAD} & \left[\begin{array}{ll} \text{VFORM} & \top \\ \text{AUX} & - \\ \text{INV} & \top \end{array}\right] \\
\text{SUBJ} & \langle \text{NP}[\textit{nom}]\boxed{1} \rangle \\
\text{COMPS} & \langle\,\rangle
\end{array}\right] \\
\text{CONT} & \left[\begin{array}{ll}
\text{QUANTS} & \langle \cdots \rangle \\
\text{NUCLEUS} & \ _{\textit{abdicates}}\left[\begin{array}{ll} \text{ARG.1} & \left[\boxed{1}\right] \\ \textit{king} \vee \textit{queen} & \end{array}\right]
\end{array}\right] \\
\text{CONX} & \left[\begin{array}{ll}
\text{C-INDS} & \ldots \\
\text{BACKGR} & \left\{\left[\begin{array}{ll}
\text{QUANTS} & \langle \cdots \rangle \\
\text{NUCLEUS} & \ _{\textit{resigns}}\left[\begin{array}{ll} \text{ARG.1} & \left[\boxed{1}\right] \\ \textit{king} \vee \textit{queen} & \end{array}\right]
\end{array}\right]\right\}
\end{array}\right]
\end{array}\right]
\end{array}\right]$$

The above output of d21 shows the standard coindexing of INDEX information in the syntactic and the semantic arguments. Also, the feature name ARG.1 is supressed in one-place predicates for spatial economy. Adapted to the approach adopted in the preceding sections, the output under discussion would yield the AVM for **abdicate** as shown before:

$$\left[\begin{array}{ll} \text{PHON} & \langle \textit{abdicate} \rangle \\ \text{SYNSEM} \mid \text{LOC} & \left[\begin{array}{ll} \text{CAT} & \left[\begin{array}{ll} \text{HEAD} & \left[\begin{array}{ll} \text{VFORM} & \top \\ \text{AUX} & - \\ \text{INV} & \top \end{array}\right] \\ \text{SUBJ} & \langle \text{NP}[\textit{nom}] \cdot \boxed{2} \rangle \\ \text{COMPS} & \langle\,\rangle \end{array}\right] \\ \text{CONT} & \left[\begin{array}{ll} \text{QUANTS} & \langle \cdots \rangle \\ \text{NUCLEUS} & \underset{\textit{abdicate}}{\left[\text{ARG.1} \quad \boxed{2} \underset{\textit{king} \vee \textit{queen}}{\left[\text{ARG.1} \quad \boxed{1}\right]}\right]} \end{array}\right] \\ \text{CONX} & \left[\begin{array}{ll} \text{C-INDS} & \ldots \\ \text{BACKGR} & \left\{\underset{\textit{resign}}{\left[\begin{array}{ll} \text{QUANTS} & \langle \cdots \rangle \\ \text{NUCLEUS} & \left[\text{ARG.1} \quad \boxed{2}\right] \end{array}\right]}\right\} \end{array}\right] \end{array}\right] \end{array}\right]$$

Recall that the substructure $\underset{\textit{abdicate}}{\left[\text{ARG.1} \quad \boxed{2} \underset{\textit{king} \vee \textit{queen}}{\left[\text{ARG.1} \quad \boxed{1}\right]}\right]}$

abbreviates

$$\underset{\textit{abdicate}}{\left[\text{ARG.1} \quad \boxed{2} \left(\begin{array}{l} \underset{\textit{king}}{\left[\text{ARG.1} \quad \boxed{1}\right]} \\ \vee \\ \underset{\textit{queen}}{\left[\text{ARG.1} \quad \boxed{1}\right]} \end{array}\right)\right]}.$$

#### 4.3.1.5 Mapping of Type 1 definitions to HPSG

The functions discussed below are either ignored, or appear only implicitly in the HPSG representation.

- The functions `pre` and `co-text0` appear in each entry of Type 1 definitions:

```
(pre
    (co-text0
            '()
    )
)
```

  Since these functions are never assigned values in the test vocabulary, they are presently both ignored in the mapping to HPSG.

- The meanings of `op-word` and `hinge` are implicitly contained in the HPSG representation: they constitute the implicational relation between the CONT and COXT values in the logical analysis and reflect the position in the semantic hierarchy.

```
(op-word
          (hinge
               '(if)
          )
)
```

*The different types of the LHS*

- `match1` before `head1`, as in "If something **abides** ...":

```
(lhs-1
          (co-text1
            (match1
               '(...)
               ...
               '(...)
            )
          )
          (head1
               '(...)
          )
)
```

This constellation is typically the case in intransitive verbs. The HPSG structure yielded by our analysis is:

$$\left[\begin{array}{ll} \text{PHON} & \langle \textit{abide} \rangle \\ \text{SYNSEM} \mid \text{LOC} & \left[\begin{array}{ll} \text{CAT} & \left[\begin{array}{ll} \text{HEAD} & \left[\begin{array}{ll} \text{VFORM} & \top \\ \text{AUX} & - \\ \text{INV} & \top \end{array}\right] \\ \text{SUBJ} & \left\langle \text{NP}[\textit{nom}]\boxed{1} \right\rangle \\ \text{COMPS} & \langle\,\rangle \end{array}\right] \\ \text{CONT} & \left[\begin{array}{ll} \text{QUANTS} & \langle \cdots \rangle \\ \text{NUCLEUS} & \left[\begin{array}{l} \text{ARG.1}\ \boxed{1} \\ \textit{s-th} \end{array}\right] \\ \textit{abides} & \end{array}\right] \\ \text{CONX} & \left[\begin{array}{ll} \text{C-INDS} & \ldots \\ \text{BACKGR} & \left\{\left[\begin{array}{ll} \text{QUANTS} & \langle \cdots \rangle \\ \text{NUCLEUS} & \left[\begin{array}{l} \text{ARG.1}\ \boxed{1} \\ \textit{s-th} \end{array}\right] \\ \textit{continues} & \end{array}\right]\right\} \end{array}\right] \end{array}\right] \end{array}\right]$$

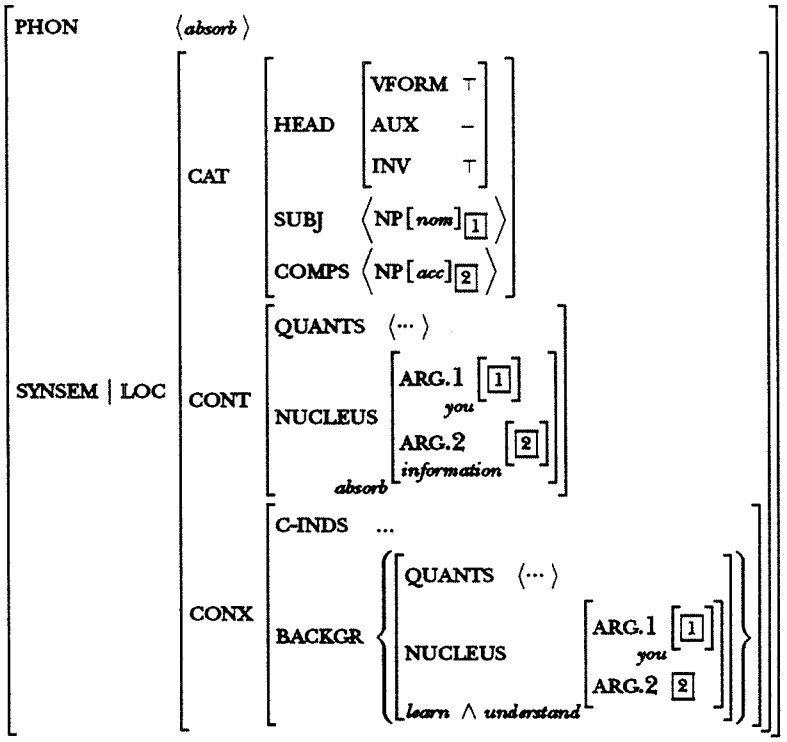
Note that the RHS of the definitions is still largely unanalysed.

- `match1` immediately before `head1` followed by `match2`, as in "If you **absorb** information ...":

```
(lhs-1
          (co-text1
            (match1
               '(...)
               ...
               '(...)
            )
          )
          (head1
               '(...)
          )
          (co-text2
            (match2
               '(...)
```

```
                ...
                '(...)
            )
        )
    )
```

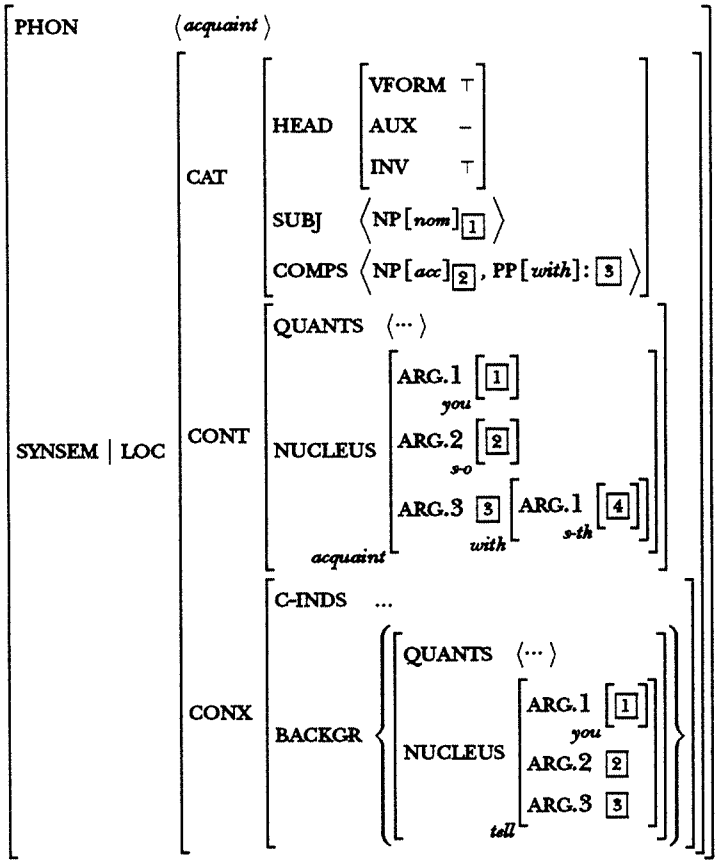
This is typical of transitive verbs, and is mapped to the following HPSG structure:




- The sequence `match1`, `head1`, `match2`, and `match3`, as in "If you **acquaint** someone with something ...":

```
(lhs-1
        (co-text1
          (match1
             '(...)
             ...
             '(...)
          )
        )
        (head1
             '(...)
        )
        (co-text2
          (match2
             '(...)
             ...
             '(...)
          )
         (match3
             '(...)
             ...
             '(...)
          )
        )
)
```

There are only few ditransitive verbs in the test vocabulary. Their HPSG representation results in an expanded subcategorisation list:



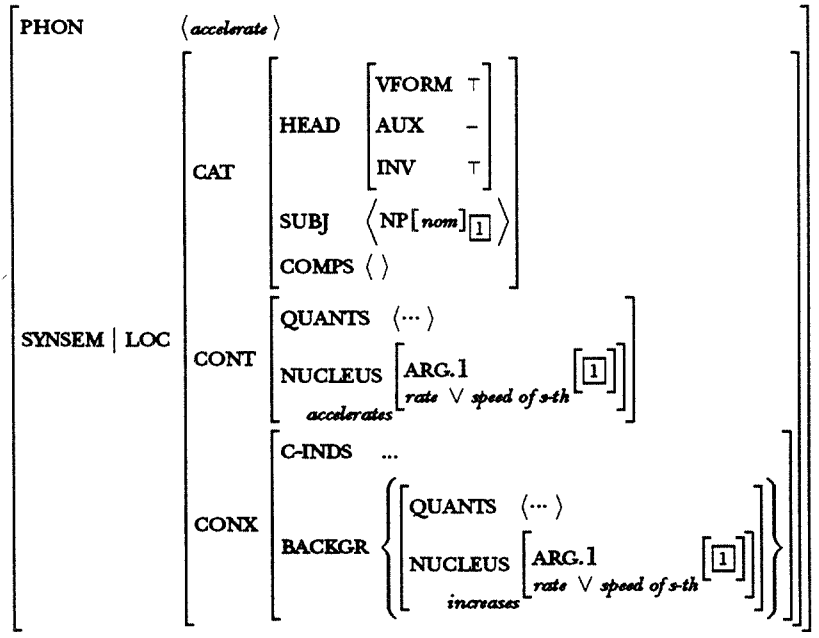

The missing type names of arguments 2 and 3 in the BACKGROUND value range are due to the fact, that the analysis of the RHS is still not complete in complex cases like the above.

*The mapping of the RHS*

The analysis of the discriminating parts in the RHS still creates problems, and can currently only be achieved in cases where we find a synonym which has no complex structure, like

1 VB
When the rate or speed of something **accelerates**, it increases.

Our, in this case almost complete, analysis yields this HPSG structure:

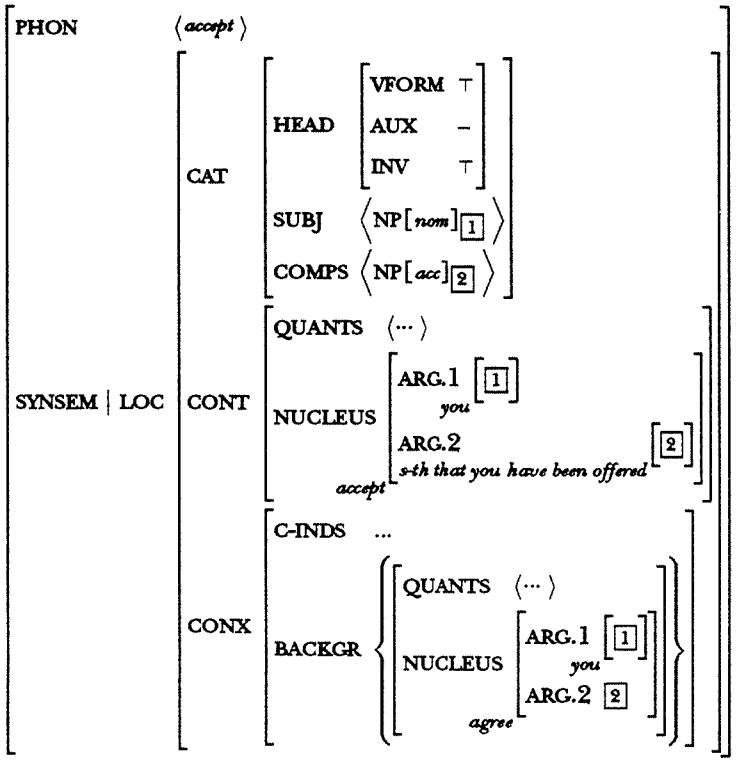



What remains to be done, of course, is the breaking up of the disjunction in the LHS—*rate* ∨ *speed of s-th*—and the analysis of the resulting two prepositional phrases.

The incomplete analysis of the discriminating parts can lead to even non-sensical HPSG structures. Consider for example

1 VB WITH OR WITHOUT OBJ
If you **accept** something that you have been offered, you agree to take it.

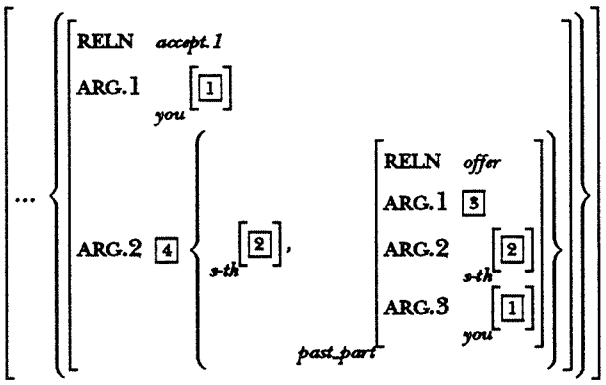
The resulting structure is:



Of course, you cannot "agree something", which is implied here. However, this is chiefly due to the fact that "to take" is analysed as a `discriminator` and not as a part of the `superordinate` in the Birmingham output:

```
(rhs-2
     (match1
          '(you)
     )
     (superordinate
          '(agree)
     )
     (discriminator
          '(to)
          '(take)
     )
     (match2
          '(it)
     )
)
```

The analysis of *s-th that you have been offered* yielded an almost satisfactory result in an earlier version[18] of `d21` (still employing RELN),




but the involved procedures lead to several unwanted consequences in similar structures, which necessitates a re-formulation of the relevant algorithms.

#### 4.3.1.6 Mapping of Type 3 definitions to HPSG

The following functions below are either ignored, or appear only implicitly in the HPSG representation.

- Like with Type 1 definitions, the functions `pre` and `co-text0` also appear in each entry of Type 3 definitions:

```
(pre
    (co-text0
          '()
    )
)
```

  Since these functions are never assigned values in the test vocabulary, they are both ignored in the mapping to HPSG.

- The most frequent value of `hinge` is of course "is":

[18] Cp. Deliverable 6, Section 3. Note, however, that the CONT and CONX value ranges have been re-structured in the later stages of the project.

```
(link-word
          (hinge
               '(is)
          )
)
```

It is interpreted by standard as constituting the CONT/CONX relation in the HPSG analysis, and hence does not appear explicitly in the corresponding AVMs.

- The second of the three possible values `hinge` can take is "are":

```
(link-word
          (hinge
               '(are)
          )
)
```

However, quantification is currently ignored. The logical relation between the LHS and the RHS would not change though, and hence "is" and "are" are treated as identical values for the time being.

- Finally we have the value "means":

```
(link-word
          (hinge
               '(means)
          )
)
```

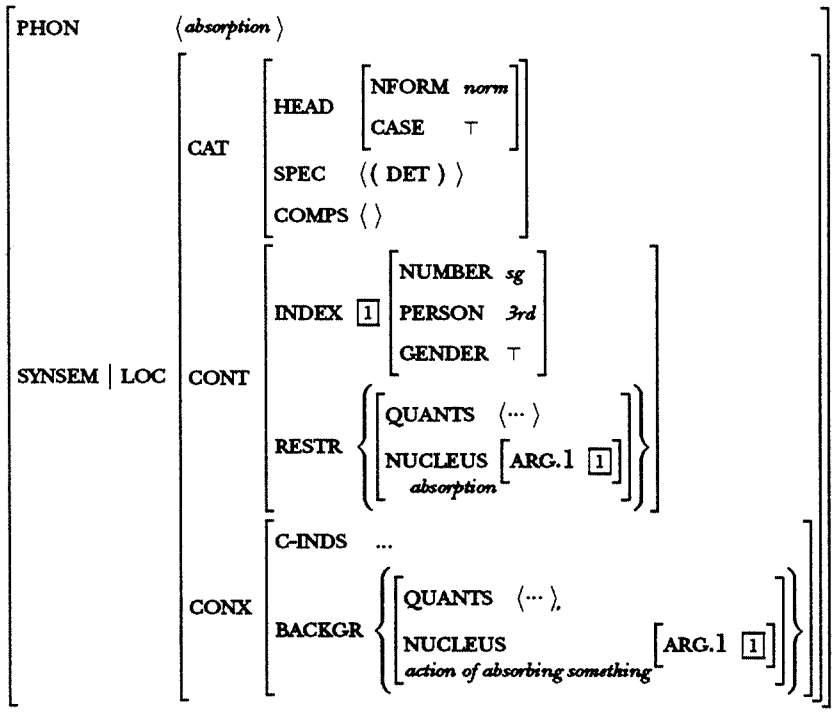
"means" only appears as value of `hinge` in definitions of adjectives. The only exception in the test vocabulary is the definition of **pictorial**, which is identified there as a 'COUNT N', and that is of course not the case. Hence the generalisation holds. The logical structure does not seem to deviate from the structure of one-place noun predicates, and therefore the "means" and "is" values of `hinge` can be treated in the same way.

*The different types of the LHS*

- `head` as the single item, as in "**Absorption** is ...":

```
(lhs-1
          (head1
               '(...)
          )
)
```

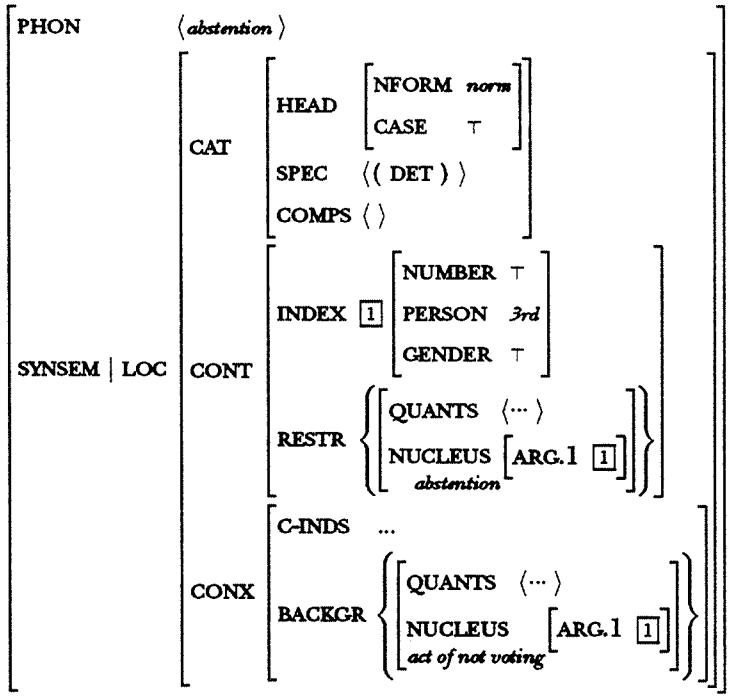
This structure maps to a simple one-place predicate in HPSG, without any complication. `head1` is compared with the first value of `lemma`, the capitalisation will then be ignored, and we can directly derive the defined predicate. In the case of **absorption** the following HPSG structure is yielded:




- head preceded by `matching_article`, as in "An **abstention** is ...":

```
(lhs-1
        (match_article
             '(...)
        )
        (head1
             '(...)
        )
)
```

The mapping in these cases is equally straightforward, since `head1` supplies the only (one-place) predicate and the article is at present not analysed. In the instance of **abstention** we arrive at the following structure:

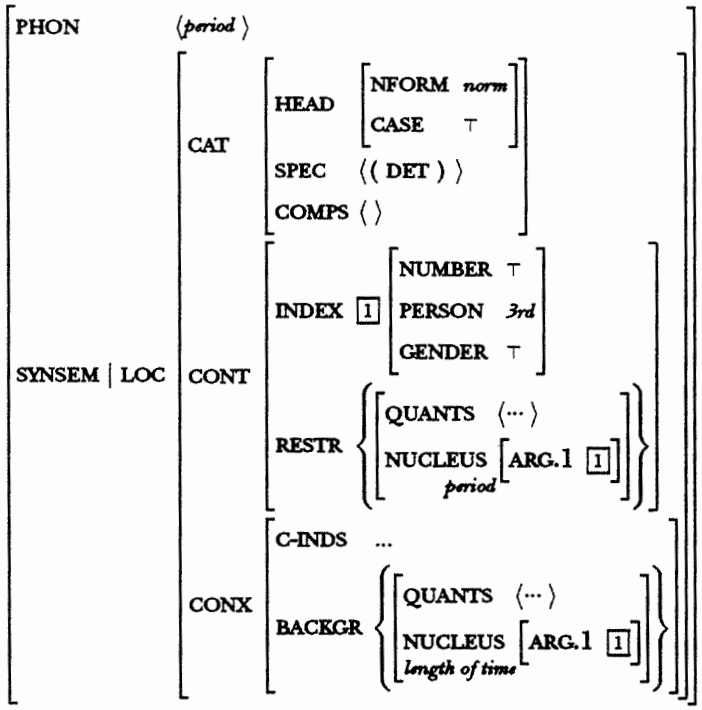



Please recall that at this point the RHS is still not completely analysed.

- head preceded by matching_article and co-text1, as in "A particular period ..." (This is the only case in the test vocabulary):

```
(lhs-1
          (match_article
               '(...)
          )
          (co-text1
               '(...)
          )
          (head1
               '(...)
          )
)
```

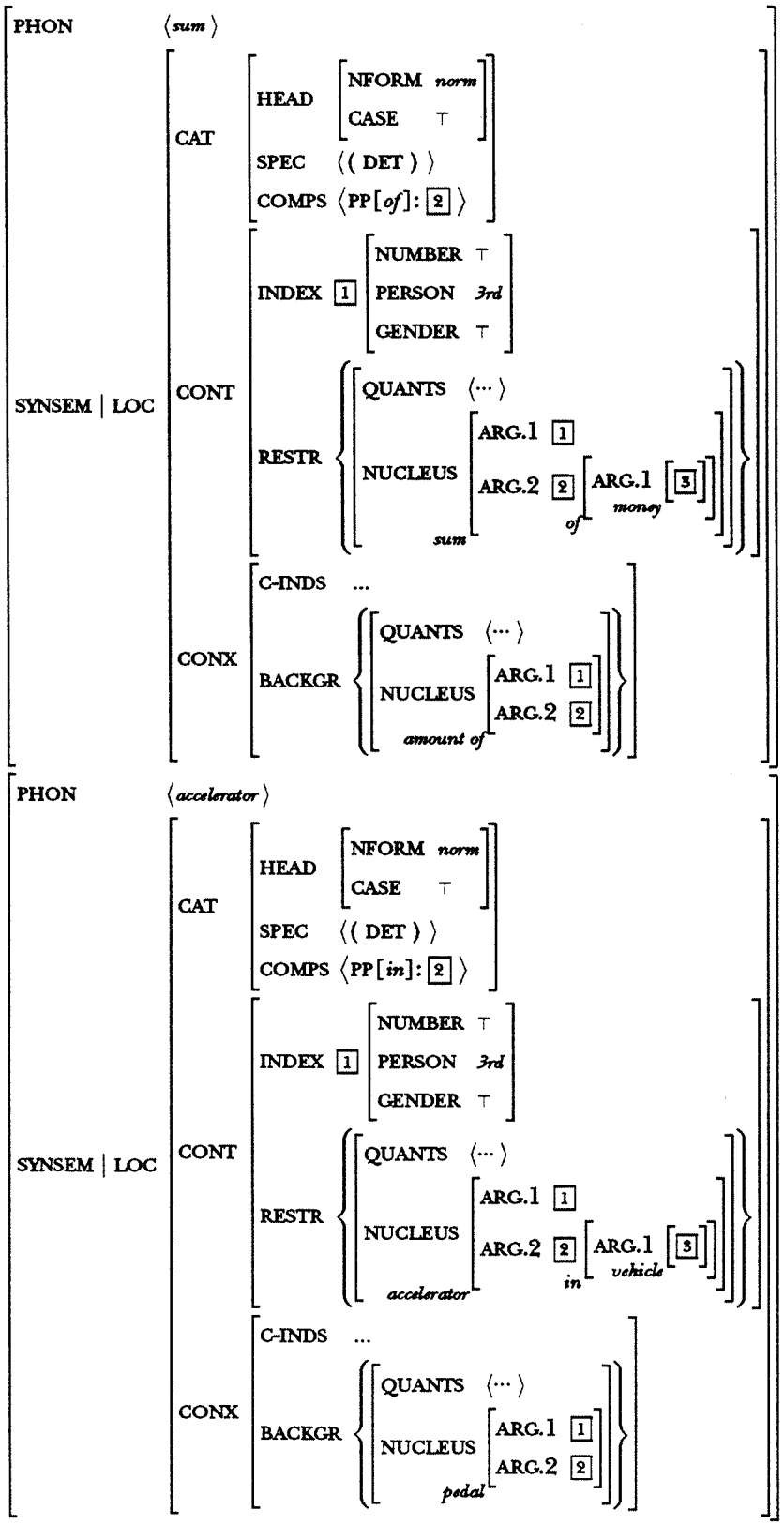
"particular" is by standard not translated into a corresponding type in the HPSG structure, hence we arrive at:




- head preceded by matching_article and followed by co-text2, as in "A **sum** of money is ..." or "The **accelerator** in a vehicle is ...":

```
(lhs-1
          (match_article
               '(...)
          )
          (head1
               '(...)
          )
          (co-text2
               '(...)
               ...
               '(...)
          )
)
```

The expression "of money" is interpreted as a second argument of the defined predicate. Unfortunately, the same analysis is performed on "in a vehicle", which is not interpreted as the specification of a spatial location by an adjunct. Both interpretations carry over to the syntactic structure of the entries, and the information is mapped to the following HPSG structures, respectively:



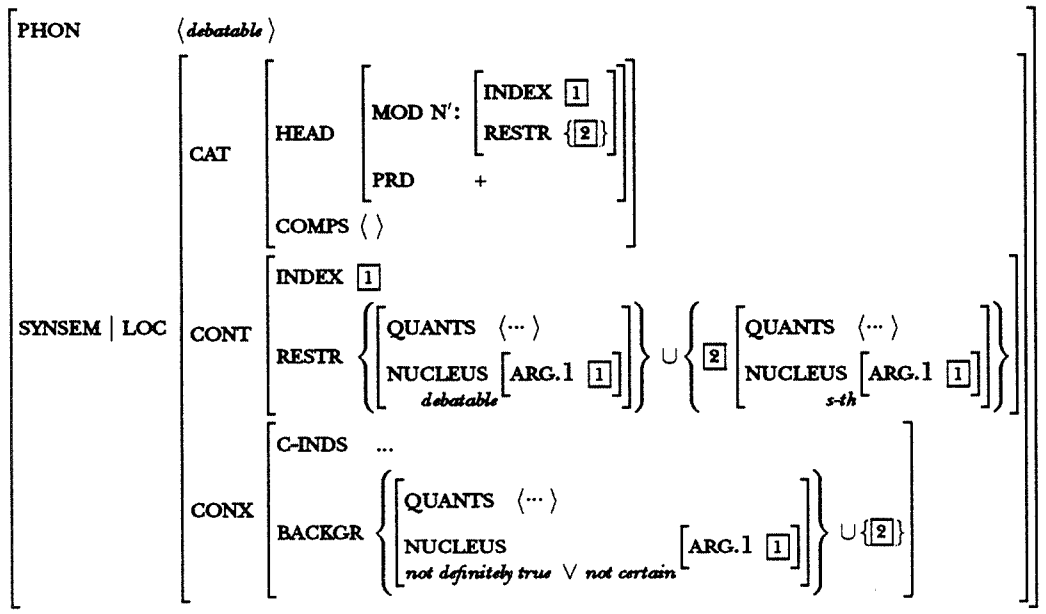

These analyses show a number of problems. First, whereas "of money" is analysed as a co-text in the LHS of the Birmingham output, "amount of" is specified as the synonym in the RHS.

Hence the incorrect argument assignments in the HPSG structure. Secondly, "in a vehicle" is not assigned to the value range of the yet to be introduced feature LOCATION, which would ensure that this expression is not treated as a second argument. This faulty analysis also triggers the specification of a prepositional phrase as the syntactic argument co-indexed with the ARG.2 value.

- `head` preceded by `co-text1`, as in "Something that is **debatable** is ...":

```
(lhs-1
        (co-text1
            '(...)
        )
        (head1
            '(...)
        )
)
```

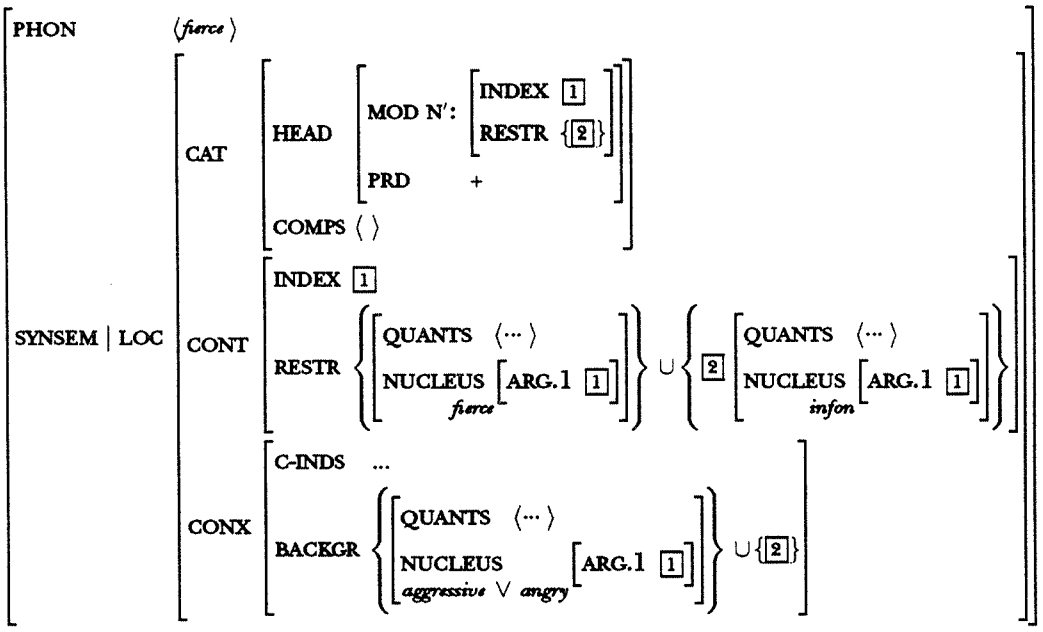
Negation in the RHS constitutes a major logical problem if the defined headword is of a very general type. Hence we only map to the following HPSG structure for the time being:




- `head` appearing before `co-text2` but not preceded by `matching_article` or `co-text1`, as in "**Fierce** means ...":

```
(lhs-1
        (head1
            '(...)
        )
    )
    (link-word
        (hinge
            '(...)
        )
    )
```

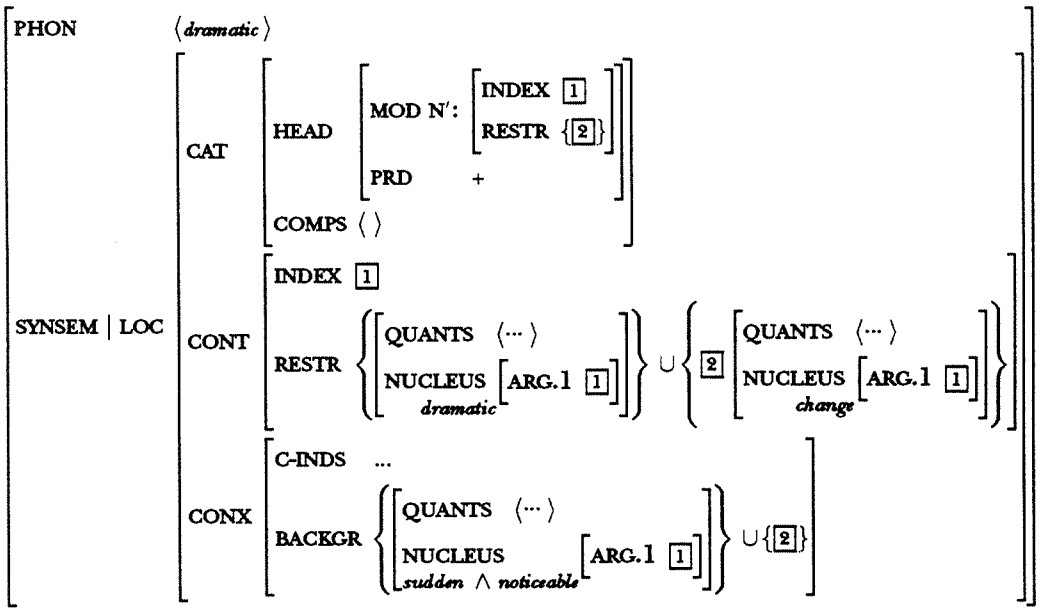
Note that the lacking specification of a selectional preference for the modified noun triggers the assignment of the most general semantic type within the RESTR set in the following HPSG structure:




- head appearing between co-text1 and co-text2, as in "A **dramatic** change ...":

```
(lhs-1
          (co-text1
               '(...)
          )
          (head1
               '(...)
          )
          (co-text2
               '(...)
               ...
               '(...)
          )
)
```

The corresponding HPSG structure below has been discussed before and does not constitute a problem for the analysis.

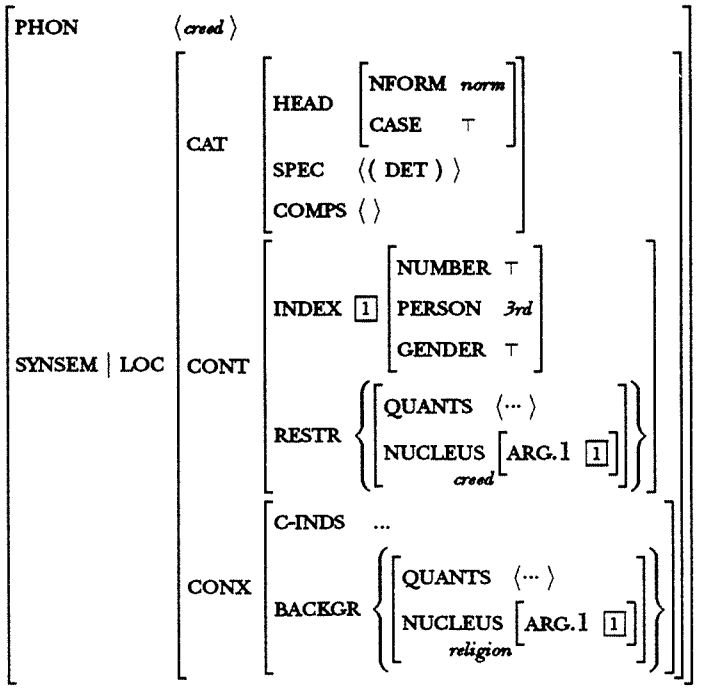



*The mapping of the RHS*

The RHS of Type 3 definitions could so far only be analysed in the cases where we find a `synonym` that is not complex, as in

**2** COUNT N
A **creed** is also a religion.

The resulting AVM in HPSG format is:




Note that the expression "also" has been ignored in the analysis, since it doesn't constitute a differentiation in the meaning of **creed**. As with Type 1 definitions, the analysis of `discriminator`

still creates difficulties, but we are optimistic that they can be resolved for the vast majority of the entries.

### 4.3.2 d2l—"dictionary to logic"

Below we give a brief description of `d2l`. This corresponds to the online manual page distributed with the software package.

**Name**

`d2l`—(mnemonic for "dictionary to logic")— convert Birmingham output to complex feature structures in HPSG format.

**Synopsis**

`[-bchvVNAP]`
`[--grammar` *pattern*`]` `[-g` *pattern*`]`
`[--entry` *pattern*`]` `[-e` *pattern*`]`
`[--deftype` *num*`]` `[-T` *num*`]`
`[--defno` *num*`]` `[--sense` *num*`]`
`[--Ntype1]` `[--bham]` `[--cobuildtext]`
`[--tex]` `[--TeX]` `[--verbose]` `[--help]`
`[-o` *file*`]` `filename`

**Description**

`d2l` takes as input the output of the Birmingham parser, parses it and converts it to HPSG format, currently yielding attribute-value matrices in TEX code. At a later stage, this could be converted to ALEP structures and SGML.

It reads its input from standard input or from *filename* and normally the result will be printed on standard output unless you use the `-o` option. The majority of options restricts the number of entries to be parsed from the input. Most of these options can be used in combination in order to yield an even more restricted set of entries to be parsed. Repetition of options is interpreted as an inclusive disjunction, e.g. `-T 1 -T 3 -A` will parse all adjectives that are of Definition Type 1 or 3, not giving you the Type 4 adjectives.

**Options**

**--defno** *num* Parse the entry with the given definition number.

**--deftype** *num*, **-T** *num*
Parse entries of the specified definition type.

**--entry** *pattern*, **-e** *pattern*
Parse the entries matching the lemma given by the regular expression *pattern*.

**--grammar** *pattern*, **-g** *pattern*
Parse the entries with the specified grammar string.

**--defno** *num* Parse the entry for the specified *definition* number.

**--sense** *num* Parse the entries for the specified *sense* number.

**--Ntype1** Parse nouns with definition type 1.

**--bham, -b** Additionally display the Birmingham output for comparison.

**--cobuildtext, -c**
Additionally display the original Cobuild entry.

**--help, -h** Display a help message.

**--tex, --TeX** Specify type of output. Need not be specified since it is used as default. Future revisions of this program might employ options yielding ALEP or SGML structures as output.

**--verbose, -v** Write information on current processing on `stderr`.

**--version** Write current version info on `stdout`. (Please use this information to report any problems.)

**-o** *file* Write output to file.

**-A** Parse all adjectives.

**-N** Parse all nouns.

**-P** Parse all phrases and phrasal verbs.

**-V** Parse all verbs.

**Files** `avm.tex` TEX macro file for typesetting AVMs.

**Notes** TEX is a trademark of the American Mathematical Society.

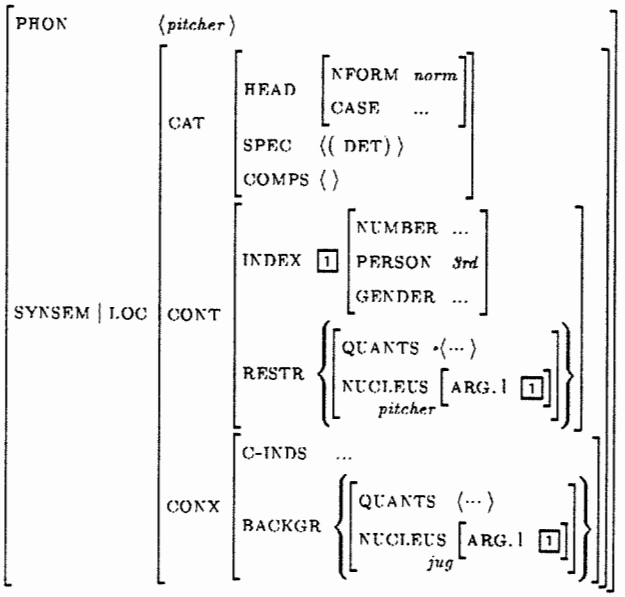
**Author** E-mail bug reports to Rolf Wilkens:
`wilkens@linguistics.ruhr-uni-bochum.de`.

Figure 4 below shows one item of the TEX–compiled output of `d21` produced by the command

```
d21 -N -v -c -b -o nouns.tex bhamdata:
```

Output produced by d2l — TEX-file compiled on May 30, 1994 at 16:02 134

**1** COUNT N
A **pitcher** is a jug




```
(    (def_number     20171)
     (sense          1)
     (def_type       3)
     (lemma          '(pitcher pitchers ))
     (grammar        '(COUNT N))
     (pre
          (co-text0
               '()
          )
     )
     (lhs-1
          (match_article
               '(a)
          )
          (head1
               '(pitcher)
          )
     )
     (link-word
          (hinge
               '(is)
          )
     )
                                        (rhs-2
                                             (match_article
                                                  '(a)
                                             )
                                             (synonym
                                                  '(jug)
                                             )
                                        )
                                        (post
                                             (note
                                                  '(an American use.)
                                             )
                                        )
                                   )
```

**FIGURE 4: Sample of d2l output produced by "`d2l -N -v -c -b -o nouns.tex bhamdata`"**

## 4.4 The implications of d2l

### 4.4.1 The reusability of Cobuild-based HPSG entries

#### 4.4.1.1 The relation to existing ALEP grammars

As has quite clearly been shown in Deliverable 8 (Section 4.1), the bulk of information extracted from Cobuild entries can be mapped to ALEP syntax—via representation in HPSG—by employing the algorithms outlined above. In cases where the Cobuild information proved not to be representable without substantial effort outside the scope of the project, the problems reside, first, in the lack of formal expressiveness of ALEP, and, secondly, in the developmental status of HPSG, which still has several theoretical shortcomings (cp. Section 4.2 on possible approaches to overcome these).

The relationship between HPSG and ALEP has been discussed in detail in van Genabith et al. (1994:91–115), where we also find the description of the ALEP implementation of a fragment of English. In this section, we will focus on the ALEP type and feature systems employed there and will argue that

- Cobuild information derived by `d2l` can be mapped to the Essex system straightforwardly.
- A considerable portion of the rich semantic information derived from Cobuild will be lost when mapped to such systems with little semantic expressiveness.

Consider now the Essex-HPSG entry for **eats** from van Genabith et al. (1994:102) [sic]:

```
eats~
wordty:{phon=>[phty:{string=>[eats|Rest],reststr=>Rest}],
        synsem=>synsemty:{loc=>locty:{cat=>catty:
                     {head=>mainvty:{vform=>fin},
                      lex=>plus,
%%SLASHED OBJECT      comps=>[],
                      subj=>[synsemty:{loc=>locty:
            {cat=>catty:{head=>nounty:{case=>nom}},
             con=>nomobjty:{index=>@SUBJ refty:{numb=>sg,
                                                pers=>'3'}
                                               }}]},
                                      con=>psoaty:
          {nucleus=>@PSOA psoa2ty:{reln=>eats,
                                   arg1=>SUBJ,
                                   arg2=>OBJ
                                   }}},
                                nonloc=>nonlocty:
                         {inher=>udcty:{
%%SLASHED OBJECT
           slash=>[locty:{cat=>catty:{head=>nounty:{case=>acc}},
                          con=>nomobjty:{index=>@OBJ refty:{}
                                         }}],
                  que=>[]}}},
        qstore=>quanlity:{string=>Quants,reststr=>Quants}
       }.
```

And now compare and bear in mind the corresponding Cobuild entry:

1 VB WITH OR WITHOUT OBJ
When you **eat** something, you put it into your mouth, chew it, and swallow it.

Most of the information expressed in the ALEP entry originates in the complex HPSG templates for a highly economic type system. For instance, the specification of a nominative case NP subject is due to the selectional properties inherent in virtually all English verbs, and the specification of `lex` as `plus` is predictable from the absence of phrasal daughters. Information of this kind can be built into general templates a priori and doesn't pose a problem for our mapping algorithms. There is, however, information specific to **eats** and its immediate supertypes that must be encoded

in addition to the general information supplied beforehand by theoretical assumptions. Three of these complex informational components are relevant for our present purpose:

i. the semantic information contained in the `cont` specification,
ii. the information on syntactic variation expressed by the `nonloc` and `comps` value ranges,
iii. the semantic information relevant for morphological processes

*Semantic information*
The algorithms underlying `d21` can supply the semantic information needed for the Essex ALEP system without any problem. Since the semantic arguments are encoded in strict agreement with HPSG terms, this information is in fact already available and only needs to be translated to the ALEP syntax by a trivial process. Compare the the following TEX source code which was derived automatically from the Cobuild entry for **acquire**[19]:

```
\avm{%
phon&\phonvalue{acquire}\cr%%
synsem | loc&\TypesLaptrue\avm{%
cat&\TypesLaptrue\avm{%
head&\TypesLaptrue\avm{%
vform&\T\cr%%
aux&\-\cr%%
inv&\T\cr%%
}\cr%%
subj&\listvalue{%
\npnom1}\cr%%
comps&\listvalue{%
\npacc{2}}\cr%%
}\cr%%
cont&\TypesLaptrue\avm{%
quants&\emptyList\cr%%
nucleus&\TypesLaptrue\typedavm {acquire }\avm{%
arg.1&\TypesLaptrue\typedavm {you }\avm{%
\avmtag{1}\cr%%
}\cr%%
arg.2&\TypesLaptrue\typedavm {s-th }\avm{%
\avmtag{2}\cr%%
}\cr%%
}\cr%%
}\cr%%
}\cr%%
conx&\TypesLaptrue\avm{%
c-inds&...\cr%%
backgr&\setvalue{%
\TypesLaptrue\avm{%
quants&\emptyList\cr%%
nucleus&\TypesLaptrue\typedavm {obtain }\avm{%
arg.1&\TypesLaptrue\typedavm {you }\avm{%
\avmtag{1}\cr%%
}\cr%%
arg.2&\tagvalue{2}\cr%%
}\cr%%
}}\cr%%
}\cr%%
}\cr%%
}
```

It should be obvious that an adaptation of this output to an ALEP format corresponding to the Essex system would not be very difficult.

The ALEP entries in the Essex system do not, however, reflect any semantic restriction or preference information on the subject and object. Also, there is apparently no specification of the CONX feature of standard HPSG which are used to represent the superordinate complex of the

[19] The Cobuild entry for **eat** was not included in the test vocabulary and hence the according information could not be extracted automatically. The two-place predicate *acquire* should, however, suffice for the present purpose.

Cobuild definition. Hence, a very important aspect of the entry for **eat**, namely that it inherits information from **take**, **chew**, and **swallow** cannot be mapped to the present ALEP system although it can be readily supplied by d21—the information in our entries would simply have to be deleted.

*Syntactic variance information*

Due to the lexical-semantic emphasis of the project, we have so far not studied the encoding of non-local information mainly relevant for syntactic description. For their treatment of unbounded dependency constructions, the Essex team needs to encode slashed categories in the lexical entries, though. We are convinced that a more thorough study of the grammar information of the Cobuild definitions than was possible in the course of the project would yield the desired results. The extremely large class of English verbs that allow this kind of construction can very probably be isolated by comparatively checking the grammar information of the Cobuild entries and the semantic types of their superordinates and synonyms.

The syntactic information that we can derive and represent immediately is the fact that the syntactic realisation of the second semantic argument of **eat** is optional. This follows directly from the grammar information VB WITH OR WITHOUT OBJ contained in the Cobuild entry. Hence, we can ensure the generalisation that the transitive and intransitive variants of **eat** share the same semantic arguments—even if the disjunctive COMPS list cannot be represented in ALEP.

*Morphologically relevant semantic information*

For ALEP systems where inflectional information is not handled by lexical rules and where, consequently, inflected and base forms result in separate entries, we can easily supply the necessary specifications. Although we have primarily studied the base forms of the Cobuild entries, we can access the relevant inflection information in Cobuild. Consider again the Birmingham parser output for the first sense of **permit**:

```
(    (def_number     19842)
     (sense          1)
     (def_type       1)
     (lemma          '(permit permits
                       permitting permitted ))
     (grammar        '(VB with OBJ))
     (pre
          (co-text0
               '()
          )
     )
     (op-word
          (hinge
               '(if)
          )
     )
     (lhs-1
          (co-text1
            (match1
               '(you)
            )
          )
          (head1
               '(permit)
          )
          (co-text2
            (match2
               '(something)
               '(,)
            )
          )
     )
     (rhs-2
          (match1
               '(you)
          )
          (synonym
               '(allow)
          )
          (match2
               '(it)
          )
     )
     (post
          (note '(a formal use.)
          )
     )
)
```

The base form mapping procedure we employ could very simply be applied to the three inflected forms of the lemma list, yielding four to $n$ separate entries differing only in the VFORM specification.

In short: all the information necessary for HPSG/ALEP entries of the Essex system can be supplied by d21, but since this information forms a proper subset of information we have available, there is little chance that an ALEP grammar can currently handle all of the linguistically relevant information we can offer.

#### 4.4.1.2 The relation to the ITU-Corpus

Consider now a part of the concordance listing for **assembly** in the ITU-corpus:

```
rcular rail supports the mechanical assembly and allows its rotation in azimuth,
nna is composed of: -the electrical assembly (as described in  5.2.2) which cons
acing between an analogue multiplex assembly (e.g. a 60 channels FDM supergroup)
upergroup)  and a digital multiplex assembly (e.g. two 30 channels PCM groups) -
 channels in the baseband multiplex assembly. ii) In consequence, SSB transmissi
nsmit) the terrestrial FDM baseband assembly into the minimum number of supergro
 and since 1986 (CCIR XVIth Plenary Assembly), it has been studying the performa
h-over-elevation (Az-El) mechanical assembly of the wheel and track type. In thi
tive elements. It is composed of an assembly of various telecommunication sub-sy
n digital. The complete cable is an assembly of multiple individual (10-50) coax
---------------+ Note 1.- Group: an assembly of 12 telephone channels derived fr
 kHz). Note 2.- Supergroup (SG): an assembly of 60 telephone channels derived fr
be determined by the XVIIth Plenary Assembly of the CCIR, taking  5.2.4  Antenna
tion and user data between a packet assembly/disassembly (PAD) facility and a pa
ture, thus converting it into a new assembly possessing different statistical an
mmetrical) terrestrial FDM baseband assembly; -the so-called ßatellite multiple
system which permits the electrical assembly to be steered in any possible orien
stal) which supports the electrical assembly (usually on two orthogonal movable
```

Assume that this text needs to be translated into German. In order to achieve this via an ALEP system or a similar machine, the text must be parsed by an analysis process employing an English grammar and English lexical entries. The output of the process can then be mapped into a language-independent representation, which then, in turn, serves as the input to a generating system component that produces natural language output in German.

A closer look at the above concordances reveals that the single English expression **assembly** translates into two German words:

- *Konstruktion*, in the cases of
  - — "electrical assembly"
  - — "mechanical assembly"
  - — "multiplex assembly" ...
- *Versammlung*, in the case of
  - — "Plenary Assembly"

The third sense of **assembly** in Cobuild would actually translate into *Montage* rather than *Konstruktion* but an insertion of something like "... the result of the process..." would yield a *Konstruktion* reading (Cp. Deliverable 8, Section 5.1 on a general strategy of adapting Cobuild definitions to techical vocabulary in a similar fashion). Sense 1, however matches the necessary reading:

1 COUNT N
An **assembly** is a large number of people gathered together, especially a group of people who meet regularly to make laws.

3 UNCOUNT N
The **assembly** of a machine or device is the process of fitting its parts together.

Now, if we encoded the entries in the regular ALEP format with a reduced set of semantic information which mainly serves for syntactic processing, we would arrive at something similar to the following:

```
assembly~
wordty:{phon=>[phty:{string=>[assembly|Rest],restst=>Rest}],
        synsem=>synsemty:{loc=>locty:{cat=>catty:
                    {head=>nounty:{nform=>norm},
                     lex=>plus,
                     comps=>[],
                     ...

assembly~
wordty:{phon=>[phty:{string=>[assembly|Rest],restst=>Rest}],
        synsem=>synsemty:{loc=>locty:{cat=>catty:
                    {head=>nounty:{nform=>norm},
                     lex=>plus,
                     comps=>[synsemty:{loc=>locty:
            {cat=>catty:{head=>prepty:{pform=>of}}},
             ...
```

This means, for one thing, that the optional complement (a prepositional phrase with "of") cannot be expressed by a disjunctive COMPS list but must result in two separate entries, as displayed above. But lacking semantic information other than the encoding of the number of arguments and their matching with syntactic categories, these entries would imply correct parses of, for example, "plenary assembly of telephone channels". Of course, expressions like this are not to be expected in the ITU corpus or other official technical documents, but the question remains into which German term **assembly** must be translated after an English sentence has been parsed.

Unfortunately, there is no simple solution to this dilemma. An ad hoc attempt to supply the necessary information—which is present in the feature structures of HPSG—in ALEP entries could involve storing the relevant semantic feature complexes in the respective COMPS lists, like, for example:

```
assembly~
wordty:{phon=>[phty:{string=>[assembly|Rest],restst=>Rest}],
        synsem=>synsemty:{loc=>locty:{cat=>catty:
                    {head=>nounty:{nform=>norm},
                     lex=>plus,
                     comps=>[synsemty:{loc=>locty:
            {cat=>catty:{head=>prepty:{pform=>of}}},
             ...
                      con=>psoaty:
            {nucleus=>@PSOA psoa1ty:{reln=>people,
                                     arg1=>...
             ...
```

The MOD feature of adjectives like **plenary** must then also contain this selectional information.

However, this approach would have a number of problematic consequences (Cp. also Deliverable 8, Section 4.1 on technical aspects of ALEP entries). First, the subcategorisation principle (cp. van Genabith et al. (1994:112f.)) would have to be revised in a rather complex way, and secondly, the semantic information could only be used by rather complicated default techniques, given that inheritance, let alone multiple inheritance, within lexical entries is not possible in ALEP. Encoded as "hard" constraints the semantic information would of course lead to wrong grammaticality judgments otherwise.

### 4.4.2 The user's perspective

*Human user—unassisted*

The output of d21 offers a valuable source of information not only for NLP systems but also for the theoretical linguist. Given that the definition types analysed in our test vocabulary cover almost the entire set of entries of the dictionary, we can expect our procedures to produce information-enriched HPSG entries by the thousands in the near future. Together with a small set of manually compiled entries—the closed class items, like prepositions or pronouns—our system would yield a very large database accessible by either computer or in printed form. In opposition to the ALEP

user interface, d21 produces output in a very ergonomic form facilitating easy visualisation of information. From a very practical point of view, the HPSG entries produced by d21 can provide numerous examples syntax and semantics studies can be built on—also on a very abstract theoretical level without taking implementational matters into account, if you wish.

The produced entries carry over a wide range of information that has so far not been treated in sufficient depth in the HPSG framework. Our encoding of CONX information, selectional preferences, and hierarchical semantic information in addition to the normally discussed items increases the value of the system as a logically structured and analysed lexicon for the theoretical linguist.

A very important aspect of the results in the form of HPSG entries is the fact that these entries allow a totally different view on grammaticality or well-formedness judgements by theoretical linguists. Due to the corpus-based methods by which Cobuild was produced, the lexical entries derived from it are also enormously meaningful under an empirical point of view. They can actually supply the testing basis and methods that are so frequently demanded by psychologists, who very often blame linguists for their apparently too abstract judgements.

With the logical structure of Cobuild entries isolated, it should cause little effort to make the derived information available also in LFG (Lexical-Functional Grammar) and other logically friendly formats. Also, the phonetic information contained in Cobuild could be supplied for phonological studies. Its extraction from the entries is a trivial task—due to the encoding in the Cobuild data files.

*Human user—assisted by machines*

The most important aspect from this perspective seems to us the possible development of lexicographic tools producing input for NLP systems. d21 could very well be used for

- the design of an interface to lexicographic production systems
- the design of a graphical interface in order to facilitate post-processing

The idea is as follows: once the interface to a lexicographic production system has been implemented, each new lexical entry in Cobuild format can be parsed and logically analysed immediately—and automatically—after its definition by a compiler. Since a compiler's intuition should not be hampered by such analytical considerations, an interactive system for post-processing by a theoretical linguist or NLP specialist would ensure the correctness of the results of the logical output. This system must necessarily also have access to corpus information.

*The machine*

As we have briefly demonstrated in the preceding section, the information encoded in lexical entries for ALEP grammars in current use forms a proper subset of the information that can be supplied by the output of d21. Therefore it seems reasonable to assume that the majority of problems we encountered in information extraction from Cobuild would actually become irrelevant if we merely had to derive semantic and syntactic information considered in these systems. For instance, the most problematic part of the algorithm is the analysis of the rhs, i.e. information that is not normally taken into account in existing ALEP systems. Given these minimal requirements for supplying information, the rate of success of our procedures would increase drastically, and hence, the then possible input for ALEP machines would certainly exceed by far what is available now.

Independent of the specific NLP system the output of d21 might be implemented in, the semantic information encoded in our entries is very likely to form the basis of the language-neutral component in automatic multilingual applications or translation systems. Moreover, since the phonetic information supplied by Cobuild entries is easily accessible, it should also cause very little effort to make it available for speech recognition or production systems. Consider for instance an electronic dictionary in which queries can be made acoustically without the use of a keyboard.

# References


ALLPORT G., BARNBROOK G. & SINCLAIR J M (1993a)
"The Representation of Cobuild Definitions." ET-10/51 Working Paper, University of Birmingham.

ALLPORT G., BARNBROOK G. & SINCLAIR J M (1993b)
"A Grammar of Cobuild Definition Statements." ET-10/51 Working Paper, University of Birmingham.

CALZOLARI, N. (1990)
"Structure and Access in an Automated Lexicon and Related Issues." in L. Cignoni & C. Peters (eds.). Computational Lexicology and Lexicography. Special Issue Dedicated to Bernard Quemada. I., *Linguistica Computazionale*, Vol. VII, pp. 139—161.

CALZOLARI N., HAGMAN J., MARINAI E., MONTEMAGNI S., SPANU A. & ZAMPOLLI A. (1993).
"Encoding Lexicographic Definitions as Typed Feature Structures" in Beckmann, F. & Heyer, G. (eds.). *Theorie und Praxis des Lexikons.* Berlin & New York: Walter de Gruyter, pp. 274–315.

COOPER, R. (1990)
Classification-based Phrase Structure Grammar: An Extended Revised Version of HPSG. Ph.D. Thesis, University of Edinburgh.

DEVLIN, K. (1991)
*Logic and Information.* Cambridge, UK: Cambridge University Press.

ET-10/51 (1993)
"The Cobuild Functional Parser, Pisa TFS Grammar, Bochum BLF Format, and Common Interface: Draft Specifications." ET-10/51 Deliverable 3, CEC, Luxembourg.

ET-10/51 (1993)
"The Parsing of Cobuild Definitions and Mapping of the Output to Typed Feature Structures and Bochum Logical Form", ET-10/51 Deliverable 4, CEC, Luxembourg.

VAN GENABITH J., MARKANTONATOU S., SADLER L. & VERHAGEN M. (1994). "English HPSG in ALEP_0." in: Markantonatou, S. & Sadler, L. (eds.). *Grammatical Formalisms: Issues in Migration.* Luxembourg: Office for Official Publications of the Commission of the European Comunities. pp. 91–115

GROSS, M. (1993) "Local grammars and their representation by finite automata." in Hoey, M. P. (ed.) *Data, Description, Discourse.* London: HarperCollins.

HAGMAN, J. (1991)
"Common and Odd Relations in MRD Definitions and their Treatment in Taxonomy Building." ESPRIT BRA-3030 ACQUILEX Working Paper.

HANKS P. (1987)
"Definitions and Explanations." in Sinclair, J M (ed.). *Looking Up: An Account of the COBUILD Project in Lexical Computing.* London and Glasgow: Collins ELT, pp. 116–136.

MARINAI E., PETERS C. & PICCHI E. (1990)
"The Pisa Multilingual Lexical Database System." Esprit BRA 3030, Twelve Month Deliverable, ILC-ACQ-2-90, Università di Pisa.

MONTEMAGNI S., MARINAI E. & CALZOLARI N. (1992)
"Describing Cobuild Lexical Entries as Typed Feature Structures, with Particular Reference to the Semantics of Definitions." Interim Report—ET-10/51 Deliverable 2, CEC, Luxembourg, pp. 12–21.

MONTEMAGNI S. (1992)
"Using a Typed Feature Structure (TFS) Representation System for Encoding Lexical Knowledge Extracted from Dictionary Definitions: Draft Specifications for TFS Descriptions." Interim Report—ET-10/51 Deliverable 1, CEC, Luxembourg, pp. 13–61.

MOON R. (1987)
"The Analysis of Meaning." in Sinclair, J M (ed.). *Looking Up: An Account of the COBUILD Project in Lexical Computing*, London and Glasgow: Collins ELT, pp. 86–103.

PICCHI E. (1991)
"D.B.T.: A Textual Data Base System." in L. Cignoni and C. Peters (eds.). Computational Lexicology and Lexicography. Special Issue Dedicated to Bernard Quemada. I., *Linguistica Computazionale*, Vol. VII, pp. 177—205.

POLLARD C. & SAG I. (1987)
*Information-based Syntax and Semantics. Vol. 1.* Stanford, CA: CSLI (Distributed by University of Chicago Press).

POLLARD C. & SAG I. (1994)
*Head-driven Phrase Structure Grammar.* Stanford, CA: CSLI (Distributed by University of Chicago Press).

SCHNELLE, H. (1991)
*Die Natur der Sprache: Die Dynamik der Prozesse des Sprechens und Verstehens.* Berlin and New York: de Gruyter.

SINCLAIR J M (1987)
"Grammar in the Dictionary." in Sinclair, J M (ed.). *Looking Up: An Account of the COBUILD Project in Lexical Computing*, London and Glasgow: Collins ELT, pp. 104—115.

SINCLAIR J M (1990)
"The nature of lexical statement." in Yoshimura et al. (eds.). *Linguistic Fiesta.* N.P.: Kurosio Publishers. [Revised and reprinted as Chapter 9 of Sinclair (1991).]

SINCLAIR J M (1991)
"Words about Words." in Sinclair, J M (ed.). *Corpus, Concordance, Collocation*, Oxford: Oxford University Press, pp. 123—137.

# List of Authors

**Geoff Barnbrook**

School of English—Corpus Linguistics
University of Birmingham
Edgbaston
Birmingham B15 2TT
United Kingdom
barnbrog@ibm3090.bham.ac.uk

**Nicoletta Calzolari**

Istituto di Linguistica Computazionale del C.N.R.
Dipartimento di Linguistica
Università di Pisa
Via della Faggiola, 32
I-56100 Pisa
Italy
eagles@vm.cnuce.cnr.it

**Stefano Federici**

Istituto di Linguistica Computazionale del C.N.R.
Dipartimento di Linguistica
Università di Pisa
Via della Faggiola, 32
I-56100 Pisa
Italy
perimila@vm.cnuce.cnr.it

**Martin Hoelter**

Sprachwissenschaftliches Institut
Ruhr-Universität Bochum
D-44780 Bochum
Germany
hoelter@linguistics.ruhr-uni-bochum.de

**Simonetta Montemagni**

Istituto di Linguistica Computazionale del C.N.R.
Dipartimento di Linguistica
Università di Pisa
Via della Faggiola, 32
I-56100 Pisa
Italy
acquilex@vm.cnuce.cnr.it

**Carol Peters**

Istituto di Elaborazione della Informazione
Università di Pisa
Via Santa Maria, 46
I-56100 Pisa
Italy
carol@vm.iei.pi.cnr.it

**John Sinclair**

School of English—Corpus Linguistics
University of Birmingham
Westmere Mews
Edgbaston
Birmingham B15 2TT
United Kingdom
sinclaij@ibm3090.bham.ac.uk