ET-10/51
Deliverable 3

Report 3: The Cobuild Functional Parser, Pisa TFS Grammar, Bochum BLF Format, and Common Interface (Draft Specifications)

Graham Allport, Geoff Barnbrook, Mona Baker,
Nicoletta Calzolari, Stefano Federici, Martin Hoelter,
Simonetta Montemagni, Carol Peters,
Helmut Schnelle, John Sinclair, Elena Tognini-Bonelli

Sprachwissenschaftliches Institut
Ruhr-Universität Bochum
P.O. Box 10 21 48
W-4630 Bochum 1
Federal Republic of Germany

Contents

1 Birmingham: A Draft Definition of the Functional Parser Output 1

1.1 Introduction . . . 1
1.2 Main Areas of Discussion . . . 1
1.3 Parser Output Definition . . . 1
1.3.1 Type A . . . 1
1.3.2 Type B . . . 4
1.3.3 Type C . . . 5
1.3.3.1 Subtype 1 . . . 5
1.3.3.2 Subtype 2 . . . 7
1.3.4 Type D . . . 7
1.3.4.1 Subtype 1 . . . 7
1.3.4.2 Subtype 2 . . . 8
1.3.5 Type E . . . 8
1.3.6 Type F . . . 10
1.3.6.1 Subtype 1 . . . 10
1.3.6.2 Subtype 2 . . . 11

2 Pisa: Draft Specifications of TFS Grammar 12

2.1 Extraction of Semantic Information from Cobuild Definitions . . . 12
2.2 Evaluation of Information and Conversion into a TFS Formalism . . . 16
2.3 Pisa TFS Representation and the Common HPSG Format . . . 17
2.3.1 Attributes Specific to the Pisa Representation . . . 18
2.3.2 Attributes Common to HPSG and Cobuild . . . 19

3 Bochum: Draft Specifications of BLF Format and Common Interface 21

3.1 The BLF Format . . . 21
3.1.1 Format for noun specifications . . . 22
3.1.2 Format for Verb Specifications . . . 23
3.2 The Common HPSG Format . . . 24
3.2.1 Noun Specifications . . . 24
3.2.2 Verb Specifications . . . 28

1 Birmingham: A Draft Definition of the Functional Parser Output

1.1 Introduction

The parsing software for the Cobuild Student's Dictionary is now partly developed and the basic structure of its output has been largely determined. This interim report specifies draft output formats for the main definition types identified in Deliverable 2 (The Semantics of Definitions, Part I). Because of the different processsing needs of the Pisa and Bochum teams, the files delivered to them from Birmingham may ultimately differ in appearance, but the principles of analysis will be identical.

1.2 Main Areas of Discussion

Because the output format is being developed as an input to the work of the other partners in the project the final details will be agreed as a result of evaluation carried out by both Bochum and Pisa. The most likely areas for revision in the light of this evaluation will be the detailed analaysis of the DISCRIMINATOR2, FURTHER NOTES and COMMENT units.

1.3 Parser Output Definition

1.3.1 Type A

An **allergy** is an illness that you have when you eat, smell, or touch a substance which does not normally make people ill.

SENSE NO.	*1*
GRAMMAR NOTE	*COUNT N or UNCOUNT N*
ARTICLE	*An*
COTEXT1	
HEADWORD	*allergy*
COTEXT2	
HINGE	*is*
MATCHING ARTICLE	*an*
DISCRIMINATOR1	
SUPERORDINATE	*illness*
DISCRIMINATOR2	*that you have* *when you eat, smell, or touch a substance* *which does not normally make people ill*
FURTHER NOTES	

A **snowflake** is one of the soft, white bits of frozen water that fall as snow.

SENSE NO.	*1*
GRAMMAR NOTE	*COUNT N*
ARTICLE	*A*
COTEXT1	
HEADWORD	*snowflake*
COTEXT2	
HINGE	*is*
MATCHING ARTICLE	*one of the*
DISCRIMINATOR1	*soft, white*
SUPERORDINATE	*bits of frozen water*
DISCRIMINATOR2	*that fall* *as snow*
FURTHER NOTES	

The **allure** of something is a pleasing or exciting quality that it has.

SENSE NO.	*1*
GRAMMAR NOTE	*SING N*
ARTICLE	*The*
COTEXT1	
HEADWORD	*allure*
COTEXT2	*of something*
HINGE	*is*
MATCHING ARTICLE	*a*
DISCRIMINATOR1	*pleasing or exciting*
SUPERORDINATE	*quality*
DISCRIMINATOR2	*that it has*
FURTHER NOTES	

1.3.2 Type B

if you **abbreviate** a piece of writing or speech, you make it shorter.

SENSE NO.	*1*
GRAMMAR NOTE	*VB WITH OBJ*
HINGE	*If*
COTEXT1	*you*
HEADWORD	*abbreviate*
COTEXT2	*a piece of writing or speech*
COTEXT3	
MATCHING COTEXT1	*you*
SUPERORDINATE VERB	*make*
MATCHING COTEXT2	*it*
MATCHING COTEXT3	
DISCRIMINATOR	*shorter*
FURTHER NOTES	

if you **accuse** someone of something, you say they have done something wrong.

SENSE NO.	*1*
GRAMMAR NOTE	*VB WITH OBJ*
HINGE	*If*
COTEXT1	*you*
HEADWORD	*accuse*
COTEXT2	*someone*
COTEXT3	*of something*
MATCHING COTEXT1	*you*
SUPERORDINATE VERB	*say*
MATCHING COTEXT2	*they*
MATCHING COTEXT3	
DISCRIMINATOR	*have done something wrong*
FURTHER NOTES	

if you **abandon** yourself to an emotion, you feel it strongly and do not try to control it.

SENSE NO.	*3*
GRAMMAR NOTE	*REFL VB WITH FEEL*
HINGE	*if*
COTEXT1	*you*
HEADWORD	*abandon*
COTEXT2	*yourself*
COTEXT3	*to an emotion*
MATCHING COTEXT1	*you*
SUPERORDINATE VERB	*feel* *and do not try to control*
MATCHING COTEXT2	
MATCHING COTEXT3	*it* *it*
DISCRIMINATOR	*strongly* *about*
FURTHER NOTES	

1.3.3 Type C

1.3.3.1 Subtype 1

a **macabre** event or story is very strange and horrible.

SENSE NO.	*1*
GRAMMAR NOTE	*ADJ*
ARTICLE	*A*
COTEXT1	
HINGE1	
HEADWORD	*macabr*
COTEXT2	*event or story*
HINGE2	*is*
DISCRIMINATOR1	*very*
SUPERORDINATE ADJECTIVE	*strange and horrible*
DISCRIMINATOR2	
FURTHER NOTES	

a **manned** vehicle is controlled by people travelling in it.

SENSE NO.	*1*
GRAMMAR NOTE	*VD1*
ARTICLE	*A*
COTEXT1	
HINGE1	
HEADWORD	*manned*
COTEXT2	*vehicle*
HINGE2	*is*
DISCRIMINATOR1	
SUPERORDINATE ADJECTIVE	*controlled*
DISCRIMINATOR2	*by people travelling in it*
FURTHER NOTES	

someone or something that is **abnormal** is unusual or exceptional, especially in a worrying way.

SENSE NO.	*1*
GRAMMAR NOTE	*VD1*
ARTICLE	
COTEXT1	*Someone or something*
HINGE1	*that is*
HEADWORD	*abnormal*
COTEXT2	
HINGE2	*is*
DISCRIMINATOR1	
SUPERORDINATE ADJECTIVE	*unusual or exceptional*
DISCRIMINATOR2	*especially in a worrying way*
FURTHER NOTES	

1.3.3.2 Subtype 2

Something that is **inconvenient** causes problems or difficulties for you.

SENSE NO.	*1*
GRAMMAR NOTE	*ADJ*
ARTICLE	
COTEXT1	*Something*
HINGE1	*that is*
HEADWORD	*inconvenient*
COTEXT2	
SUPERORDINATE VERB	*causes*
COTEXT3	*problems or difficulties*
DISCRIMINATOR	*for you*
FURTHER NOTES	

1.3.4 Type D

1.3.4.1 Subtype 1

if you are **positive** about something, you are completely sure about it.

SENSE NO.	*1*
GRAMMAR NOTE	*PRED ADJ*
HINGE1A	*If*
COTEXT1	*you*
HINGE1B	*are*
HEADWORD	*positive*
COTEXT2	*about something*
MATCHING COTEXT1	*you*
HINGE2	*are*
DISCRIMINATOR1	*completely*
SUPERORDINATE ADJECTIVE	*sure*
DISCRIMINATOR2	*about it*
FURTHER NOTES	

1.3.4.2 Subtype 2

if you are **privy** to something secret, you know about it:

SENSE NO.	*1*
GRAMMAR NOTE	*PRED ADJ WITH PREP 'to'*
HINGE1A	*If*
COTEXT1	*you*
HINGE1B	*are*
HEADWORD	*privy*
COTEXT2	*to something secret*
MATCHING COTEXT1	*you*
SUPERORDINATE VERB	*know*
DISCRIMINATOR	*about*
MATCHING COTEXT2	*it*
FURTHER NOTES	

1.3.5 Type E

about in front of a number means approximately.

SENSE NO.	*4*
GRAMMAR NOTE	*ADV*
COTEXT1	
HEADWORD	*About*
COTEXT2	*in front of a number*
HINGE	*means*
SUPERORDINATE	*approximately*
MATCHING COTEXT2	
DISCRIMINATOR	

to **mine** a substance such as coal or gold means to obtain it from the ground by digging deep holes and tunnels.

SENSE NO.	*3*
GRAMMAR NOTE	*VB WITH OR WITHOUT OBJ*
COTEXT1	*To*
HEADWORD	*mine*
COTEXT2	*a substance such as coal or gold*
HINGE	*means*
SUPERORDINATE	*to obtain*
MATCHING COTEXT2	*it*
DISCRIMINATOR	*from the ground* *by digging deep holes and tunnels*
FURTHER NOTES	

to spend time **on** something means to spend time doing it or making it.

SENSE NO.	*18*
GRAMMAR NOTE	*PREP*
COTEXT1	*To spend time*
HEADWORD	*on*
COTEXT2	*something*
HINGE	*means*
MATCHING COTEXT1	*to spend time*
SUPERORDINATE	*doing* *or making*
MATCHING COTEXT2	*it* *it*
DISCRIMINATOR	
FURTHER NOTES	

1.3.6 Type F

1.3.6.1 Subtype 1

you use **largely** to say that a statement is mostly but not completely true.

SENSE NO.	*1*
GRAMMAR NOTE	*ADV*
COTEXT1	*You*
USAGE	
REPORTING1	*use*
COTEXT2	
HEADWORD	*largely*
HINGE	*to*
MATCHING COTEXT1	
MATCHING COTEXT2	
REPORTING2	*say that*
COTEXT3	*a statement*
COMMENT	*is mostly but not completely true*
FURTHER NOTES	

some people use **unique** to mean very unusual and special.

SENSE NO.	*3*
GRAMMAR NOTE	*ADJ*
COTEXT1	*Some people*
USAGE	
REPORTING1	*use*
COTEXT2	
HEADWORD	*unique*
HINGE	*to*
MATCHING COTEXT1	
MATCHING COTEXT2	
REPORTING2	*mean*
COTEXT3	
COMMENT	*very unusual and special*
FURTHER NOTES	

1.3.6.2 Subtype 2

you can use **lady** as a polite way of referring to a woman.

SENSE NO.	*1*
GRAMMAR NOTE	*COUNT N*
COTEXT1	*You*
USAGE	*can*
REPORTING1	*use*
COTEXT2	
HEADWORD	*lady*
HINGE	*as*
ARTICLE	*a*
DISCRIMINATOR1	*polite*
SUPERORDINATE	*way of referring*
DISCRIMINATOR2	*to a woman*
FURTHER NOTES	

2 Pisa: Draft Specifications of TFS Grammar

For the Pisa group, the present must be considered very much as an interim report on work in progress. It includes comments on issues which we are now addressing, and mentions problems which at the moment remain open and will certainly be the subject of common discussion between the partners in the next few months of the project, before final decisions can be taken.

At the moment, the activity in Pisa aimed at the definition of a parser and grammar for the extraction of semantic information from the Cobuild dictionary and its representation in TFSs is divided into three main tasks:

- analysis of the Cobuild definitions and extraction of semantic information
- interactive evaluation of the information extracted and conversion into a TFS formalism
- representation in Typed Feature Structures

Although, at a first glance, it might appear that a direct path from extraction to representation of information in TFS terms would have been possible and simpler, an intermediate stage has been found necessary for the several reasons which will be discussed below. Indeed, until the first two stages are in an advanced stage of development, it will not be possible for us to proceed very far with the third.

2.1 Extraction of Semantic Information from Cobuild Definitions

As is known, the Cobuild definitions are divided into 2 parts:

1. In the first part, the lemma being defined is given in its most typical syntactic and semantic environment;
2. The second part gives an explanation of the meaning of the lemma, specifying the unique contribution of the particular word in the contexts in which it can occur. This is done by referring to a superordinate representing the semantic class to which the item belongs and by providing information which discriminates the lemma being defined with reference to the general class represented by the superordinate.

Clearly, two different kinds of information can be extracted from the two parts. However, they can both be used, in different ways, by NLP systems. We have thus considered the two parts separately.

It was our original intention to concentrate our attention on the second part of the definition. So far, our analysis has been mainly concentrated on the Noun class. For noun definitions, at the moment this part of the definition in Cobuild is structured into: DISCRIMINATOR(1): SUPERORDINATE: DISCRIMINATOR(2).

In the parsed definitions which we had originally received from Birmingham, the discriminator parts were not further subdivided. The range of information contained in this part of the definition is considerable: obviously the same discriminator can

carry more than one kind of semantic information. However, in the raw text there is very little on which to base the rules of the semantic parser. On the following pages, we give two examples of the present stage of our analysis for the second part of noun definitions. On each page, at the top we have the formalized entry as it had been provided by Birmingham and below the information which we are currently able to extract from it. The second part, headed 'Proposal', shows the entry as we would like to have it and below the semantic information which could then be extracted.

- Currently parsed entry:

```
(entry '((slot machine)
         ((senseno (0))
          (grammar (count n))
          (article (a))
          (cotext1 ())
          (headword (slot machine))
          (cotext2 ())
          (hinge (is))
          (matchingarticle (a))
          (discriminator1 ())
          (superordinate (machine))
          (discriminator2 (from which you can get food, drink, or
                           cigarettes or on which you can gamble. you
                           work it by putting coins into a slot in the
                           machine)))))
```

- Results:

```
lemma:      slot machine
senseno:    0
is_a:       machine
source_of:  food, drink, or cigarettes or on which
            you can gamble
```

- Proposal:

```
(entry '((slot machine)
         ((senseno (0))
          (grammar (count n))
          (article ((a det)))
          (cotext1 ())
          (headword ((slot n) (machine n)))
          (cotext2 ())
          (hinge ((is vb)))
          (matchingarticle ((a det)))
          (discriminator1 ())
          (superordinate ((machine n)))
          (discriminator2 ((from prep) (which pron) (you pron)
                           (can modal) (get verb) (food n) (, comma)
                           (drink n) (, comma) (or conj) (cigarettes n)
                           (or conj) (on prep) (which pron) (you pron)
                           (can modal) (gamble vb) (. period)
                           (you pron) (work vb) (it pron) (by prep)
                           (putting vb) (coins n) (into prep) (a det)
                           (slot n) (in prep) (the det)
                           (machine n))))))
```

- Results:

```
lemma:      slot machine
senseno:    0
is_a:       machine
source_of:  food, drink, cigarettes
on_which:   gamble
```

- Currently parsed entry:

```
(entry '((spark plug)
         ((senseno (0))
          (grammar (count n))
          (article (a))
          (cotext1 ())
          (headword (spark plug))
          (cotext2 ())
          (hinge (is))
          (matchingarticle (a))
          (discriminator1 ())
          (superordinate (device))
          (discriminator2 (in the engine of a motor vehicle, which
                           produces electric sparks to ignite the
                           fuel)))))
```

- Results:

```
lemma:      spark plug
senseno:    0
is_a:       device
producing:  electric sparks to ignite
            the fuel
```

- Proposal:

```
(entry '((spark plug)
         ((senseno (0))
          (grammar (count n))
          (article ((a det)))
          (cotext1 ())
          (headword ((spark n) (plug n)))
          (cotext2 ())
          (hinge ((is vb)))
          (matchingarticle ((a det)))
          (discriminator1 ())
          (superordinate ((device n)))
          (discriminator2 (((in prep) (the det) (engine n) ((of prep)
                           (a det) (motor n) (vehicle n))) (, comma)
                           (((which pron) subj) (produces vb)
                           (((electric adj) (sparks n)) obj)
                           ((to prep) (ignite vb) (((the det) (fuel n))
                           obj)))))))
```

- Results:

```
lemma:      spark plug
senseno:    0
is_a:       device
producing:  electric sparks
purpose:    (ignite (obj fuel))
location:   (engine (of (motor vehicle)))
```

Our problem in designing a semantic parser for the second part of the definition is that, in its present state, without an additional parsing which assigns to all the word-forms in the definition their disambiguated part of speech and lemma, the only method we can use to search for information is a simple pattern searching, string matching mechanism. The difficulty here is that this kind of method is not sophisticated enough to permit us to obtain sufficiently reliable and detailed results. In the next stage of work, however, the parsed definitions supplied by Birmingham should have the part of speech (not disambiguated) attached to each word form and a splitting of the discriminators into significant word groups. When we receive this input, we will begin to examine strategies which should permit us to extract useful semantic information from this part of the definition.

With reference to the extraction of semantic information from this part of the definition, the role of the superordinate must be considered carefully. The superordinate is crucial in the design of the type system as it represents the semantic class of the lemma being defined and thus carries the common semantic nucleus which should be inherited by the lemma (currently this is not possible in Alep). It is therefore essential that there is agreement on a common concept of superordinate.

For the above reasons, we have concentrated our attention on the extraction of information from the first part of the definition which gives the headword in its typical context. The reason for this is twofold. Firstly, this is the truly innovative part of the definition specific to Cobuild, and it contains much valuable information which should be largely exploitable by NLP; secondly, it is already in a highly structured form and the development of rules to extract semantic information should thus pose less problems. The elements of the structure of this part of the definition are: HEADWORD; COTEXT(1); COTEXT(2); HINGE. The rules now being developed in our semantic parser are based on the grammatical information associated with the headword and the cotext data.

At the moment, we have focussed our attention here on definitions for the verb class and we have been evaluating the extraction and representation of four main kinds of syntactic and semantic information from this part of the definition. We feel that it will be possible to extract information which permits us to make some kind of general classification of verb "types", e.g. verbs which imply an inherent, typical action vs. possible action; verbs denoting illegal, reprehensible or unusual acts; verbs whose value is highly subjective. Information can also be extracted on subject/object selection preferences, on typical adjuncts, and on typical prepositional government (not always specified in the Grammar Codes of the Student's Dictionary).

2.2 Evaluation of Information and Conversion into a TFS Formalism

In this section we explain the reasons which have led us to adopt this three-stage approach and, in particular, the necessity for this intermediate stage.

- There is a need to evaluate and assess the information that can be extracted from the definitions before attempting to formalize and represent it: it is not always possible to immediately formalize the conditions which are used as cues for the extraction of particular information from the definitions when defining the rules for a semantic parser. The first results must thus be carefully examined in order to establish whether they are totally reliable and significant or whether at times interferences, perhaps depending on particular local conditions in the definitions, mean that further refinements to the initial hypotheses are required. In addition, at times, a particular set of conditions can lead to more than one possible interpretation and it may not always be possible to disambiguate these automatically. For this reason, we feel that it will be necessary to implement an interactive procedure which permits us to evaluate and verify the results generated automatically by the semantic parser.
- The Cobuild definitions are very rich in semantic information, but it is not always possible to represent everything that is extractable from them in TFS terms. This may be either because of the constraints imposed by the formalisms adopted (TFS and HPSG) but is also due to the proposed implementation using the ALEP system which, as has already been pointed out, does not appear particularly suitable for an exhaustive representation of lexical knowledge. It is also true that, at the present state of the art, not all of the information which can be extracted is directly exploitable by NLP systems, even though it is hoped that in the future this information can be handled by such systems.
- There is a need to produce intermediate results of our semantic parsing for those cases for which at the moment we are unable to assign a certain semantic interpretation (see the example given by the ON_WHICH relation in the SLOT_MACHINE). Such results, which we intend to store in a form halfway between a natural language representation and an abstract formalization, will comprise the basis for the formulation of new semantic rules in our future work.
- The typology of the information extracted in this stage and all its possible configurations will serve as a guide in the design and definition of the final type system. They will form the building blocks out of which we can construct the TFS lexical entries.

2.3 Pisa TFS Representation and the Common HPSG Format

As far as representation is concerned, work has only progressed on the theoretical side, given that so far most of our efforts have been concentrated on the definition strategies for extracting semantic information from Cobuild definitions. The choice of the three-stage approach, described above, should make it clear that the final design of the system of types and the subsequent representation of lexical entries in terms of Typed Feature Structures is not possible until the extraction process has been completed. Once a skeleton type system has been defined (this is what has been done by Pisa in the previous stages, and is reported in Deliverables 1

and 2), the design of the type system must be guided by the actual information extracted from Cobuild definitions, and not by abstract categories which have been established a priori. Therefore, our contribution on this topic at this stage will be mostly theoretical. There are two issues on which we would like to focus attention, issues which are crucial with respect to the relationship between the Pisa TFS representation and the common HPSG format:

i. the Pisa attributes, and
ii. the attributes and their respective values in the common HPSG interface

2.3.1 Attributes Specific to the Pisa Representation

It should be remembered that all attributes which are specific to the Pisa representation have been proposed as possible formalizations of the information contained in Cobuild entries. Whether and how they can be used by current NLP systems is another issue which is still open at the moment and will perhaps need to be taken into account in later stages of the project. The attributes we are considering here are only those involved in the definition of very general types—such as *sign*, *noun*, *verb*—which are part of the common interface.

In the common interface, the LEMMA, SENSEID and ORTHPHON attributes are all part of the definition of the type *sign*. They are meant to encode respectively the lemma, the sense number, and the orthographical and phonological information listed for each word defined in the dictionary. Their placement so high in the hierarchy of types—that is at the level of *sign*—might be revised in further stages of the project, given that this information is only relevant for lexical entries. In fact, in the type system proposed by Pisa in Deliverable 2, the LEMMA and SENSEID attributes (together with other dictionary specific attributes) were associated with the *word* type, which was in its turn a subtype of *sign*. The same attributes were also present within the definition of another type, *lexphrase* (subtype of *phrase*, in its turn subtype of *sign*), which was specifically meant to represent phrasal signs recorded as lexical entries within the dictionary.

The ORTHPHON attribute requires an additional comment. This attribute formalizes the beginning of the dictionary entry, where the spelling for each form of the lemma being defined is specified, together with the pronunciation (the pronunciation is normally given for the first form of the word; if the form is pronounced in different ways, the variants are afterwards specified; when the pronunciation of one of the inflected forms is very different from that of the base form, then its pronunciation is also given). Apart from cases in which some of the spelling or pronunciation specifications hold only for a particular word sense, this information is common to all word-senses belonging to the same part of speech. Therefore, it might appear redundant to specify this kind of information for each word-sense (word senses are our minimal representation units). In order to propose a solution in this respect, the inheritance issue needs to be tackled. More specifically, it should be possible for this kind of information to be inherited by all word senses, unless differently specified (in this latter case, it should be possible to modify part of it by overriding part of its values, or the whole structure should be simply redefined). This implies a slightly

different design of the type system which will be evaluated in a following stage of the project.

Moreover, two other attributes appearing in the common interface are relevant to the Pisa representation; they are the EXAMPLE attribute, which defines the COXT type together with the canonical HPSG BACKGROUND attribute, and the LEXSEM attribute, which, together with the canonical INDEX and RESTR, defines the CONT type. While the EXAMPLE attribute will contain the example sentences recorded within each word-sense, the LEXSEM attribute is intended to represent the lexical semantics viewpoint, that is the meaning of the word being defined. This latter attribute has been the focus of our attention so far, as can be seen from the system of types proposed in Deliverable 2.

2.3.2 Attributes Common to HPSG and Cobuild

This 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 Summary, p. 4). From this perspective, the task of defining common attributes and the range of their possible values appears to be twofold. On the one hand, it should result from a formalization process of the information contained in Cobuild, while, on the other hand, it should make such information to be usable by current NLP systems. These two different aspects do not go always in the same direction, and conflicts may arise. Therefore, this involves finding a balance between the adherence to the data and their usability in the framework of current formal grammars and NLP systems. Let us illustrate this point with a few examples.

In the type system proposed in Deliverable 2 by Pisa, the attribute COUNT was defined as follows: `COUNT => &boolean (cou/mass/uncou/nsing/nplur)`. This definition contained information which was directly relevant to the COUNT(ability) attribute—that is 'cou', 'mass', and 'uncou' — together with other information which was not relevant, such as the 'nsing' and 'nplur' values, corresponding to the 'n sing' and 'n plur' Cobuild specifications (the 'n sing' specification refers to nouns only used in the singular and which must have a determiner in front of them, while 'n plur' refers to nouns only used in the plural form). The reason why we have been obliged to encode this information in this way was because of the possible cooccurrence of grammar codes (see Deliverable 2, p. 17); there are Cobuild entries for which a disjunction such as 'n plur OR n uncount' is specified. The method shown above was thus a simplistic way of encoding this kind of disjunction. In the meantime, Birmingham provided us — as requested—with a list of all possible combinations of grammar codes, on the basis of which to design the system of types with respect to this kind of information. We are currently working on this issue. We reported the case here because it must be evaluated whether the attributes and the values to be assigned to them are expected to reflect the Cobuild data or should be translated in canonical HPSG terms. This latter solution might cause the loss of useful information, which cannot be converted into HPSG despite its usefulness from an NLP point of view.

Another case in point is the 'supp' specification, associated with supporting words, that is words which are not "usually used on their own in this meaning and either need an adjective in front of them, or a relative clause or prepositional phrase after them" (Cobuild Student's Dictionary, p. xiii). This kind of specification, which is very useful from the human user perspective, cannot be translated as such in HPSG terms: from the formal grammar point of view, this statement appears to be underspecified, and needs to be integrated with other information, referring to categories such as subcategorized complement, or modifier. Therefore, we are considering what should be our general policy in this respect, whether to formalize it as it occurs in Cobuild (that is as it has been conceived for human users), or to integrate it with the missing information, and translate it into HPSG terms, or to lose it because there is no direct mapping between the Cobuild specification and the HPSG formulation.

These points together with other similar cases will evaluated during the next stage of our work and decisions will have to be taken before we can make significant progress with the representation side of our task.

3 Bochum: Draft Specifications of BLF Format and Common Interface

In the second phase of the project, Bochum has to perform two tasks. First, the development of the Bochum Logical Form must be pursued further in order to arrive at a theoretically well-founded semantic analysis that can then serve as the basis on which future formats for NLP systems can rest. In devising such a framework, it must necessarily be taken into account that the output of the analysis is adaptable—with respect to its representation—and translatable into the currently predominant NLP formats, which employ attribute value systems. Bochum therefore decided at an early stage in the project to ensure the representation of the output in an HPSG-compatible format (cp. Deliverable 2, Part 3).

Secondly, Bochum will have to supply a common feature-based format that can serve as an interface in which the output of the different analyses of the partners can be represented and, after a second intermediate step, ultimately implemented in Alep. Given the shortcomings of the Alep system, which have been pointed out in the previous reports by the Pisa team, this last step will most probably cause a considerable loss of information.

3.1 The BLF Format

The general strategy of a semantic analysis using BLF has already been pointed out in Deliverables 1 and 2. Here, we will concentrate on the representational issues that are crucial for the aspect of reusability in feature based NLP systems like Alep. Recall, however, that there are three indispensable stages in the BLF analysis:

i. disambiguation of Cobuild definitions
ii. conversion of definitions to a regimented *if ... then* format
iii. conversion of definitions to predicate logic format

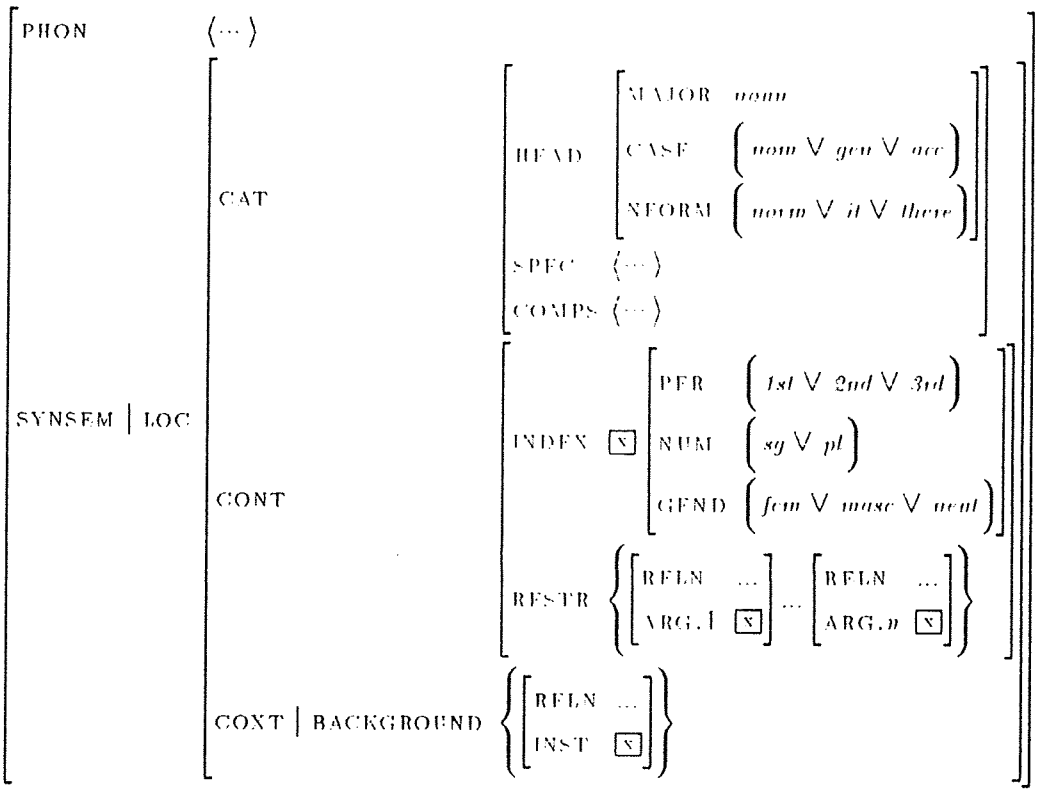
The actual application of this analytic strategy will of course be highly dependent on the output of the Birmingham parser. As was shown in Deliverable 2, the transfer of information from the functional parser output to the feature-structured BLF "dialect" is definitely tractable in principle.

The central problem we will face in the automatic conversion of the functional parser output will be the isolation of predicates from the Cobuild definitions. Given the promising revisions of the breakdown of entries produced by the Birmingham team, the DISCRIMINATOR1...*n* and SUPERORDINATE units of the output seem to be a significant step towards a solution.

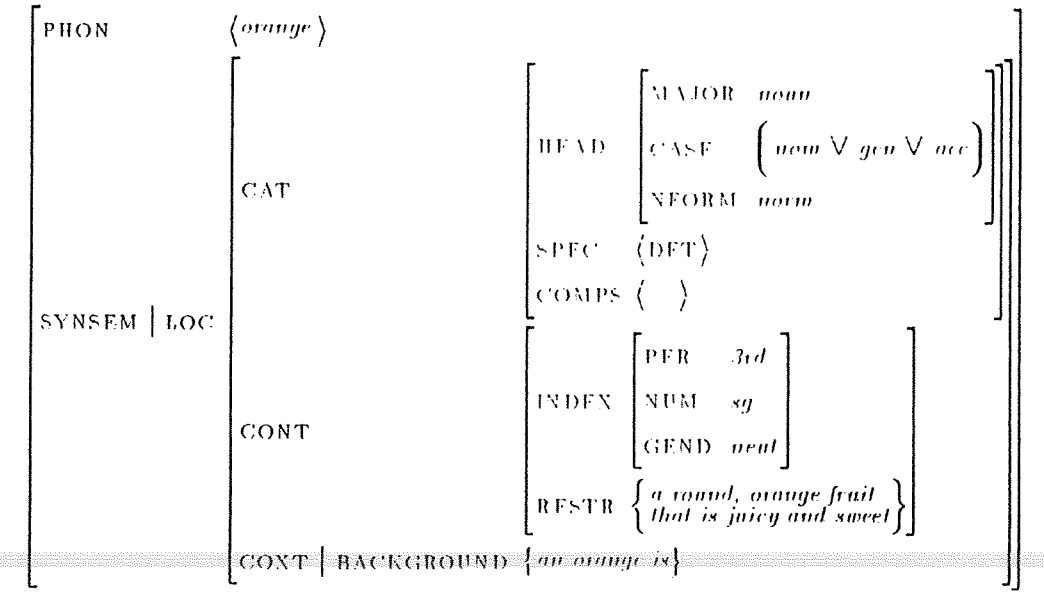
In the remainder of this section we will briefly show the integration of the semantic content of Cobuild entries for nouns and verbs in a general feature-style format of BLF.

3.1.1 Format for noun specifications

Consider now the following AVM, which depicts the general format of noun specifications:



To a large degree, this structure complies with the general framework of HPSG. The important divergence is depicted in the semantic part of the AVM, where we interprete the respective values differently. The purpose of this strategy may become clearer if you compare the above structure with the one below:



The major innovation is that we now find – as indicated in the latter AVM—the entire content of a lexical phrase within the semantic information domain of a single entry, where we interprete the relation between the SYNSEM | LOC | COXT | BACKGROUND and SYNSEM | LOC | CONT | RESTR values as being of implicational, i.e. *if ... then*, nature.

The textual items in the latter structure must be compared with the corresponding values of the functional parser output samples depicted in the first section. From these, the predicates—presented as feature substructures in the former AVM—will have to be extracted.

3.1.2 Format for Verb Specifications

The general structure of verb specifications in adapted BLF is shown below:

$$\left[\begin{array}{ll}
\text{PHON} & \langle \cdots \rangle \\
\text{SYNSEM} \mid \text{LOC} & \left[\begin{array}{ll}
\text{CAT} & \left[\begin{array}{ll}
\text{HEAD} & \left[\begin{array}{ll} \text{MAJOR} & \textit{verb} \\ \text{VFORM} & \left(\textit{bse.} \vee \textit{inf.} \vee \textit{fin.} \vee \ldots \right) \end{array}\right] \\
\text{SUBJ} & \langle \text{NP}[\textit{nom}]\!:\!\boxed{1} \rangle \\
\text{COMPS} & \langle \text{NP}[\textit{acc}]\!:\!\boxed{2}, \ldots, \text{NP}[\textit{acc}]\!:\!\boxed{n} \rangle
\end{array}\right] \\
\text{CONT} \mid \text{RESTR} & \left\{\left[\begin{array}{ll} \text{RELN} & \ldots \\ \text{ARG.1} & \boxed{1} \\ \text{ARG.2} & \boxed{2} \\ \vdots & \vdots \\ \text{ARG.}n & \boxed{n} \end{array}\right] \ldots \left[\begin{array}{ll} \text{RELN} & \ldots \\ \text{ARG.1} & \boxed{1} \\ \text{ARG.2} & \boxed{2} \\ \vdots & \vdots \\ \text{ARG.}n & \boxed{n} \end{array}\right]\right\} \\
\text{COXT} \mid \text{BACKGROUND} & \left\{\left[\begin{array}{ll} \text{RELN} & \ldots \\ \text{ARG.1} & \boxed{1} \\ \text{ARG.2} & \boxed{2} \\ \vdots & \vdots \\ \text{ARG.}n & \boxed{n} \end{array}\right]\right\}
\end{array}\right]
\end{array}\right]$$

Where NP[*nom*]:[1] will usually expand to

$$\left[\begin{array}{ll}
\text{SYNSEM} \mid \text{LOC} & \left[\begin{array}{ll}
\text{CAT} \mid \text{HEAD} & \left[\begin{array}{ll} \text{MAJOR} & \textit{noun} \\ \text{CASE} & \textit{nom} \\ \text{NFORM} & \textit{norm} \end{array}\right] \\
\text{CONT} \mid \text{INDEX} & \boxed{1}\left[\text{NUM} \quad \textit{sg}\right]
\end{array}\right]
\end{array}\right]$$

It differs from the noun specifications mainly in that it allows n-place predicates to account for transitive, ditransitive, and tritransitive verbs. Additionally, the syntactic argument positions are more complex than those of regular nouns. Please note that the semantic content of each syntactic argument is coindexed with the corresponding value of the semantic argument. As already indicated in the second section by the Pisa team, it will be necessary to investigate a possible translation of of the Cobuild grammar codes to categorial information in HPSG terms.

3.2 The Common HPSG Format

The common representational format currently under development in Bochum is being devised to ensure

- the possibility of evaluating on a common basis the different approaches of the three teams involved in the project.
- the reusability of findings in CEC grammars, which are based on HPSG
- a standardized and consistent – albeit with well defined deviations– framework as input to the Alep system

3.2.1 Noun Specifications

The following structure should be regarded as a kind of skeleton to which the differing informational items being supplied by the three sites could be attached. Three periods indicate the only places where information can and should diverge.

$$
\left[\begin{array}{ll}
\textsc{phon} & \langle \cdots \rangle \\
\cdots & \\
\textsc{synsem} \mid \textsc{loc} & \left[\begin{array}{ll}
\textsc{cat} & \left[\begin{array}{ll}
\textsc{head} & \left[\begin{array}{ll}
\textsc{major} & \mathit{noun} \\
\textsc{case} & \left(\mathit{nom} \vee \mathit{gen} \vee \mathit{acc}\right) \\
\cdots & \\
\textsc{nform} & \left(\mathit{norm} \vee \mathit{it} \vee \mathit{there}\right)
\end{array}\right] \\
\textsc{spec} & \langle \cdots \rangle \\
\textsc{comps} & \langle \cdots \rangle
\end{array}\right] \\
\textsc{cont} & \left[\begin{array}{ll}
\textsc{index} & \left[\begin{array}{ll}
\textsc{per} & \left(\mathit{1st} \vee \mathit{2nd} \vee \mathit{3rd}\right) \\
\textsc{num} & \left(\mathit{sg} \vee \mathit{pl}\right) \\
\textsc{gend} & \left(\mathit{fem} \vee \mathit{masc} \vee \mathit{neut}\right)
\end{array}\right] \\
\textsc{restr} & \{\cdots\} \\
\cdots &
\end{array}\right] \\
\textsc{cont} \mid \textsc{background} & \{\cdots\}
\end{array}\right]
\end{array}\right]
$$

Consider now the two following attributes, which have been suggested by the Pisa team in order to account for specific information supplied by Cobuild entries:

$$\left[\text{SYNSEM} \mid \text{LOC} \mid \text{CAT} \mid \text{HEAD} \left[\begin{array}{ll} \text{PROP} & \left(+ \vee - \right) \\ \text{COUNT} & \left(cnt \vee uncnt \vee mass \right) \end{array}\right]\right]$$

In case this information is employed by one of the partners or is agreed upon as a common substructure, it would expand the skeleton – which up to this position is designed in strictly canonical HPSG format! – as follows:

$$\left[\begin{array}{ll} \text{PHON} & \langle \cdots \rangle \\ \cdots & \\ \text{SYNSEM} \mid \text{LOC} & \left[\begin{array}{ll} \text{CAT} & \left[\begin{array}{ll} \text{HEAD} & \left[\begin{array}{ll} \text{MAJOR} & noun \\ \text{CASE} & \left(nom \vee gen \vee acc \right) \\ \text{PROP} & \left(+ \vee - \right) \\ \text{COUNT} & \left(cnt \vee uncnt \vee mass \right) \\ \text{NFORM} & \left(norm \vee it \vee there \right) \end{array}\right] \\ \text{SPEC} & \langle \ \rangle \\ \text{COMPS} & \langle \ \rangle \end{array}\right] \\ \text{CONT} & \left[\begin{array}{ll} \text{INDEX} & \left[\begin{array}{ll} \text{PER} & \left(1st \vee 2nd \vee 3rd \right) \\ \text{NUM} & \left(sg \vee pl \right) \\ \text{GEND} & \left(fem \vee masc \vee neut \right) \end{array}\right] \\ \text{RESTR} & \{\cdots\} \\ \cdots & \end{array}\right] \\ \text{CONT} \mid \text{BACKGROUND} & \{\cdots\} \end{array}\right] \\ \cdots & \end{array}\right]$$

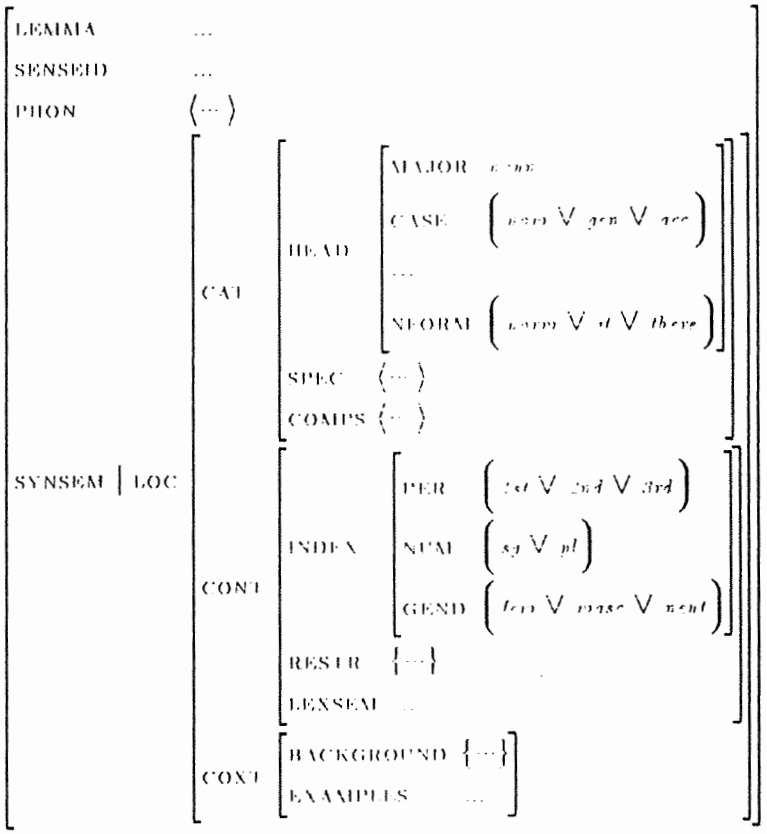
Since the core format remains unchanged, future developers of HPSG-based input grammars for Alep will have the option of choosing between at least two separate styles of semantic analysis, or even a combination of them. Depending on how much (morpho)syntactic information like the above can be extracted from Cobuild codes in the course of the project, several options might become available there, too. The Pisa team, for example, translated a complex attribute ORTHPHON from Cobuild information:

$$\begin{bmatrix}\text{PHON} & \langle\cdots\rangle \\ \text{ORTHPHON} & \begin{bmatrix}\text{PHON} & \begin{bmatrix}\text{BASEPHON} & \ldots \\ \text{VARPHON} & \begin{bmatrix}\text{VARIANTS} & \ldots \\ \text{ENG\_VAR} & \left(\mathit{british} \vee \mathit{american} \vee \mathit{irish}\right)\end{bmatrix}\end{bmatrix} \\ \text{SPELL} & \begin{bmatrix}\text{SPELLFORM} & \ldots \\ \text{VARSPELL} & \begin{bmatrix}\text{VARIANTS} & \ldots \\ \text{ENG\_VAR} & \left(\mathit{british} \vee \mathit{american} \vee \mathit{irish}\right)\end{bmatrix}\end{bmatrix}\end{bmatrix}\end{bmatrix}$$

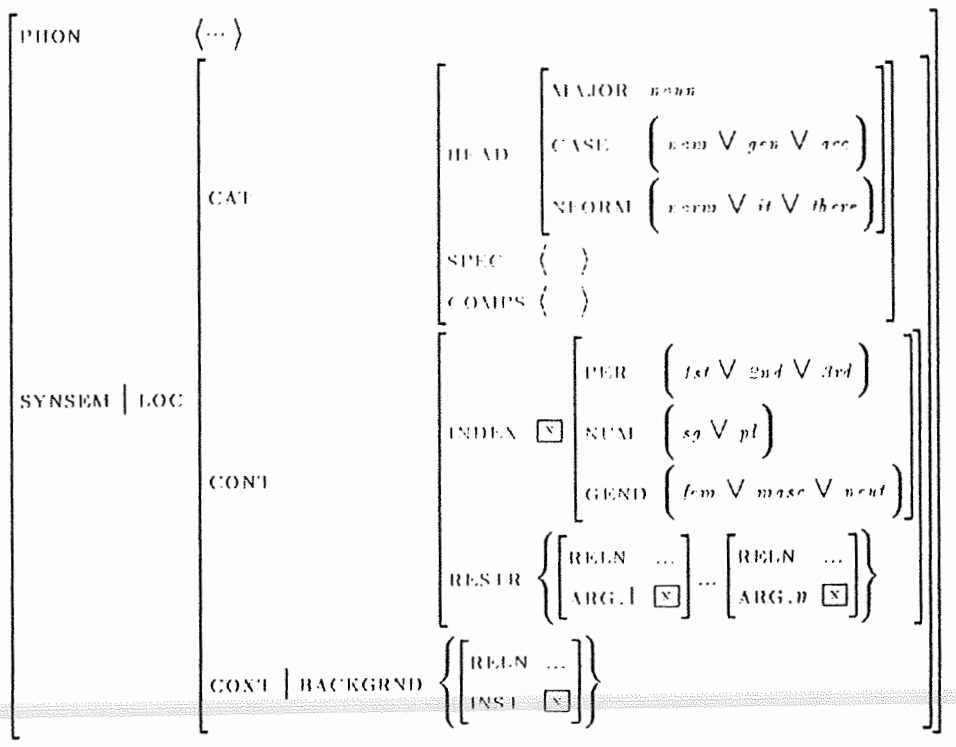
which could expand in the skeleton as follows:

$$\begin{bmatrix}\text{PHON} & \langle\cdots\rangle \\ \text{ORTHPHON} & \begin{bmatrix}\text{PHON} & \begin{bmatrix}\text{BASEPHON} & \ldots \\ \text{VARPHON} & \begin{bmatrix}\text{VARIANTS} & \ldots \\ \text{ENG\_VAR} & \left(\mathit{british} \vee \mathit{american} \vee \mathit{irish}\right)\end{bmatrix}\end{bmatrix} \\ \text{SPELL} & \begin{bmatrix}\text{SPELLFORM} & \ldots \\ \text{VARSPELL} & \begin{bmatrix}\text{VARIANTS} & \ldots \\ \text{ENG\_VAR} & \left(\mathit{british} \vee \mathit{american} \vee \mathit{irish}\right)\end{bmatrix}\end{bmatrix}\end{bmatrix} \\ \text{SYNSEM} \mid \text{LOC} & \begin{bmatrix}\text{CAT} & \begin{bmatrix}\text{HEAD} & \begin{bmatrix}\text{MAJOR} & \mathit{noun} \\ \text{CASE} & \left(\mathit{nom} \vee \mathit{gen} \vee \mathit{acc}\right) \\ \text{NFORM} & \left(\mathit{norm} \vee \mathit{it} \vee \mathit{there}\right)\end{bmatrix} \\ \text{SPEC} & \langle\cdots\rangle \\ \text{COMPS} & \langle\cdots\rangle\end{bmatrix} \\ \text{CONT} & \begin{bmatrix}\text{INDEX} & \begin{bmatrix}\text{PER} & \left(\mathit{1st} \vee \mathit{2nd} \vee \mathit{3rd}\right) \\ \text{NUM} & \left(\mathit{sg} \vee \mathit{pl}\right) \\ \text{GEND} & \left(\mathit{fem} \vee \mathit{masc} \vee \mathit{neut}\right)\end{bmatrix} \\ \text{RESTR} & \{\cdots\}\end{bmatrix} \\ \text{CONT} \mid \text{BACKGROUND} & \{\cdots\}\end{bmatrix}\end{bmatrix}$$

In the case of Pisa, the BACKGROUND attribute would be left unspecified, while the paths of the attributes LEMMA, SENSEID, LEXSEM, EXAMPLES (cp. Section 2) would have to be re-directed in order to keep the place of the remaining information in line with the common interface:



Bochum would store the extracted information in different places, but also without changing the core structure:



Please note that, as indicated in the preceding section, the BACKGROUND and RESTR values are specified here in opposition to the approach chosen by the Pisa team, who will store the bulk of the semantic information in the LEXSEM value range.

3.2.2 Verb Specifications

Verb specifications follow the same procedure as nouns. Recall that the expansion of the empty slots in the case of the Bochum analysis will again involve the specification of the BACKGROUND and RESTR attributes, as shown below.

$$\begin{bmatrix} \text{PHON} & \langle \cdots \rangle \\ \text{SYNSEM} \mid \text{LOC} & \begin{bmatrix} \text{CAT} & \begin{bmatrix} \text{HEAD} & \begin{bmatrix} \text{MAJOR} & \textit{verb} \\ \text{VFORM} & \left(\textit{bse.} \vee \textit{inf.} \vee \textit{fin.} \vee \ldots \right) \end{bmatrix} \\ \text{SUBJ} & \langle \cdots \rangle \\ \text{COMPS} & \langle \cdots \rangle \end{bmatrix} \\ \text{CONT} \mid \text{RESTR} & \{ \cdots \} \\ \text{COXT} \mid \text{BACKGRND} & \{ \cdots \} \end{bmatrix} \end{bmatrix}$$

$$\begin{bmatrix} \text{PHON} & \langle \cdots \rangle \\ \text{SYNSEM} \mid \text{LOC} & \begin{bmatrix} \text{CAT} & \begin{bmatrix} \text{HEAD} & \begin{bmatrix} \text{MAJOR} & \textit{verb} \\ \text{VFORM} & \left(\textit{bse.} \vee \textit{inf.} \vee \textit{fin.} \vee \ldots \right) \end{bmatrix} \\ \text{SUBJ} & \langle \text{NP}[\textit{nom}]:\boxed{1} \rangle \\ \text{COMPS} & \langle \text{NP}[\textit{acc}]:\boxed{2}, \ldots, \text{NP}[\textit{acc}]:\boxed{n} \rangle \end{bmatrix} \\ \text{CONT} \mid \text{RESTR} & \left\{ \begin{bmatrix} \text{RELN} & \ldots \\ \text{ARG.1} & \boxed{1} \\ \text{ARG.2} & \boxed{2} \\ \vdots & \vdots \\ \text{ARG.}n & \boxed{n} \end{bmatrix} \ldots \begin{bmatrix} \text{RELN} & \ldots \\ \text{ARG.1} & \boxed{1} \\ \text{ARG.2} & \boxed{2} \\ \vdots & \vdots \\ \text{ARG.}n & \boxed{n} \end{bmatrix} \right\} \\ \text{COXT} \mid \text{BACKGROUND} & \left\{ \begin{bmatrix} \text{RELN} & \ldots \\ \text{ARG.1} & \boxed{1} \\ \text{ARG.2} & \boxed{2} \\ \vdots & \vdots \\ \text{ARG.}n & \boxed{n} \end{bmatrix} \right\} \end{bmatrix} \end{bmatrix}$$