6 Constraint-based Specification

Due to its interdisciplinary nature, the realm of natural language processing has been addressed by a community of researchers from diverse scientific disciplines and research traditions, using several different formal and computational tools, and aiming at various development purposes and research goals. More often than not, it turns out to be quite hard to compare or combine results on the same or adjacent topics, which may be perfectly confluent, but whose methodological underpinnings may give them a totally disparate appearance and impact.

Nevertheless, this active and reciprocal exposure to different approaches has also contributed to an emerging trend towards integrative frameworks. A remarkable success in pursuing such desideratum has been achieved with the HPSG environment. Since its inception in the late eighties, it has evolved to become, according to Uszkoreit (1996), the "single most influential framework" in basic research in natural language processing. Accordingly, this is the language processing framework in the context of which our study on the specification and implementation of binding constraints will be developed in the following pages.

Having clarified in previous chapters the proper place of binding constraints in the architecture of grammar levels, and how these constraints are represented at the Semantics level, we aim now to establish their full integration into grammar.

6.1 The Grammatical Knowledge and Processing Framework

Feature-, unification- or constraint-based

HPSG is the acronym for Head-driven Phrase Structure Grammar, which is also the title of the second of two seminal books by Carl Pollard and Ivan Sag, (Pollard and Sag, 1987) and (Pollard and Sag, 1994). This framework was set up on the basis of cardinal ideas and tools from linguistics and computer science. Its design is the result of an ingenious blend of influences from data type theory, knowledge representation, unification-based formalisms, object-oriented programming, and several non-derivational research traditions in natural language syntax, such as categorial grammar, generalized phrase-structure grammar, arc-pair grammar and lexical-functional grammar. It is a mature achievement of research on feature-based, unification-based or, more generally, constraint-based natural language processing (vd. Shieber, 1986, Uszkoreit, 1990, 1993 for an introduction and overview), based on the seminal work on Functional Unification Grammar by Martin Kay in the late seventies (vd. Kay, 1984, for the rationale, and Netter, 1996, Ch.1, for a comprehensive historical perspective). References of works on logical, computational, linguistic and cognitive issues in the HPSG framework are collected at the HPSG bibliography site, www.dfki.de/lt/HPSG; on browsing this site, it is possible to gain an idea of the intense research activity involved in this area.

HPSG is a precise but flexible interleaving of proposals concerning the shape of different conceptual and methodological layers in the modeling of natural language grammars. Although these layers were designed to be consistently integrated into a single framework, they are quite autonomous. To a considerable extent, each one of them may undergo changes and be experimented with without compromising its inclusion within the whole framework.

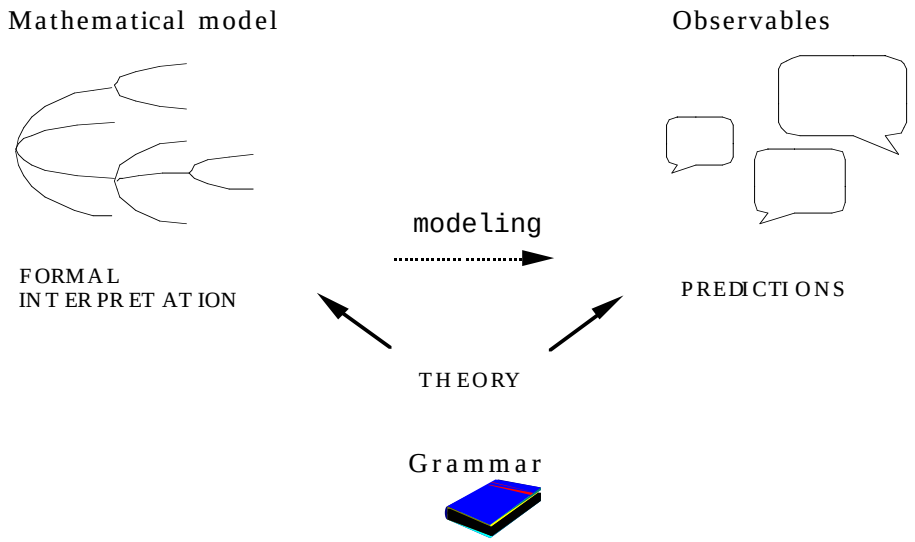
We will now provide a description of HPSG by examining the major features of each of these layers.

Ontological setup

As in other empirical theories, in HPSG the theory and empirical phenomena it refers to are mediated by a mathematical structure that constitutes a model of the empirical domain over which the theory is supposed to unfold its predictions. This model structure is put in correspondence with the relevant observables so that entities and relations in the empirical domain are represented by entities and relations in the model. This provides a convenient setup for rigorous and falsifiable theorizing as well as a suitable basis for improving results and progress in research. Therefore, with such a tripartite ontological setup, as sketched in (1), the theory can be seen either as talking about the formal model, or as being interpreted therein.

In the research on natural language grammar the relevant observables are assumed to be types of natural language expressions and their subparts. The observables are mathematically modeled by a system of sorted feature structures, which are graph theoretic entities. The theory, in turn, is a formal specification interpreted in that modeling domain. The constraints of the specification establish predictions in the sense that they define what objects from the domain are admissible as belonging to the natural language in question and those that are not.

(1)



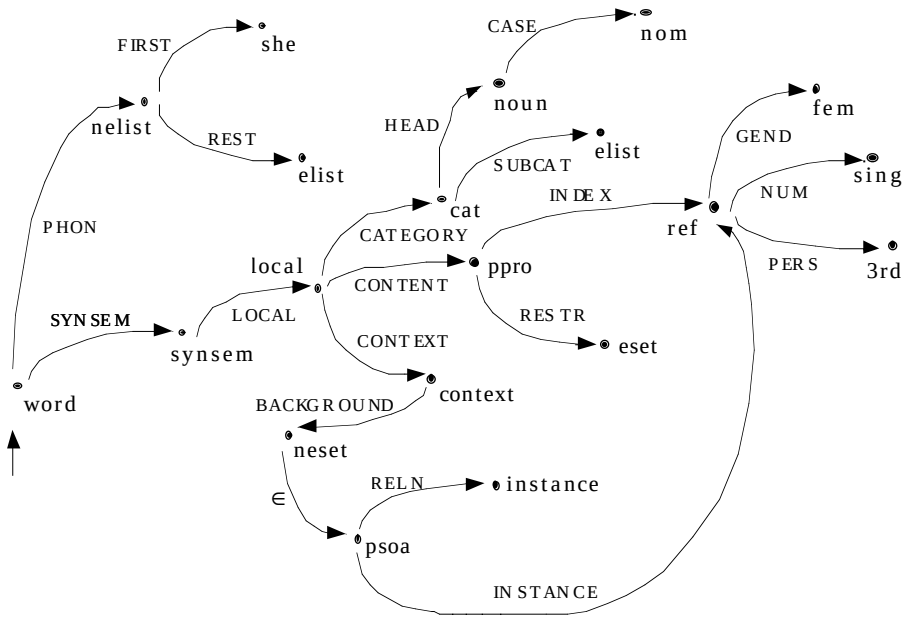
Mathematical model

A feature structure is a labeled acyclic graph such that nodes are tagged with sort labels and arcs with attribute labels. Accordingly, a non-atomic feature structure of sort s – a graph whose top node is labeled s – is said to have attributes – those with which the arcs leaving node s are tagged. An attribute a in turn is said to have a value, which is another (atomic or not) feature structure to which the arc labeled with a is directed. An example of a feature structure corresponding to the word she is depicted in (2).

Given that they should be total models of linguistic objects, feature structures are required to be (i) totally well-typed, and (ii) sort resolved. In informal terms, this means that: (i) for each sort s of the graph, every arc/attribute that has s as its source, i.e. which is appropriate to "characterize" s, is actually present; (ii) every node is assigned a sort s that is most specific in the sort hierarchy in which s possibly enters.

A crucial property of feature structures is that two distinct paths in the graph can lead to one and the same node. This means that structure sharing is allowed, i.e. that two paths in the graph can share the same structure as their common value.

(2)



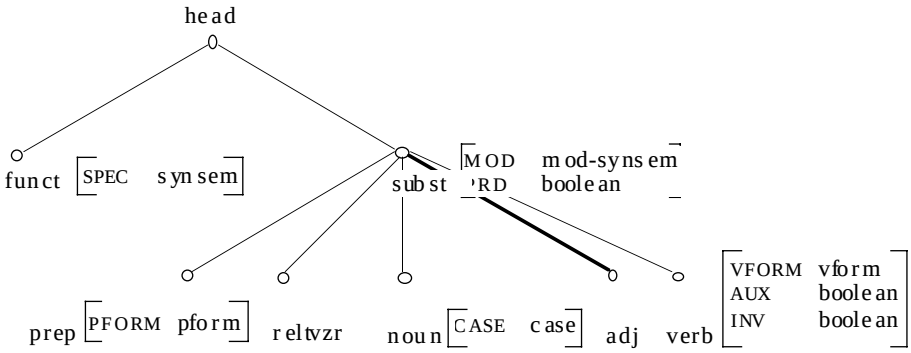
Grammar

From an algebra-theoretic point of view, a theory/grammar in HPSG is a specification consisting of a signature and a presentation.

The signature defines what data types are allowed as possible pieces of structure encoding linguistic information. It includes a sort hierarchy and an appropriateness definition. The sort hierarchy is a partial order of sorts where the possible types of linguistic entities and their subtypes are fixed. The appropriateness definition, in turn, states what the characteristics of each sort are in the hierarchy. This is done by associating with sorts constraints that establish what the appropriate feature structures are for the type of objects the sorts correspond to. The sort hierarchy is an inheritance taxonomic tree such that a sort inherits the appropriateness constraints of its supersorts.

The diagram in (3) depicts an example of a sort hierarchy and associated appropriateness conditions (the syntax of appropriateness conditions is explained below) the top node of which is the sort head.

(3)



The presentation in turn is a set of implicative constraints which are interpreted as being true or false in the model domain. They can also be seen as descriptions that constrain the set of feature structures that, in accordance with the data types defined in the signature, are predicted as being admissible by the theory as well formed expressions of the language at stake.

Description language

The constraints of the theory/grammar are stated using a special purpose language whose expressions are known as attribute-value matrix diagrams (AVMs). In a rough presentation of its syntax, one could say that non-atomic AVMs comprise two column matrices, where the first column displays the attributes and the second column the corresponding values. The values of attributes are AVMs. Matrices receive a left subscript which indicates its sort, and an atomic AVM just consists of an atomic sort:

(4)

$$_{sort}\begin{bmatrix} \text{ATTRIBUTE 1} & \text{AVM 1} \\ \ldots & \ldots \\ \text{ATTRIBUTE } n & \text{AVM } n \end{bmatrix}$$

There is a special symbol for stating structure sharing. When two attributes exhibit the same tag, the same boxed numeral, that means their values are token-identical. Tags may be followed by AVMs, which state the value the tag stands for.

Descriptions of sets are given within curly braces and descriptions of lists are abbreviated by using the angle-bracket notation.

It is worth noting that since AVMs express constraints on the model domain, i.e. they are descriptions of admissible (total) model objects, they may be partial descriptions of the object to which characterization they contribute.

Principles and lexical entries

Usually, the constraints stated in the presentation are termed, depending on the type of linguistic information they convey, as grammatical principles or lexical entries.

The grammar of a specific language includes a set of grammatical principles that are claimed to belong also to the grammar of any other natural language. These are known as Universal Grammar (UG) principles. An example of an implicative constraint of this kind can be found below in (5). It depicts the Head Feature Principle, which states that if a phrase is the projection of a head, then

CAT|HEAD value of the phrase is identical to CAT|HEAD value of its head (the linguistic significance of the different attributes is clarified below).

(5)

$$_{phrase}\begin{bmatrix}\text{DTRS} & headed-struc\end{bmatrix} \rightarrow \begin{bmatrix}\text{SYNSEM|LOC|CAT|HEAD} & \boxed{1} \\ \text{DTRS|HEAD-DTR|SYNSEM|LOC|CAT|HEAD} & \boxed{1}\end{bmatrix}$$

Besides the UG principles, each grammar for a specific language has principles which hold only for that language or for a subset (family) of languages to which that language belongs. All these principles are conjunctive constraints as they enter the grammar as a conjunction of constraints to which any well formed expression must comply with.

On a par with the set of conjunctive principles, the constraints corresponding to lexical entries form the set of disjunctive principles. Below a partial representation of the lexical sign corresponding to the pronoun she is depicted, which is a partial description in AVM format of the feature structure in (2):

(6)

$$_{word}\begin{bmatrix}\text{PHON} & <\int i:> \\ \text{SYNSEM} & _{synsem}\begin{bmatrix}\text{LOC} & _{local}\begin{bmatrix}\text{CATEGORY} & _{cat}\begin{bmatrix}\text{HEAD} & _{noun}[\text{CASE} \quad nom] \\ \text{SUBCAT} & <>\end{bmatrix} \\ \text{CONTENT} & _{ppro}\begin{bmatrix}\text{INDEX} & _{ref}[...] \\ \text{RESTR} & \{\}\end{bmatrix} \\ \text{CONTEXT} & _{context}\begin{bmatrix}\text{BACKGR} & \{_{psoa}[...]\}\end{bmatrix}\end{bmatrix}\end{bmatrix}\end{bmatrix}$$

Disjunction may also be introduced by certain grammatical constraints which are then termed schemata, usually grouped under a single principle.

A synopsis of the implicative constraints according to their different classes is given below:

(7) Language L = UG Λ P_{n+1} Λ ... Λ P_{n+m} Λ (($S_1 \vee ... \vee S_p$) $\vee$ $L_1 \vee .... \vee L_q$)

Where:

UG = P_1 Λ ... Λ P_n and P_1 Λ ... Λ P_n are universal principles

P_{n+1} Λ ... Λ P_{n+m} are language L specific principles

($S_1 \vee ... \vee S_p$) are schemata grouped in language L specific principles

$L_1 \vee .... \vee L_q$ are lexical signs of L (basic or the output of lexical rules)

Finally, the constraints known as lexical rules include constraints expressing generalizations over lexical signs. A lexical rule is a pair of two

"meta-descriptions" since it relates lexical signs (constraints) and not the objects described by ordinary constraints. Once the lexical rule applies, the constraint specified on the left hand side picks out the lexical signs whose structure comply with it and these signs are overwritten with the additional constraints specified on the right hand side of the rule. Below a lexical rule is exemplified, which might be taken as the basis for a lexicalist account of passive constructions:

(8)

$$_{word}\left[\text{SUBCAT} \quad \left\langle \boxed{1}, \boxed{2}, \ldots \right\rangle\right] \quad \mapsto \quad \left[\text{SUBCAT} \quad \left\langle \boxed{2}, \ldots, \text{PP}[by]_{\boxed{1}} \right\rangle\right]$$

Linguistic configuration

With the ontological setup and description formalism in place, the linguistic configuration of the grammars for particular languages can now be addressed. By linguistic configuration, we mean the set of formal options concerning the accommodation of linguistic generalizations that involve issues common to all languages and to most of the different linguistic phenomena and constructions.

In HPSG, a sign-based approach to linguistic constraints was adopted, where the different aspects of the linguistic objects at stake, e.g. phonological, syntactic, semantic and pragmatic, are described in a single representation. This makes HPSG a monostratal linguistic framework. No grammatical principle or lexical entry has precedence over any other in terms of constraint satisfaction, and a linguistic object may be described by one single expression of the description language which integrates the outcome of all different analytical levels of linguistic theorizing.

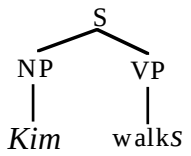
Subcategorization information is fully lexicalized in the relevant predicator and the subcategorizing of syntactic arguments occurs via argument cancellation in the style of categorial grammar.

Valence alternations, as in passive constructions, are given a lexicalized account where lexical rules are responsible for stating the relevant generalization involving relations between lexical items.

Phrase constituency is factored out in immediate dominance and linear precedence relations in view of a general account of languages with and without free word order. Moreover, the syntactic constituency is rendered not as a tree, but it is fully encoded in terms of feature structures where the class of grammatical

function of the phrases at stake is specified (head-daughter, complement-daughters, etc.). The attribute DTRS (for "daughters") of a phrase has as value a feature structure where the signs of the relevant daughters nodes are encoded. For instance, a phrase such as Kim walks, whose syntactic structure is typically represented in terms of a constituency tree as in a. below, receives a constraint-based account in HPSG as in b.:

(9) a.



b.

$$\begin{bmatrix} \text{PHON} & \langle Kim,\ walks \rangle \\ \text{DTRS} & \begin{bmatrix} \text{HEAD-DTR} & \begin{bmatrix} \text{PHON} & \langle walks \rangle \\ \text{SYNSEM} & VP \end{bmatrix}_{phrase} \\ \text{COMP-DTRS} & \left\langle \begin{bmatrix} \text{PHON} & \langle Kim \rangle \\ \text{SYNSEM} & NP \end{bmatrix}_{phrase} \right\rangle \end{bmatrix}_{head-comp-struc} \\ \text{SYNSEM} & \ldots \end{bmatrix}_{phrase}$$

As for unbounded syntactic dependencies, the relationship between the syntactic gap and its filler is seen as a matter of structure sharing, on a par with a thread-based approach to such non-local dependencies.

The importance of structure sharing, however, is not restricted to the account of unbounded dependencies. Following Pollard and Sag, 1994, p.19, it is worth noting that "it is not going too far to say that in HPSG structure sharing is the central explanatory mechanism [...] Indeed, the relationships between fillers and traces, between "understood" subjects and their controllers, between pronouns and their antecedents, between "agreement sources" and "agreement targets", and between the category of a word and the category of its phrasal projections are all analyzed as instances of structure sharing".

Finally, although the original semantic component was designed by Pollard and Sag in the spirit of situation semantics, Frank and Reyle (1995) have shown that an underspecified, principle-based semantics inspired on DRT could be better integrated into the overall sign-based philosophy of HPSG grammars. We will refer to this in more detail in the next subsection.

Language specific theories

Given this ontological and linguistic setup, Pollard and Sag in their second book designed a grammar for a substantial fragment of English. That grammar covers core phenomena such as phrase structure, complementation, agreement and interpretation, as well as some constructions that are central in the linguistic debate and to a certain extent form a benchmark for checking the adequacy and explanatory potential of linguistic frameworks, such as relative clauses, unbounded dependency constructions, complement control and binding. Given the high level of descriptive economy and formal rigor provided by the HPSG framework, it was possible to state the whole grammar for that fragment of English in the nine pages of the Appendix of Pollard and Sag, 1994, while the remaining four hundred pages of the book were used to document the thoroughness of the approach and to check the empirical adequacy of the proposals about specific linguistic phenomena and constructions.

Given the grammar architecture described above, involving language specific as well as universal principles, the grammar fragment designed by Pollard and Sag embodies a number of principles which, with convenient parametric adaptation for each language, may be taken as accounting for essential aspects of natural languages. It is not feasible to present, even in an abridged formulation, the core of that grammar within the limits of the present section. Nevertheless, for the sake of providing a brief idea of the basic structuring of linguistic information proposed in HPSG, we will discuss the AVM of the pronoun she in (6), whose feature structure is in (2), and comment on its subparts.

The PHON value encodes a phonological representation of she. The SYNSEM value of a sign, whose subsort in this example is word, encodes the information that can be subcategorized by a predicator. The feature structure of LOC value describes the information shared between a trace and its filler. The value of CATEGORY renders information on syntactic category and subcategorization frame. CONTENT and CONTEXT are reserved, respectively, for semantic and pragmatic information. Finally, the DTRS value – not present in the AVM for the word she in (6), but present in (9)b. in the AVM for the phrase Kim walks – retains information on the constituent structure of non-lexical signs.

Principle-based semantics

As mentioned above, we will adopt a slightly revised version of the linguistic configuration proposed by Pollard and Sag (1994). While the original HPSG framework comes equipped with a type of semantic representation designed in accordance with the basic tenets of Situation Semantics theory (Cooper et al., 1990), and incorporates a quantifier storage approach to scope ambiguities (cf. Cooper (1983)), the type of semantic representation we adopt is inspired in DRT. It follows the proposal of Frank and Reyle (1995), whose core insights, however, have a wide range of justifications and implications, not restricted to merely replacing a situation theory-based representation by a DRT-based one.

A first notorious aspect of this proposal is that a quite modular view of syntax and semantics is adopted, where "syntax as well as semantics provide structures of equal right, that the principles internal to the syntactic and semantic level are motivated only by the syntactic and semantic theory, respectively" (p.10).

Second, the semantic representation language is designed so that it allows for an underspecified representation of meaning, namely by permitting a single expression of the representation language to possibly encode several different readings of the corresponding natural language expression.

Third, while the construction of semantic representation is on a par with the construction of syntactic representation, the interface between syntax and semantics is conceived as a source of conditions that, together with conditions with other origins – morphological, pragmatic, and so on –, induces a monotonic specification of the interpretation options left open by the underspecified representation.

The underspecification language for semantic representation is imported from Reyle, 1993a, where an Underspecified Discourse Representation Theory (UDRT) is defined, for which a proof theory was developed by Reyle (1993b). The underspecification technique adopted in UDRT builds on the subordination relation between DRSs.

In simple but explanatory terms, a box/DRS is subordinated to another box/DRS if the first is pictorially inside the latter, and the different boxes inside the outermost DRS K can be seen as forming a tree under the subordination relation whose top node is K. The basic idea of UDRT is to relax the subordination relation between boxes so that the resulting underspecification in the expressions of the semantic representation language correlates with ambiguity in the corresponding natural language expressions. Accordingly, the different boxes/DRSs making up a single UDRS are assigned uniquely identifying labels and the representation language is enriched with an annotation schema that keeps record of the partial order of these

labels. The boxes/DRSs can now be seen as forming a lattice, and the process of interpretation can be conceived as a process of monotonically adding further specifications thus reducing the degree of ambiguity represented by the UDRSs in question.

The integration of UDRT-based representation language into HPSG proposed by Frank and Reyle (1995) involves replacing the value of CONTENT as proposed in Pollard and Sag, 1994 by the feature structure depicted below:

(10)

$$_{udrs}\begin{bmatrix} \text{LS} & \begin{bmatrix} \text{L}-\text{MAX} & \ldots \\ \text{L}-\text{MIN} & \ldots \end{bmatrix} \\ \text{SUBORD} & \{\ldots\} \\ \text{CONDS} & \{\ldots\} \end{bmatrix}$$

This feature structure encodes the three types of information that define an UDRS: what its subUDRSs are – given by CONDS value, which is a set of labeled conditions –; what the partial order is between their labels – given by SUBORD value, which is a set of subordination relations between labels –; and what the top and bottom labels are of the outermost UDRS at stake – given by LS-MAX and LS-MIN values.

On a par with this reshuffling of the HPSG feature geometry concerning semantics, Frank and Reyle (1995) proposed a principle with several clauses, termed Semantics Principle, that ensure the correct construction of UDRSs. There is a clause for constructing the set of conditions in CONDS value (Clause I); there is a clause for determining what are the top and bottom labels are in LS value (Clause III); and three other clauses on scope taking DRSs that partially determine the value of SUBORD (Clauses II, IV and V). These clauses are compiled below in an informal style:

(11) Semantics Principle

Clause I: CONDS value of a sign is the union of CONDS values of the daughters.

Clause II: SUBORD value of a sign is defined by the union of SUBORD values of the daughters.

Clause III: The minimal and maximal labels of the head are available all along the head projection: L-MAX and L-MIN values of the sign are L-MAX and L-MIN values of the head daughter.

Clause IV: The minimal label of verb condition is subordinated to the minimal labels of its arguments conditions.

Clause V: If an argument is characterized as a scope bearing argument, its maximal label is subordinate to the maximal label of the local domain.

6.2 A Module of Linguistic Knowledge at Odds in Grammar

Having presented the framework for the grammatical knowledge we adopted, we can turn now to the accommodation of binding constraints in grammar. The goal of the present section is to understand to what extent it is possible to build on existing proposals in the literature in order to fully integrate these constraints not only in semantics but also in the grammar of natural languages.

In HPSG: the "informal" module

If the structure of the second book on HPSG by Pollard and Sag is inverted by bringing the Appendix to the fore, one finds a rather different approach for presenting and discussing HPSG. The reader discovers in Pollard and Sag, 1994, a seven-page grammar documented with a four hundred page development report that can be found in the nine chapters of the book preceding that grammar.

As noticed above, an interesting feature of that grammar fragment is that it appears as a fully explicit and formal presentation of the linguistic phenomena dealt with in the remainder of the book – the exercise of writing the principles stated in the Appendix in terms of AVMs is undertaken in Rieheman, 1995. Another interesting point is that this fragment shows in practice the value of having a constraint-based framework as the one proposed in HPSG: the four hundred pages of grammatical analysis can be condensed in a seven page grammar which can be tested for the empirical adequacy of its predictions and can easily receive a

corresponding computational implementation (cf. chapter 8 Computational Implementation).

With respect to binding theory, developed in chapter 6 of Pollard and Sag, 1994 – the principles of which are repeated below together with the fourth binding principle –, a somewhat surprising situation occurs. It has passed quite unnoticed in the literature that the set of binding constraints is the only module of grammar knowledge not encoded in terms of HPSG description language, nor is it clearly integrated into the grammar of the referred Appendix.

(12) Binding Theory

Principle A
A locally o-commanded short distance reflexive must be locally o-bound.

Principle Z
An o-commanded long distance reflexive must be o-bound.

Principle B
A pronoun must be locally o-free.

Principle C
A nonpronoun must be o-free.

While this is something readily acknowledged by the authors (cf. Ivan Sag p.c.), it is possible to find a couple of works that, even though they do not put forward such an implementation, elaborate on this question.

In his dissertation, Bredenkamp aimed at studying the issues involved in the specification of binding theory in the HPSG framework (Bredenkamp, 1996). To the best of our knowledge, he is the first one to notice that "the formal devices which the grammar uses to evaluate, for example o-binding, are not made explicit" (p.93) in the HPSG book. The research path he followed consisted basically in trying to import binding theory into the HPSG framework in the format provided in Lexical Functional Grammar (Dalrymple, 1993) and in Government and Binding framework (in the version of Reinhard and Reuland (1993)). His findings were very interesting in the sense that the many negative results collected helped to make evident that the question as to how to integrate binding constraints in HPSG is not a trivial one, and moreover that this is an important open research issue.

The other place where the problem of implementing Pollard and Sag's (1994) binding theory in HPSG is addressed is the EAGLES report by Backofen et al. (1996). This work is a commissioned report whose goal is to provide an overview of "current developments in grammar formalisms, and closely related issues as the underlying grammatical theories, mathematical foundations, implemented grammars and development environments" (p.5). Section 2.6 of Chapter 3 of that report has a part devoted to the implementation of binding theory in HPSG. It is

worth noting two interesting remarks given here (p.65). The first concerns principle B: "the problem here is that there is no apparent way to get rid of the universal quantification, as lack of coindexation must hold for all the elements locally o-commanding [the pronoun]. In the absence of such a universal quantification no reasonable implementation of principle B is possible". The second remark regards principle C: "it seems to be impossible to implement principle C within a declarative view of a framework for HPSG. Indeed to check if something is o-free amounts to span a potentially unbounded structure negating all the possible coindexations".

The notorious suggestion that these remarks seem to be leading us to is that, in the eyes of their authors, some kind of essential limitation of the description language of HPSG has been reached. This implies that binding is perhaps a kind of phenomena whose properties reveal that the HPSG formalism is not fully adequate for specifying linguistic knowledge in its entirety (a suggestion our proposal will seek to contradict in the following sections).

Be that as it may, the interesting aspect to stress is that, despite very intensive research in all areas of grammar and different types of phenomena in the framework of HPSG, a module as central as binding theory is still waiting for an appropriate account in terms of a constraint-based grammar approach, in general, and for an appropriate integration into HPSG, in particular.

In GB: the post parsing module

Given this negative state of affairs, it will certainly be interesting to take a look at the status of binding theory in other prominent grammatical frameworks. Starting with the Government and Binding (GB) framework (Chomsky, 1981, 1986), we may notice that in spite of the many different proposals that have been discussed in GB literature concerning the exact specification of binding constraints and the grammar level (S-Structure, NP-Structure,...) at which they should hold, its basic underlying processing schema formulated in early eighties remained quite stable.

Elaborating on a previous discussion published in the Appendix of Chomsky, 1980, Chomsky (1981, pp.186-187) put forward some suggestions concerning the underlying procedures for integrating binding constraints into grammar. Interestingly, this proposal reveals that on a par with the specification of linguistic generalizations in terms of binding principles and accessory notions (indices, c-command, "Governing Category", "SUBJECT", "Accessibility", and the like), the integration of binding theory in GB grammars supposes a commitment to

non-trivial extra-grammar processing issues. When compiled in a systematic fashion, these suggestions form an algorithm with the following outline:

(13) After parsing has been completed:

(i) iteration:	repeat (ii)-(iii) until all possible different assignments of indices have been exhausted.
(ii) indexation:	assign indices to NPs;
(iii) filtering:	store the annotated tree if the indexation of NPs respects binding constraints, otherwise delete it.

Skipping issues related to the practical implementation of this algorithm, which will be addressed below, it is worth noticing that it embodies a processing schema for the linguistic module of binding constraints that is problematic on several counts.

First, it envisages binding constraints as post-parsing filters, thus somehow assigning them, at least from a processing point of view, a place outside the grammar proper. The linguistic knowledge embodied in a grammar, binding constraints aside, is called for to govern the process of parsing. Then, when parsing is completed, the output tree is used to produce as many other trees as there are different assignments of indices to the various NPs of the output tree. The set of these post-parsing indexed trees is then used as the search space. The job of binding theory will then be to provide criteria for partitioning that search space into two sets of indexed trees: the set of those trees whose indexing observes every binding constraint, and the set of those trees whose indexing does not observe them all.

The conceptual awkwardness of having linguistic knowledge – as binding constraints genuinely appear to be – that is not made operative during the grammatical parsing process has already been pointed out in the literature. The first serious remark on this issue we found is due to Correa (1988), who observed that although the integration of binding constraints "into rules which may be used to derive structure that already satisfies the [constraints] is not a straightforward task" (p.123), that should be the path to follow. This was indeed the path followed by Correa, as we will discuss below in the next subsection.

Second, on a par with its lack of conceptual soundness, the GB approach to the integration of binding constraints into grammar also disregards any concern with representational economy. Following the algorithm in (13), one will get for a certain input string at least as many output trees/grammatical representations as there are coindexations of NPs (modulo indices renaming) that do not violate any binding constraint.

Third, by not adopting an underspecification strategy to represent the outcome of binding constraints satisfaction, the GB approach also disregards any concern with

interfacing the grammar module with a system of anaphor resolution. The input from a GB grammar for such a system will not be a grammatical representation to be refined and further specified vis-à-vis the interplay of preferences for anaphor resolution, but a forest of differently indexed trees that will have to be internally searched and compared with each other by the resolution system.

Finally, computational economy also seems to be disregarded given that the algorithm generates a number of intermediate results which are discarded. As Ingria and Stallard (1989, p.263) put it "the paradigm assumed there is Generate and Test: indices are freely assigned and the Binding Conditions are applied to rule in or rule out particular assignments. Clearly, from a computational standpoint this is grossly inefficient."

Looking for objective support for this concern, and leaving aside the filtering part of the processing strategy, Fong (1990) observed that the free indexation problem can be expressed as a well known combinatorial partitioning problem. He argued that "the problem of assigning m distinct indices to n noun phrases is isomorphic to the problem of partitioning a set of n elements into m non-empty disjoint subsets. The correspondence here is that each partitioned subset represents a set of noun phrases with the same index" (p.106). Accordingly, he has shown that the number of possible indexations grows exponentially as the Bell number of NPs in the input sentence, given by $\sum_{m=1}^{n} \left\{ {n \atop m} \right\}$.

Consideration of these highly cumbersome aspects of GB binding theory gave rise to a couple of alternatives that tried to improve integration of binding constraints in grammar.

Correa: gathering indices while walking down the tree

The first proposal for enhancing integration of GB binding theory into grammar was made by Correa (1988). Simplifying certain details, the proposed algorithm for making grammatical representations conform to binding constraints can be outlined as follows:

(14) Start from the top of the tree with two empty stacks A and B where indices will be collected, respectively local c-commanding indexes and non-local c-commanding indexes. While walking down a tree where every NP has a distinct index,

when an NP is found:
(i) copy: leave a copy of A (if it is an anaphor) or B (if it is a pronoun) at the NP node;
(ii) assign: take the first index of the stack copied into the NP node, take the NP index y, and annotate the NP with y=x;
(iii) collect: add NP index y to A

When a local domain border is crossed
(iv) reset: reset B to A B;
reset A to the empty stack.

This algorithm was given two different implementations, one by Correa (1988), and the other by Ingria and Stallard (1989) – further elaboration by Giorgi et al. (1990) and Pianesi (1991) led to a restatement of the algorithm using formal languages techniques with no sensible improvement in perspicuity. The difference between them is based on the fact that the first applies the procedures of the algorithm on a par with the application of parsing rules, while the latter applies the algorithm once parsing has been completed.

Given the specific shape of this algorithm for integrating binding into grammar, the possibility of using the do-it-while-parsing strategy of Correa's implementation is highly dependent on a top-down parsing strategy. It has, however, the advantage of discarding a special-purpose post processing module for binding as required in Chomsky's (1981) approach. On the other hand, Ingria and Stallard's implementation has the advantage of being independent of the parsing strategy adopted. This is done however at the cost of still requiring a special purpose post parsing module. The contrast between the advantages of these implementations thus suggests that, with this algorithm, although it is possible to integrate binding constraints into grammar, it is at the cost of restricting choices concerning admissible parsing strategies.

Moreover, the algorithm is acknowledged to disregard binding principle C, and to provide no account for backwards anaphora, or cross-over cases (Correa, 1988, p.127, Ingria and Stallard, 1989, pp.268ff). Given its dependency on the constituent structure of the tree to be walked down, it will also not escape from the well known drawbacks associated with configurational formulations of binding theory (cf. Pollard and Sag, 1994, chap.6).

Still, the alternative algorithm in (14) presents some considerable advantages. If we disregard step (ii) – which is but a sort of disguised recency preference spuriously mixed with binding constraints – and allow the result of verifying

binding constraints to be the assignment to an NP of the set of indices of its admissible antecedents, then we have considerable improvements vis-à-vis the GB algorithm in (13). On the one hand, we are able to discard the proliferation of differently indexed trees. On the other hand, we end up with an underspecified grammatical representation with suitable interface points as regards anaphor resolution modules.

Besides these positive aspects, Correa's algorithm also improves the implementation of binding constraints in terms of computational efficiency. He notices that "...the time needed to compute A and B at each node from the A and B at that node on which they directly depend may be assumed to be constant: the operations involved are assignment, push, and pop only. Assuming further [...] that the number of nodes in the derivation trees generated is proportional to the input length [...] since A and B are computed at most once at each node in the tree, the processing time needed is linear – a significant improvement over [Chomsky's Binding Theory]" (p.127).

Johnson: searching in a Leibnizian tree

The second alternative version for the GB mainstream algorithm in (13) is due to Johnson (1995). The algorithm designed by this author is embodied in Prolog code. Trying to abstract away from details associated to that format, we can provide the following outline:

(15)
- (i) Repeat (ii) until all NP_i ($1 \leq i \leq n$) in the tree have been used as starting points;
- (ii) Walk up the tree from NP_i and repeat (iii) until the top node of the tree is reached;
- (iii.i) When other locally c-commanding NP_j is found:
  - (iii.i.i) if NP_i is a short-distance reflexive, annotate NP_i with i=j;
  - (iii.i.ii) if NP_i is a non-reflexive, annotated NP_i with i=/=j;
- (iii.ii) When other non-locally c-commanding NP_j is found: if NP_i is a non-pronoun, annotate NP_i with i=/=j.

Although the algorithm appears in a bottom up fashion, Johnson (1995) ingeniously developed an implementation of it which is independent of the specific parsing strategy used. Besides, in spite of the post parsing flavor of the algorithm, Johnson's implementation likewise does not require a special purpose post parsing module. These results were obtained by introducing two accessory devices.

On the one hand, each node in the tree is "conceptualized as a pair consisting of a tree and a vertex in that tree" (p.62). Consequently, the whole tree where a given NP appears is locally accessible to be "walked up" as its replica is present at the pair (Category, Tree), which is the NP node itself.

On the other hand, binding constraints are introduced by lexical entries of NPs, as illustrated below with a few examples from Johnson, 1995, p.76:

(16)
```
lexicon (himself, Node) :-- [
    label (Node, np:_),
    locally_bound(Node)].

lexicon (harold, Node) :-- [
    label (Node, np:_),
    free(Node)].
```

The predicates `locally_bound/1` and `free/1` are defined in terms of other auxiliary predicates as illustrated below, and resort to the use of constructive negation (Johnson, 1995, p.77):

(17)
```
locally_bound(Bindee) :-- [
    locally_c_commands(Binder, Bindee),
    label(Binder, BinderLabel),
    label(Bindee, BindeeLabel),
    BinderLabel = BindeeLabel)].

free (Node) :-- [
    ~bound(Node) ].

bound (Node) :-- [
    c_commands(Binder, Bindee),
    label(Binder, BinderLabel),
    label(Bindee, BindeeLabel),
    BinderLabel = BindeeLabel)].
```

If a top-down search strategy is not used, the predicates `c_commands/2` and `locally_c_commands/2` activated by lexical items and used "to walk up" the tree are simply delayed until the parent node of their second argument is instantiated (vd. Johnson, 1995, p.81, for more details).

This algorithm for processing binding constraints shows clear advantages over the original GB proposal. It avoids post parsing by allowing for binding constraints to be fully integrated into grammar.

It also improves Chomsky's (1981) algorithm in terms of computational efficiency inasmuch as it does not resort to free indexation of trees. This is possible because, contrary to Chomsky's approach, the ambiguity of pronouns receives a different representation from the ambiguity of reflexives. In Johnson's approach only reflexives and their antecedents end up coindexed – due to the equality

constraint in `locally_bound/1` and `bound/1` –; while the index of a pronoun is only made "unequal" with non-grammatical antecedents of it – due to the inequality constraints induced by the negation operator in `locally_free/1` and `free/1` (cf. Johnson, 1995, p.82). Grammatical coindexations of pronouns with their possible antecedents are thus dismissed, and only the ungrammatical cases are signaled (the following examples are from Johnson, 1995, p.82):

```
|?- prove(parse([rupert,might,like,him], Tree), Residue).

Residue = [A\=B],
Tree = ip/[np:B/ -rupert,i1/[i/ -might,vp/[v/ -like,np:A/ -him]]]

|?- prove(parse([rupert,might,like,himself], Tree), Residue).

Residue = [],
Tree = ip/[np:A/ -rupert,i1/[i/ -might,vp/[v/ -like,np:A/ -
        himself]]]
```

The only grammatical coindexations represented in the outcome are the ones of reflexives with their antecedents.

It is tempting to compare the two approaches, by Correa and Johnson, as one can see that each one succeeds where the other fails.

On the one hand, with the technique of collecting sets of indices, Correa completely avoided the proliferation of indexed trees, while keeping the full scope of binding theory predictions (for principles A and B only) inasmuch as every grammatically possible interpretation (coindexations) of ambiguous anaphoric NPs ends up represented.

This strong point of Correa's approach is the weak point of Johnson's proposal as in the latter the proliferation of indexed trees is not avoided. It is a fact that in contrast to Chomsky's approach, the proliferation of indexed trees is reduced. It is reduced to the set of trees with different admissible indexations between reflexives and their antecedents. But this reduction of tree proliferation in Johnson's account is done at the expense of removing from grammatical representation the explicit encoding of certain types of anaphoric dependencies, namely the ones involving non-reflexives (ruled by principles B and C). Moreover, while this proliferation is not fully eliminated, just as in Chomsky's binding theory, no suitable interface device is made available to connect grammar with anaphor resolution systems.

On the other hand, Correa's approach is not independent of the parsing strategy adopted if one wants to avoid having a special purpose post parsing module to check binding constraints.

In part due to the technique of replicating the whole tree at each node, Johnson overcame that difficulty, and made it possible for binding constraints to be checked out without resorting to a post parsing module or being dependent on a specific

parsing strategy. As Johnson claims, this way of making global context available at a local level – very much as what happens in Leibniz monadology, where each atom is a replica/representation of the whole universe surrounding it from its perspective – can be generalized to check out, not only binding constraints, but any other non-local constraint.

What these considerations naturally led us to is the observation that an approach combining the two successful techniques put forward by Correa and Johnson is the step that suggests itself. Locally packaging non-local context is a requisite for fully integrating non-local constraints into grammar, without resorting to post parsing modules. And packaging grammatical antecedents into sets of markers is a requisite for fully representing ambiguity of anaphoric items and having suitable interface points with anaphor resolvers, without resorting to a forest of trees.

In the following sections, we will try to follow these guidelines and provide an implementation of binding constraints in HPSG by specifying them with the HPSG description language. But first, we will take a look at how binding constraints are handled in Lexical Functional Grammar.

In LFG: a lexical twist with a special purpose device

In the framework of Lexical Functional Grammar (LFG) the principal work on binding is due to Dalrymple (1993). It is out of the scope of the present subsection to provide a comprehensive introduction to that grammatical framework – overviews can be found in Kaplan, 1994, and Neidle, 1994. For the sake of the discussion as to how binding theory is accommodated in grammar, it suffices perhaps to point out that this is a multi-level grammatical framework, with a level of grammatical representation for constituent structure (c-structure), another for grammatical functions (f-structure), and possibly others among which there is one for representing semantic structure. The specification of binding constraints is done mostly at the level of f-structure, whose description language is an AVM-like language approximate to the description language of HPSG described above in Section 6.1.

In order to illustrate the internal geometry of an f-structure representation, the f-structure of sentence (18)a. is displayed in (18)b. (cf. Dalrymple, 1993, p.124).

(18) a. John introduced Bill to himself.

b.

$$\begin{bmatrix} \text{PRED} & \text{'introduce}\langle(\uparrow \text{SUBJ}), (\uparrow \text{OBJ}), (\uparrow \text{OBL}_{Goal})\rangle\text{'} \\ \text{SUBJ} & [\text{PRED} \quad \text{'John'}] \\ \text{OBJ} & [\text{PRED} \quad \text{'Bill'}] \\ \text{OBL}_{Goal} & [\text{PRED} \quad \text{'himself'}] \end{bmatrix}$$

As happened with Johnson's proposal, in Dalrymple's proposal the binding constraints are associated with lexical entries of NP and under the form of equalities (for reflexive items) and inequalities (for non-reflexive items). For instance, the following two readings of the [illegible] in a [illegible], are captured by the equations in b.:

(19) a. John$_i$ introduced Bill to himself$_i$.

b. $((\text{OBL}_{\text{Goal}}\ )\ \text{SUBJ})$ [illegible]

(20) a. John introduced Bill to himself.

b. $((\text{OBL}_{\text{Goal}}\ )\ \text{OBJ}$ = [illegible]

Taking the first example, the right-hand side of the equation stands for the semantic representation ( ) of the f-structure ( ) of the anaphor. The left hand side stands for the semantic representation of a subject (SUBJ) in an f-structure whose Oblique complement (OBL_{Goal}) is the anaphor ( ).

There are, however, significant differences with respect to Johnson's approach. In LFG equations, or inequations, do not relate indices but, as shown in the examples above, semantic representations of anaphoric items and their antecedents. On the other hand, and this is certainly the most significant trait of LFG approach, the equations may integrate expressions involving the special purpose inside-out functional uncertainty device.

Functional uncertainty makes it possible to describe the relation between two f-structures in a way that "uncertainty" is allowed concerning the relative distance between them in the geometry of the global f-structure where they occur. Basically, this is done by introducing regular expressions into the description language of f-structures by means of which an infinite disjunction of paths within f-structures is encoded in finite terms.

Inside-out functional uncertainty is a specific type of functional uncertainty where the different possible relations between two f-structures are established with

respect to the more embedded f-structure of the f-structures being related. For instance, the anaphoric potential delimited by the binding constraint corresponding to the Portuguese long-distance reflexive ela própria in the example a. below is captured by the equation in b. involving inside-out functional uncertainty over the "length" of the path in f-structure between the object ela própria and the possible subject antecedent:

(21) a. A Maria$_i$ disse que a Susana$_j$ disse que a Helena$_k$ disse... que a Carolina$_l$ gosta dela própria$_{i/j/k/.../l}$.

b. ((COMP* OBL) SUBJ) =

Given that an expression involving functional uncertainty is interpreted existentially (vd. Dalrymple, 1993, p.12 for details), the uncertainty encoded in the regular expressions is reflected in the multiplicity of possible solutions to the equality.

Besides including functional uncertainty, equations are also labeled with different off-path requirements over the path of undetermined length specified with the help of the regular expression. Among other things, these requirements help to set apart local and non-local domains. For instance, the following equation resorts to the fact that the f-structure of tensed clauses bears the attribute TENSE to impose a locality requirement for certain short-distance anaphors, based on the fact that tensed clauses are local domains for these anaphors (and f-structure with TENSE cannot be "crossed"):

(22) ((DomainPath*) Antecedent path) =
(TENSE)

Some formal properties of functional uncertainty have been worked out. For instance, in spite of initial skepticism by Kaplan and Bresnan (1982), Kaplan and Maxwell (1988) have shown that the verification problem (determining whether an f-structure satisfies a particular specification with functional uncertainty) is decidable for cyclic and acyclic f-structures, and the satisfiability problem (determining whether a solution exists given some specification with functional uncertainty) is decidable for acyclic f-structures. However, many formal details remain to be clarified.

For instance, the formal properties of off-path requirements have not been worked out. It is also not clear how exactly the binding equations, or inequations, should be defined for each lexical item as they have to specify concrete grammatical functions for anaphoric items and their antecedents, and both anaphoric items and antecedents may bear one of a range of different grammatical functions. Also, it is

not defined how non-lexical NPs (e.g., definite descriptions in Portuguese and English) may be assigned the respective equation, or inequation (a problem also occuring in Johnson's approach). These circumstances, or at least some of them, may help us understand why in the comprehensive overview by Backofen et al. (1996, pp.95ff) not one of the LFG implemented grammars is reported as comprising an implementation of binding theory.

While the solution for these drawbacks seems to be merely waiting for LFG researchers to conduct a more detailed examination, other factors however seem to present more serious difficulties for the LFG approach concerning the integration of binding constraints in grammar. The first point to note is that the LFG approach espouses a very reductionistic view of the semantics of anaphoric dependencies. Notice that the equations do not relate indices that tag anaphors and corresponding antecedents, indices that simply serve to identify the admissible antecedents and support the specification of any of the several different ways in which anaphors and antecedents can be semantically related (cf. Section 5.4). LFG binding equations, or inequations, relate semantic representations, this way restricting and incorrectly enforcing any kind of anaphoric relation to the sole modality of coreference.

The other problematic point to notice is similar to the observation we put forward concerning the system of equalities and inequalities proposed by Johnson. While the ambiguity of reflexives is represented by several different f-structures resulting from the several different (existential) interpretations of functional uncertainty, the ambiguity of pronouns is simply omitted in the single f-structure resulting from the universal interpretation of negative equations associated with pronouns (vd. Dalrymple, 1993, p.123). Again, the proliferation of representations, not being eliminated, is restricted at the expense of simply not assigning room in grammatical representation for the ambiguity of pronouns.

Consequently, another negative point of the LFG approach is based on the fact that, as no underspecification is provided for the encoding of grammatically possible anaphoric links, no interface is provided to connect grammar with extra-grammatical anaphora processing systems.

A module at odds in grammar

The above discussion on different approaches to integrate binding constraints in grammar is highly illuminating in many respects. It has shown many promising paths along which research on this issue should continue: lexicalization of the

binding constraints (Dalrymple, 1993, and Johnson, 1995); utilization of packaging strategies of non-local context in order to make it locally available (Johnson, 1995); economic, but yet full representation of anaphoric potential by means of packaging ambiguity of anaphors with underspecification techniques (Correa, 1988); and setting up of suitable interface points between grammar and extra-grammatical reference processing systems (Correa, 1988).

However, this overview of the approaches aiming to integrate binding theory into grammar clearly shows the contrast between the robustness of the linguistic generalizations encoded in binding theory and the frailty of their formal underpinnings. Another important point that comes to light is the fact that the different approaches all espouse the same conceptual pattern for understanding binding, a pattern that dates back to the basic specification+processing schema of Chomsky (1981).

Above we have mostly considered works concerned with the processing part of this schema. Correa (1988) put forward an approach to resolve the problem of proliferation of representations but made it dependent on parsing strategy. Johnson (1995) removed this dependency by making non-local context locally available, but only partially reduced the proliferation of representations, and this at the expense of not representing the ambiguity of pronouns. Dalrymple (1993) tried to remove from binding theory any concern with processing issues by extending the constraint-based description language of LFG with a device for underspecifying the path in grammatical representation between anaphors and their antecedents; but aside from a non-configurational definition of constraints, all the other basics of Chomsky's approach were embraced, with no solution for the proliferation of representations and correlated drawbacks.

As for the binding theory as argued by Pollard and Sag (1994), the reasons underlying the fact that it had not been turned into an HPSG binding theory may perhaps be more evident now. All the issues concerning the processing part of the schema specification+processing from Chomsky's proposal for binding were disregarded. Accordingly, and in the context of the present discussion, it should not appear as a surprise if we deem Pollard and Sag's (1994) account of binding merely as an improvement of GB binding theory with a special non-configurational variant of the constraints, which belong to the specification part of the schema originally put forward in Chomsky, 1981. Under this perspective the suggestion of Koening (1998) for introducing a device in HPSG description language for stating "inside-out constraints" would thus help just to make an HPSG emulation of the LFG emulation of the original GB binding theory.

Summing up, on the one hand, we have some interesting but disperse techniques which have been suggested in works that tried to overcome the problems of Chomsky's (1981) account of binding and that may help to formally integrate

binding constraints in grammars. On the other hand, we have a highly flexible and rigorous framework for the grammatical knowledge and processing of natural languages provided by a constraint-based setup. Accordingly, the path that seems desirable to be pursued is the one that will extract the best from this kind of framework and from these techniques to build up a coherent binding theory fully integrated into HPSG.

In the following sections we will explore the rationale that the epistemic twist on binding inspired by the results obtained in the previous chapters will help us implement this desideratum.

6.3 Binding Constraints in a Constraint-based Framework

Binding constraints are one of the most prominent and interesting types of non-local dependency in grammar. The impetus towards generalization has led research on binding to focus more on what it has in common with other long-distance dependencies than on what, in the group of this type of dependency, may be specific to binding constraints. From our point of view, given the bold syntactic nature of most non-local dependencies, this circumstance has not favored a convenient awareness of the eminently semantic status of binding constraints (cf. discussion in chapter 5 Formal Semantics).

Learning from the difficulties encountered by the syntax-driven paradigm dating back at least to Chomsky, 1981, we believe that the key for improving binding theory is to give close consideration to and provide an intertwined solution for both issues, i.e. non-locality and ambiguity.

The semantic twist

Handling the semantics of anaphora involves ensuring a complete but economic representation of the intrinsic ambiguity of anaphoric items. Given the formal tools reviewed above, it is quite compelling to acknowledge that packaging the anaphoric potential of anaphors in terms of sets of grammatically admissible antecedents is the best device for that purpose (in line with the seminal ideas of Bach and Partee (1980) on indices collection). As discussed above on several occasions, this seems to be an essential part of the semantic representation of anaphors: it avoids the non-negligible proliferation of grammatical representations due to ambiguity; and it provides suitable interface points with extra-grammatical systems of reference processing.

In chapter 5 Formal Semantics, we found that two views on binding constraints and the obliqueness quantification they express are possible. On the meaning-component view, intra-grammatical quantifiers receive a direct rendering in the semantic representation language. As with other expressions of the representation language, the representation of binding constraints supposes a domain model for the sake of their interpretation which is never defined. Even if such a model could be provided, given the intra-grammatical nature of binding constraints, their interpretation would require a huge overload of post-parsing processing, assuming this to be practically possible in the first place.

We are thus left with the alternative meaning-composition view. As with the meaning-component perspective, the semantic representation of binding constraints also gives rise to a set of candidate antecedents verifying these constraints. Nevertheless, from this alternative perspective, they are enforced at the syntax-semantics interface level and do not require extra interpretation domains.

The analysis built up for this approach in chapter 5 was developed in the DRT framework. It was possible to resort to formal devices such as set abstraction, which are not available in the language description of a constraint-based framework such as HPSG. Accordingly, the success of transposing that analysis to a constraint-based framework basically depends on the success of finding equivalent mechanisms for compiling the relevant sets of antecedents in such framework.

This calls for some reasoning on the definition of these sets and on how to possibly emulate set abstraction with relational constraints in the description language.

In this connection, take for instance the definition of the obliqueness quantifier expressed by a given pronominal p and the set B of grammatically possible

antecedents of p defined by means of DRT Σ-abstraction over the relevant quantificational expression:

(23) $$B = \{y : \forall x\, [GSI(\sim P_p, y) < x \leq y \rightarrow \neg P_p(x)]\}$$

Set B contains any y that verifies the condition $\forall x\, [GSI(\sim P_p, y) < x \leq y \rightarrow \neg P_p(x)]$. From B every marker that makes the above condition false is thus excluded, precisely the markers y such that $P_p(y)$. Therefore, B is the complement set of $\{y : P_p(y)\}$, i.e. $B = \{y : \neg P_p(y)\}$. Given the definition of P_p, B is the set of reference markers that are not local o-commanders of p. Accordingly, if U is the set of all relevant markers in the context and A is the set of local o-commanders of p, then $B=U \setminus A$.

By similar reasoning, it naturally follows that the set A of local o-commanders is the set of grammatically admissible antecedents of short-distance reflexives; that the set Z of o-commanders is the set of grammatically admissible antecedents of long-distance reflexives; and that the set C of grammatically admissible antecedents of non-pronouns is given by $C=U \setminus Z$.

Consequently, knowing the three sets U, A, and Z is the minimal information required in order to decide about the set of grammatically admissible antecedents for any of the four types of anaphors.

Feature passing and Leibnizian trees

Having discussed how to handle ambiguity of anaphors by emulating DRT Σ-abstraction over obliqueness quantification with set collection and operations, we turn now to the non-local nature of binding constraints.

A commonly adopted technique for handling non-local or long-distance dependencies in constraint-based frameworks dates back to Pereira, 1981, and was made popular by Gazdar et al. (1985). Known as feature passing (Johnson, 1995), structure sharing (Pollard and Sag, 1994), and sometimes also as gap threading when applied to unbounded dependency constructions, this technique avoids resorting to special purpose recursion mechanisms (such as functional uncertainty in LFG) by envisaging non-local constraints as an appropriate sequence of local constraints.

Feature passing

As discussed above, a rudiment of this technique applied to binding was first tried out by Correa (1988), where indices were successively collected into sets by scanning down syntactic representation. We believe this pioneering work indicates the way forward if we wish to find the sets A, Z and U, although its exact formulation requires a certain amount of qualification and a few drawbacks need to be removed.

One first point to note is that sets A and Z cannot be collected from nodes in the syntactic constituency representation. The reason for this caveat is based on the fact that collecting reference markers from each constituency node is confined to selecting relevant commanders on the basis of configurational notions. Actually, taking again the "walking down" algorithm of Correa (1988), one recognizes that the set of indices collected up to a given node includes the c-commanders of that node inasmuch as the path followed was determined by constituent structure, not by the obliqueness order of grammatical functions. Consequently, a feature passing technique for collecting sets A and Z rather than relying on collecting markers node by node in constituency representations, will have to collect markers from syntactic heads, in particular from information on the obliqueness list of arguments they subcategorize.

Besides this configurational problem, another point we need to assess involves collecting the third set, the set U of all markers – when discussing Correa's algorithm, Merlo (1993) had also made the point that improving this algorithm would have to involve, among other things, the collection of the set of all markers in the context. The first aspect to note is that the collection of U should preferably be done in a bottom up fashion. It naturally follows that the markers collected at the level of daughter nodes of a phrase will also all be collected at the mother node. When arriving at the top level, one ends up with the set of all reference markers in the relevant grammatical representation.

A second aspect to note concerns the fact that, for the collection of U, one single "track" of feature passing is not enough, as the set U relevant at every node is only obtained at the top node of the global grammatical representation. Consequently, the collection of U has to comprise two "tracks" of feature passing. One is the collecting "track" itself which brings the markers into the set while it is being construed. The other is the disseminating "track" by which the set U obtained is passed to every node.

Leibnizian trees

Before moving to the next subsection of these guidelines for developing a HPSG account of binding, it is worth discussing the alternative to feature passing strongly argued for by Johnson (1995). Johnson defends that his Leibnizian approach to non-locality improves on the feature passing technique in relevant aspects. He notes that

> "While many of these [non-local] relationships can be reduced to strictly local ones by using feature-passing, the resulting grammars sometimes have an unnatural, mechanical quality, in effect simplifying the parsing problem at the expense of complicating the grammar. Writing strictly local grammars can be more akin to computer programming than doing linguistic analysis, and usually proceeds by first identifying the relevant non-local relationship and then devising a feature-passing system to "implement" it." (p.4-5).

If we disregard the difficult questions concerning the perspicuity or "natural quality" of grammar representation and the borderline between "linguistic analysis" and "computer programming", there are some objective questions worth discussing with respect to applying this technique to binding.

Contrary to other non-local relations, e.g. unbounded dependencies, the representation of ambiguity is a crucial issue to be dealt with in the implementation of binding constraints. Neglecting this aspect leads to serious difficulties, as explained in the previous sections, namely proliferation of grammatical representations and absence of interface points between grammar and post-grammatical reference processing modules. Accordingly, also in the context of an approach based on Johnson's Leibnizian technique for packaging grammatical context, the implementation of binding constraints has to comprise the collection of sets of reference markers. Therefore, in Leibnizian trees, sets A, Z and U have to be built as many times as there are nodes of anaphoric NPs that will trigger the verification of binding conditions. As a result of avoiding one "global" feature-passing for collecting these sets, in the Leibnizian approach for non-locality they end up being collected several times at each relevant node of the tree.

Given this, and even accepting without discussion Johnson's claim as to the cumbersomeness of the feature passing technique, it follows that the alternative Leibnizian approach proposed cannot be taken as undoubtedly superior for handling binding. Inasmuch as this sort of non-local dependencies differs from other non-local dependencies by involving intrinsic ambiguity, its formal account seems at least to be less efficient with the Leibnizian technique.

Specification of binding constraints in HPSG

Given the exploratory considerations above, the development of a constraint-based specification of binding is quite direct, although it involves some reshuffling of the HPSG feature geometry. In particular, two areas will be rearranged in order to accommodate the type of information required to handle binding constraints.

On the one hand, semantic representation coded as the value of CONTENT feature is enlarged with the new feature ANAPHORA. This feature keeps information on the discourse referents contributed by NPs and on the set of their grammatically admissible antecedents.

For the sake of illustration, an initial outline of it is depicted below:

(24)

$$\begin{bmatrix} \text{PHON} & \langle \int \text{i} : \rangle \\ \text{SYNSEM}|\text{LOC}|\text{CONT} & \begin{bmatrix} \text{LS} & \ldots \\ \text{SUBORD} & \ldots \\ \text{CONDS} & \ldots \\ \textbf{ANAPHORA} & \ldots \end{bmatrix} \end{bmatrix}$$

On the other hand, the representation of the grammatical context coded in NONLOCAL feature is also enlarged in order to record sets A, Z and U.

Features TO-BIND and INHERITED proposed in Pollard and Sag, 1994 are grouped as features of the new sort udc, standing for unbounded dependency constructions. The new sort binding is introduced which has LIST-A, LIST-Z, LIST-U and LIST-protoU as attributes:

(25)

$$\begin{bmatrix} \text{PHON} & \langle \int \text{i} : \rangle \\ \text{SYNSEM} & \begin{bmatrix} \text{LOC}|\text{CONT} & \ldots \\ \text{NONLOC} & \begin{bmatrix} \text{UDC} & {}_{udc}\begin{bmatrix} \text{INHER} & \ldots \\ \text{TO}-\text{BIND} & \ldots \end{bmatrix} \\ \textbf{BINDING} & {}_{binding}\begin{bmatrix} \textbf{LIST-A} & list \\ \textbf{LIST-Z} & list \\ \textbf{LIST-U} & list \\ \textbf{LIST-protoU} & list \end{bmatrix} \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

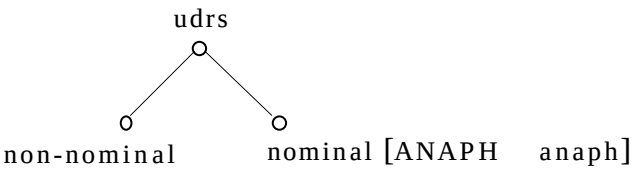
Following these two suggestions, it is quite straightforward to work out a full account of how feature passing is correctly established and how the adequate semantic representation is assigned. In what follows we discuss how the signature (sort hierarchy plus associated appropriateness conditions), the lexical signs, and the grammatical principles are adjusted or improved.

For the sake of consistency and ease of reference, the feature geometry assumed below is the one proposed by Pollard and Sag (1994). We are aware of several improvements that have been proposed since then, concerning the design of that geometry. Given the considerable self contained nature of binding with respect to other grammatical phenomena, the core insights developed below for a formal account of binding will remain, however, unchanged in its basic lines even if here and there feature geometry may be altered to cope with accounts for other phenomena that diverge from the ones originally proposed for HPSG. In any case, the specification for binding constraints designed below is put forward primarily as a prototype – or a methodology – of how to model binding in a constraint-based setup, a prototype that is expected to be conveniently adjusted to particular grammars in accordance to their specific feature geometry.

Feature declaration

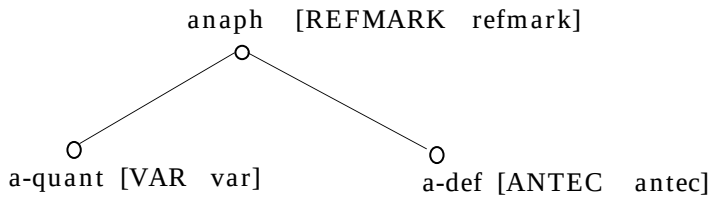
CONTENT value is a feature of sort udrs, the feature by means of which the semantics of the corresponding sign is represented. Following Frank and Reyle (1995), this sort has the attributes LS, SUBORD, and CONDS (vd. (10) and (24) above). To these attributes, the attribute ANAPHORA is now added for nominal signs. The value of this new attribute is a feature of sort anaphora:

(26)



Attributes holding for sort anaphora are designed in such a way that adequate attributes hold for the different types of nominals. For instance, while a definite NP is expected to have a feature CONT|ANAPH|ANTEC – whose value is the set of admissible antecedents –, a quantificational one is not. We thus propose the following feature declaration for the sort anaphora and its subsorts:

(27)



The fact that any sort in the hierarchy above has attribute REFMARK is justified by the circumstance that, contrary to common wisdom, quantificational NPs also contribute a marker which can serve as antecedent in non-bound, e-type anaphoric relations (vd. Section 5.4). As to the attribute ANTEC, only anaphoric NPs bear it. Generic and indefinite NPs bear just the top sort anaphora as the value of ANAPHORA. Quantificational NPs bear a-quant subsort, which besides the marker in REFMARK value, allow these nominals to introduce in the context another reference marker by means of VAR feature to ensure bound anaphora links (this distinction replicates the distinction between w and w_q reference markers discussed in Section 5.4).

Turning now to the second area in the feature geometry that needs to be extended, the nonloc sort is thus defined as follows:

(28)

$$\textit{nonloc}\begin{bmatrix} \text{UDC} & \textit{udc} \\ \text{BINDING} & \textit{binding} \end{bmatrix}$$

The sort udc has the attributes previously assigned to nonloc by Pollard and Sag (1994). The newly introduced sort binding has the attributes LIST-A, LIST-Z, LIST-U and LIST-protoU:

(29)

$$\textit{binding}\begin{bmatrix} \text{LIST}-\text{A} & \textit{list} \\ \text{LIST}-\text{Z} & \textit{list} \\ \text{LIST}-\text{U} & \textit{list} \\ \text{LIST}-\text{protoU} & \textit{list} \end{bmatrix}$$

Following the suggestion above about the need for two “tracks” for feature passing, LIST-protoU is the feature in which reference markers from the whole context are successively collected, and LIST-U is the feature by means of which those referents are disseminated.

LIST-A is the list with the obliqueness relation of the relevant local domain, and LIST-Z is the list with the relevant non-local obliqueness relation.

The values of these attributes are lists of reference markers. The elements of these lists are structure shared with the values of REFMARK and VAR features of other signs. For the sake of uniformity we assume that all these values are lists. While lists are the data type required to encode obliqueness hierarchies in the case of LIST-A and LIST-Z, in the case of LIST-U and LIST-protoU, as no obliqueness hierarchy is at stake, sets of reference markers would be enough. Besides, for the sake of accuracy and to ensure a complete account of subject-orientedness as designed in Section 3.3, the values of LIST-A and LIST-Z should be allowed to have not only reference markers but also sets of reference markers as their elements.

In this connection it is also interesting to note that, for the sake of reference processing, in general, and anaphor resolution, in particular, the representation of reference markers is expected to be enriched with further information. For instance, given agreement exists in most cases between anaphors and antecedents, the representation of reference markers will have to be enriched with information on morphological inflection of the corresponding NPs (in line with Pollard and Sag (1994) proposal for a feature of sort index). Also given the specific constraints that markers introduced by VAR impose on the anaphors that can take it as antecedents (cf. discussion on bound anaphora in chapter 5 Formal Semantics, section 5.4), markers should also bear information on its origin, i.e. the type of feature – REFMARK or VAR – that passed them to the global context to enter the process of anaphor resolution.

Besides this enrichment of the representation of reference markers, in order to get a suitable input for further reference processing, the representation of the list of grammatically possible antecedents – rendered as the ANTEC value – should also be enriched. In particular, as discussed in Eschenbach et al., 1989, in the case of plural anaphors, and given that split antecedency may occur, their ANTEC list should receive further processing before the anaphor resolver is applied. The reference markers of that list, either singular or plural, should be combined into other plural markers in the way of i-sums a la Link (1983) (see also remarks on split anaphora in Section 4.3).

Lexical signs

Given the feature declaration defined above, it is possible now to design the lexical entries for items of different anaphoric type.

As shown below, the lexical entry for a reflexive contributes a reference marker to the global context, by letting REFMARK and LIST-protoU values be token identical.

The set of grammatically admissible antecedents of the reflexive is the result of picking from LIST-A the markers that o-command the marker contributed by it. This is the result of the two place relational constraint ocomm, which locates its second argument inside its first argument and returns the predecessors of the second argument.

According to Section 3.3, if LIST-A encodes a linear, non-branching obliqueness hierarchy, the reference markers are organized as elements of a single list. If in turn, LIST-A encodes a branching hierarchy, the reference markers are organized as a list in which the elements may be not only reference markers but also sets of reference markers. For the sake of keeping the specification simpler and focusing on the essential aspects, in the discussion below, we will mostly concentrate on linear obliqueness hierarchies.

(30)

$$synsem\begin{bmatrix} \text{LOC}|\text{CONT} & \begin{bmatrix} \text{LS} & \begin{bmatrix} \text{L-MAX} & \boxed{3} \\ \text{L-MIN} & \boxed{3} \end{bmatrix} \\ \text{SUBORD} & \{\ \} \\ \text{CONDS} & \left\{\begin{bmatrix} \text{LABEL} & \boxed{3} \\ \text{ARG-R} & \boxed{2} \end{bmatrix}\right\} \\ \text{ANAPH} & \begin{bmatrix} \text{REFMARK} & \boxed{2} \\ \text{ANTEC} & ocomm(\boxed{1},\boxed{2}) \end{bmatrix} \end{bmatrix} \\ \text{NONLOC}|\text{BIND} & \begin{bmatrix} \text{LIST-A} & \boxed{1} \\ \text{LIST-Z} & list \\ \text{LIST-U} & list \\ \text{LIST-protoU} & \langle \boxed{2} \rangle \end{bmatrix} \end{bmatrix}$$

A lexical entry for long-distance reflexives is quite similar to the previous one. The sole difference concerns the relational constraint that returns the value of ANTEC, whose second argument is structure shared not with LIST-A but with LIST-Z:

(31)

$$synsem\begin{bmatrix} \text{LOC}|\text{CONT} & \begin{bmatrix} \text{LS} & \begin{bmatrix} \text{L-MAX} & \boxed{3} \\ \text{L-MIN} & \boxed{3} \end{bmatrix} \\ \text{SUBORD} & \{\ \} \\ \text{CONDS} & \left\{\begin{bmatrix} \text{LABEL} & \boxed{3} \\ \text{ARG-R} & \boxed{2} \end{bmatrix}\right\} \\ \text{ANAPH} & \begin{bmatrix} \text{REFMARK} & \boxed{2} \\ \text{ANTEC} & \textit{ocomm}(\boxed{1},\boxed{2}) \end{bmatrix} \end{bmatrix} \\ \text{NONLOC}|\text{BIND} & \begin{bmatrix} \text{LIST-A} & \textit{list} \\ \text{LIST-Z} & \boxed{1} \\ \text{LIST-U} & \textit{list} \\ \text{LIST-protoU} & \langle\boxed{2}\rangle \end{bmatrix} \end{bmatrix}$$

As for pronouns ruled by principle B, their lexical sign will include a synsem which is also similar to previous ones:

(32)

$$synsem\begin{bmatrix} \text{LOC}|\text{CONT} & \begin{bmatrix} \text{LS} & \begin{bmatrix} \text{L-MAX} & \boxed{3} \\ \text{L-MIN} & \boxed{3} \end{bmatrix} \\ \text{SUBORD} & \{\ \} \\ \text{CONDS} & \left\{\begin{bmatrix} \text{LABEL} & \boxed{3} \\ \text{ARG-R} & \boxed{2} \end{bmatrix}\right\} \\ \text{ANAPH} & \begin{bmatrix} \text{REFMARK} & \boxed{2} \\ \text{ANTEC} & \textit{non-loc-ocomm}(\boxed{4},\boxed{1},\boxed{2}) \end{bmatrix} \end{bmatrix} \\ \text{NONLOC}|\text{BIND} & \begin{bmatrix} \text{LIST-A} & \boxed{1} \\ \text{LIST-Z} & \textit{list} \\ \text{LIST-U} & \boxed{4} \\ \text{LIST-protoU} & \langle\boxed{2}\rangle \end{bmatrix} \end{bmatrix}$$

The difference again has to do with the relational constraint that returns ANTEC value. Here the relational constraint non-loc-ocomm takes (in its first argument) the list of all markers in the context given by LIST-U value and remove from it the local o-commanders (included in its second argument) of the pronoun (in its third argument) and the pronoun itself.

Coming now to non-lexical NPs, the specification of their anaphoric potential is done via the lexical specification of determiners. The synsem of a lexical entry for a definite article is given below:

(33)

$$synsem\begin{bmatrix} LOC|CONT & \begin{bmatrix} LS & \begin{bmatrix} L\text{-}MAX & \boxed{4} \\ L\text{-}MIN & \boxed{5} \end{bmatrix} \\ SUBORD & \{\boxed{4} = \boxed{5}\} \\ CONDS & \left\{\begin{bmatrix} LABEL & \boxed{4} \\ ARG\text{-}R & \boxed{3} \end{bmatrix}\right\} \\ ANAPH & a\_def\begin{bmatrix} REFMARK & \boxed{3} \\ ANTEC & non-ocomm(\boxed{2}, \boxed{1}, \boxed{3}) \end{bmatrix} \end{bmatrix} \\ NONLOC|BIND & \begin{bmatrix} LIST\text{-}A & list \\ LIST\text{-}Z & \boxed{1} \\ LIST\text{-}U & \boxed{2} \\ LIST\text{-}protoU & \langle \boxed{3} \rangle \end{bmatrix} \end{bmatrix}$$

ANTEC value is defined by the constraint non-ocomm that returns the list of non-o-commanders of the corresponding NP (for the sharing of values between relevant attributes of an NP node and of its Specifier daughter, see the next subsection on Principles).

It is worth noting that in languages such as Portuguese, NPs introduced by definite articles may also have a generic reading, like bare NPs in English. A lexical representation different from the one above must also to be ensured for definite articles so that this other reading may be represented. But it is also possible to envisage a single lexical entry for a definite article, one that assigns sort anaphora as the value of ANAPHORA. From this perspective, semantic principles will be designed in such a way that under the effect of adequate monotonic specification the relevant NPs will be specified either as a-definite or, say, a-generic.

As for the lexical entries of indefinite articles and quantificational determiners, they are as depicted below, respectively in a. and b.:

(34) a.

$$\textit{synsem}\begin{bmatrix} \text{LOC}|\text{CONT} & \begin{bmatrix} \text{LS} & \begin{bmatrix} \text{L-MAX} & \boxed{2} \\ \text{L-MIN} & \boxed{3} \end{bmatrix} \\ \text{SUBORD} & \{\boxed{2} = \boxed{3}\} \\ \text{CONDS} & \left\{\begin{bmatrix} \text{LABEL} & \boxed{2} \\ \text{ARG-R} & \boxed{1} \end{bmatrix}\right\} \\ \text{ANAPH} & \begin{bmatrix} \text{REFMARK} & \boxed{1} \end{bmatrix} \end{bmatrix} \\ \text{NONLOC}|\text{BIND} & \begin{bmatrix} \text{LIST-A} & \textit{list} \\ \text{LIST-Z} & \textit{list} \\ \text{LIST-U} & \textit{list} \\ \text{LIST-protoU} & \langle \boxed{1} \rangle \end{bmatrix} \end{bmatrix}$$

b.

$$\textit{synsem}\begin{bmatrix} \text{LOC}|\text{CONT} & \begin{bmatrix} \text{LS} & \begin{bmatrix} \text{L-MAX} & \boxed{4} \\ \text{L-MIN} & \boxed{5} \end{bmatrix} \\ \text{SUBORD} & \{\boxed{4} > \boxed{3},\ \boxed{4} > \boxed{5}\} \\ \text{CONDS} & \left\{\begin{array}{l} \begin{bmatrix} \text{LABEL} & \boxed{3} \\ \text{ARG-R} & \boxed{2} \end{bmatrix}, \begin{bmatrix} \text{LABEL} & \boxed{4} \\ \text{REL} & \textit{rel} \\ \text{RES} & \boxed{3} \\ \text{SCOPE} & \boxed{5} \end{bmatrix}, \\ \begin{bmatrix} \text{LABEL} & \boxed{5} \\ \text{ARG-R} & \boxed{7} \end{bmatrix}, \begin{bmatrix} \text{LABEL} & \boxed{8} \\ \text{REL} & \textit{abstraction} \\ \text{ARG1} & \boxed{2} \\ \text{ARG2} & \boxed{7} \\ \text{ARG-R} & \boxed{1} \end{bmatrix} \end{array}\right\} \\ \text{ANAPH} & \begin{bmatrix} \text{REFMARK} & \boxed{1} \\ \text{VAR} & \boxed{2} \end{bmatrix} \end{bmatrix} \\ \text{NONLOC}|\text{BIND} & \begin{bmatrix} \text{LIST-A} & \textit{list} \\ \text{LIST-Z} & \textit{list} \\ \text{LIST-U} & \textit{list} \\ \text{LIST-protoU} & \langle \boxed{1}, \boxed{2} \rangle \end{bmatrix} \end{bmatrix}$$

In the representations above an indefinite article contributes a reference marker, and a quantificational determiner contributes two markers to the global context.

Given that they have no anaphoric capacity, attribute ANTEC is not present, and no value is assigned to it.

Principles

With respect to grammatical principles, there are two main goals to attain. One is that the semantic representation of non-lexical NPs is correctly specified concerning the new feature ANAPH. The other is that the different relevant lists of reference markers are appropriately gathered and shared through the linguistic representation.

The first goal is partly achieved at the lexical signs of determiners, where the reference markers contributed by determiners as values of REFMARK or VAR are ensured to be correctly anchored within the semantic conditions contributed by the relevant nominal heads. In addition, the correct specification of ANAPH for non-lexical NPs also calls for an extension of the Semantics Principle in (11) set up by Frank and Reyle (1995). We propose to refine it with a sixth clause requiring the ANAPH value of a determiner to be token-identical to the ANAPH value of its NP (we are assuming a seventh ID schema, the Head-Specifier schema, cf. next chapter 7 Computational Implementation):

(35) Semantics Principle

Clause VI: In a Head-Spec phrase the SYNSEM|LOC|CONT|ANAPH values of the Specifier daughter and the phrase are token-identical.

Notice that ANAPH value of an NP daughter and of its PP mother phrase are guaranteed to be token-identical by the effect of the semantics principle already enhanced to cope with PPs, requiring the CONT value of these two signs to be token-identical.

As for the second goal to be attained with grammatical principles, it involves handling the lists of reference markers. Most constraints concerned with this issue are organized as clauses of a new principle, which we termed Binding Domains Principle (BDP).

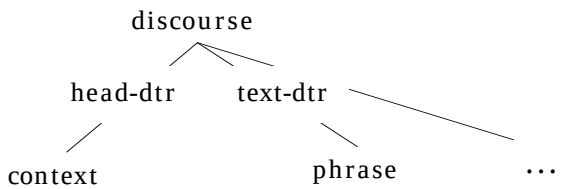
LIST-U

Let us start with LIST-U and LIST-protoU. First, in any phrase it has to be ensured that LIST-protoU value is the concatenation of LIST-protoU values of its immediate constituents.

Second, it is necessary to ensure that at the relevant top phrase of grammatical representation the markers of that grammatical context gathered in LIST-protoU value are passed to LIST-U, by means of which the total list of markers is disseminated over the representation. Accordingly, at that top phrase the values of LIST-protoU and LIST-U are token-identical.

For the purpose of identifying the top phrase where values of LIST-U and LIST-protoU are token-identical, we assume an eighth ID schema, the Head-Text schema. Although in this schema the Head node and the top node are signs of newly introduced sorts, respectively, context and discourse, no significant commitment is made here towards any sensible account of the context or structure of discourse, as the Head just allowed to have one or more sisters of sorts phrase, punctuation, etc.. The context Head has an empty phonological representation and its role consists essentially in bringing into discourse relevant reference markers of the common ground not introduced by NPs of the discourse, thus permitting the interpretation of nominals in first-mention or deictic uses:

(36) Schema 0



Although this is sufficient for our current purpose of handling binding, further research on a possible HPSG account of the structure of discourse would certainly lead to a more thorough representation.

Third, as for LIST-U, its value in any non-NP sign is token-identical to LIST-U value of each daughter of that sign. The exemption of NPs from this general condition is required to avoid violation of what in the literature, and for want of a better term, is known as i-within-i violation. It is crucial that the reference markers introduced by a given NP are not visible as possible antecedents to the anaphoric expressions inside that NP, and analogously that the reference markers

of the anaphoric expressions inside the NP are not visible as possible antecedents for that NP. Accordingly, a special provision is required to handle the Specifier daughter and the head daughter of NPs.

Compiling these requirements into the first clause of BDP, we obtain:

(37) Binding Domains Principle
Clause I

(i) in every sign, LIST-protoU value is identical to the concatenation of LIST-protoU values of its daughters;

(ii) in a sign of sort disc, LIST-protoU and LIST-U values are token-identical;

(iii) in a non-NP sign, LIST-U value is token-identical to each LIST-U value of its daughters;

(iv) in an NP phrase F,

(iv.i) in Spec daughter, LIST-U value is the result of removing the elements of LIST-A value of Head daughter from the LIST-U value of F;

(iv.ii) in Head daughter, LIST-U value is the result of removing the value of REFMARK, and the value of VAR when present, of Spec daughter from the LIST-U value of F.

LIST-A

Turning now to LIST-A, we propose to handle it in the following way.

In the lexical sign of any predicator, LIST-A is made from the ARG-R values of the arguments in SUBCAT value, as exemplified below (in the abbreviatory symbols, the subscripted tag stands for the ARG-R value of the NP):

(38)

$$synsem\begin{bmatrix} \text{LOC}|\text{CONT}|\text{SUBCAT} & \langle \text{NP}_{\boxed{1}}, \text{NP}_{\boxed{2}}, \ldots \rangle \\ \text{NONLOC}|\text{BIND} & \begin{bmatrix} \text{LIST-A} & \langle \boxed{1}, \boxed{2}, \ldots \rangle \\ \text{LIST-Z} & \textit{list} \\ \text{LIST-U} & \textit{list} \\ \text{LIST-protoU} & \textit{list} \end{bmatrix} \end{bmatrix}$$

In phrasal signs in general, LIST-A value is passed from the head to its successive projections, and also from the head to its nominal arguments (or oblique nominal arguments preceded by prepositions). Again, NPs behave in a slightly different way in the sense that LIST-A value of their heads is not passed to its maximal nominal projection.

As what happens with any other value of the attributes of sort binding, LIST-A value of a Specifier daughter and LIST-A value of the corresponding nominal phrase are token-identical.

Clause II of BDP ensures that LIST-A value is circulated according to these requirements:

(39) Binding Domains Principle
Clause II

(i) in a phrase, LIST-A value of its head, and of its nominal (or nominal preceded by preposition) or trace Subject or Complement daughters are token-identical;

(ii) in non-nominal or non-prepositional signs, LIST-A values of a sign and its head are token-identical;

(iii) in a prepositional phrase,

(iii.i) if it is a complement daughter, LIST-A values of the phrase and of its nominal complement daughter are token-identical;

(iii.ii) otherwise, LIST-A values of the phrase and its head are token-identical;

(iv) in a nominal phrase,

(iv.i) in a maximal projection, LIST-A value of the phrase and its Specifier daughter are token-identical;

(iv.ii) in other projections, LIST-A values of the phrase and its head are token-identical.

LIST-Z

Finally, a third Clause of BDP is required to handle LIST-Z. We follow in this respect the strategy of Correa (1988). At the top node of signs strictly dominated by disc node, LIST-Z value is set up as the LIST-A value.

LIST-Z value is however incremented whenever it is passed down to a daughter and either "crosses" a subordinate clause, or a submaximal nominal projection.

In the first case, LIST-Z value of a subordinate clause is the result of concatenating LIST-Z of its mother phrase with its LIST-A value.

In the second case, LIST-Z value of a submaximal nominal projection is the result of concatenating (i) the list of o-commanders of REFMARK value, or instead of the VAR value when present, of its Specifier sister node in the LIST-Z value of its mother phrase, and (ii) its LIST-A value. As happened with LIST-U, in order to avoid i-within-i violations, the reference markers of an NP cannot be visible to the anaphoric expressions inside it. Accordingly, LIST-Z value of submaximal nominal projection cannot include the reference markers of the corresponding NP.

In the remaining cases, LIST-Z value is shared unchanged with LIST-Z value of its daughters:

(40) Binding Domains Principle
Clause III

For a sign F:

(i) in a Text daughter, LIST-Z and LIST-A values are token-identical;

(ii) in a non-Text daughter,

(ii.i) in a sentential daughter, LIST-Z value is the concatenation of LIST-Z value of F with LIST-A value;

(ii.ii) in non-lexical nominal Head daughters, LIST-Z value is the concatenation of L with LIST-A value, where L is the list which results from taking the list of o-commanders of REFMARK value, or instead of the VAR value when this exists, of its specifier sister from LIST-Z value of F;

(iii) in other, non-filler, daughters of F, LIST-Z value is token-identical to LIST-Z value of F.

Note that when branching obliqueness hierarchies are considered, Clause III of BDP above will become a bit more elaborate. It may happen that the LIST-A value of a subordinate clause is not concatenated with LIST-Z value of the subcategorizing head, but is integrated in a set which is an element of LIST-Z value of the

subcategorizing head (cf. discussion on this issue in Section 3.3). As referred to above, for the sake of making the discussion clearer, we will disregard such cases here.

A non-local dependency inside another

The exemption of filler phrases from Clause III has to do with the fact that fillers have to receive its LIST-Z value not from its "local" context in the feature representation, but from the "local" context of the corresponding trace. And the same holds for their LIST-A values.

To make this requirement more tangible consider the following example from Portuguese, involving the topicalization of a reflexive across the boundary of an embedded clause:

(41) De si próprio, cada estudante disse que ele gosta.
of SI PRÓPRIO, every student said that he likes
himself, every student said he likes

In sentences such as this, LIST-A value of the reflexive is not the obliqueness hierarchy of the predicator of the main clause to which it is adjoined, but the obliqueness hierarchy of the predicator downstairs, which subcategorizes for the trace with which the reflexive entertains a so called unbounded syntactic dependency. Consequently, we have in this type of constructions a non-local dependency that has to be recovered – between the trace and the filler, which happens to be the reflexive – so that the other non-local dependency – between the reflexive and its potential antecedents – can be correctly established.

Accordingly, LIST-A and LIST-Z values of the relevant trace have to be passed to the corresponding filler. This can be done together with the other information shared between traces and fillers, encoded as the value of LOC feature, whose circulation is ensured mostly by the NONLOC Feature Principle.

For this purpose, the sort head of prepositional and nominal signs (including the trace) is thus enhanced with two new attributes, COSUBCAT and EXTCOSUBCAT, and a COSUBCAT Principle is set up:

(42) COSUBCAT Principle

In nominal and prepositional signs the values, respectively, of LOC|CAT|HEAD|COSUBCAT and NONLOC|BINDING|LIST-A, and of LOC|CAT|HEAD|EXTCOSUBCAT and NONLOC|BINDING|LIST-Z are token-identical.

This principle ensures token-identity between LIST-A (respectively LIST-Z) value and COSUBCAT (respectively EXTCOSUBCAT) value of the trace. Also LIST-A (respectively LIST-Z) and COSUBCAT (respectively EXTCOSUBCAT) values of the filler are ensured to be token-identical by this principle.

Given that the NONLOC Feature Principle is responsible for ensuring token-identity between COSUBCAT (respectively EXTCOSUBCAT) values of trace and filler, it is possible for LIST-A (respectively LIST-Z) values of filler and trace to be transitively structure-shared, as required.

Nominals as binding machines

In order to make it visible how the different constraints – specially the lexical signs and principles now defined – conjure up to a full account of binding, we will now examine an illustrative example in detail. The example is the sentence in (41), containing three NPs: a non-lexical NP, a pronoun, and a reflexive which happens to be topicalized over the boundary of an embedded clause. The grammatical representations of this example aimed at enhancing the perspicuity of the corresponding sign are presented below in (43) and (44). They are abridged versions of the full grammatical representation in Annex II, Example 3, which was automatically generated by our computational grammar (vd. next chapter 7 Computational Implementation).

Consider first (43), which is meant to help to understand how the topicalized phrase gets the correct assignment for its LIST-A and LIST-Z values.

LIST-A value in the embedded clause has its origin in the lexical sign of likes, the predicator of that clause. In this lexical sign the two reference markers identified with tags $\boxed{24}$ and $\boxed{392}$ are entered as the elements of LIST-A value, under the same obliqueness order they have in the SUBCAT list.

By virtue of Binding Domains Principle, Clause II, (i) and (ii), LIST-A and LIST-Z values of the predicator, identified with tags $\boxed{1}$ and $\boxed{2}$, are structure shared with LIST-A and LIST-Z values of the pronoun, the Subject daughter, and the trace, the Complement daughter.

Given this, by the effect of the COSUBCAT Principle, LIST-A and LIST-Z values of the trace are token-identical with the values of COSUBCAT and EXTCOSUBCAT, which are attributes of the HEAD feature.

Notice that the HEAD feature is part of LOC value, tagged with $\boxed{3}$, which is the piece of information that the NONLOC Principle guarantees to be successively

passed up through the representation until the relevant node dominating the filler is found. This is expressed by means of the feature $\left[\text{NONLOC}|\text{UDC}|\text{INHER}|\text{SLASH}\ \left\{\boxed{3}\right\}\right]$ represented in some of the nodes it is part of.

With the LOC value being passed up, the ID Schema 6, for head-filler constructions, which holds for the analysis of [[of SI PRÓPRIO] [every student said...]], ensures that the SLASH value of the head daughter and the LOC value of the filler daughter are token-identical.

Finally, the COSUBCAT Principle will impose that, also in the sign corresponding to the topicalized phrase, the COSUBCAT and EXTCOSUBCAT values are structure shared with the LIST-A and LIST-Z values. This way it is ensured that the topicalized phrase gets the correct values for LIST-A and LIST-Z values, which originated at the distance in the embedded clause.