Turning now to (44), we find a different abridged version of the grammatical representation of the example we are examining. With this version we may also see how the other lists of reference markers are handled and the anaphoric potential of nominals is defined.

Considering first LIST-Z, one can check that in the outer nodes of the matrix clause, due to the effect of Binding Domains Principle, Clause III, its value is obtained from the value of LIST-A, with which it is token-identical, and which comprises the list with a single element $\left\langle \boxed{54} \right\rangle$. In the nodes of the embedded clause, LIST-Z value is the concatenation of that upper LIST-Z value and the LIST-A value $\left\langle \boxed{24}, \boxed{392} \right\rangle$ in the embedded clause, from which the list $\left\langle \boxed{54}, \boxed{24}, \boxed{392} \right\rangle$ is the result.

Observing now LIST-protoU, it is easy to see that as one ascend in the constituency representation, the list gets longer since by the effect of Binding Domains Principle, Clause I, LIST-protoU value at a given node gathers the reference markers of the nodes dominated by it. At the discourse top node, LIST-protoU ends up as a list including all the reference markers of the NPs in the example, the list $\left\langle \boxed{415}, \boxed{24}, \boxed{247}, \boxed{54}, \boxed{392} \right\rangle$. Clause I of Binding Domains Principle ensures also that this list of all reference markers is passed to LIST-U value of the top node, and that it is then percolated down to all relevant nodes of the grammatical representation.

Taking a closer look at the NPs (whose partial representations can be found below the tree), it is easy to check that every phrase contributes to the global anaphoric potential of their linguistic context by passing the tag of its reference marker into its own LIST-protoU. In the case of the quantificational NP every student two tags are passed corresponding to REFMARK and VAR values. And in the case of the ctx node the reference marker $\boxed{415}$ is taken from the set of semantic conditions that conventionally constitute the non-linguistic context.

On the other hand, the context also contributes to establish the anaphoric potential of each NP. This is ensured by the different clauses of the Binding Domains Principle, which enforce the presence of suitable values of LIST-A, LIST-Z and LIST-U at the different nodes.

Finally, token-identity is ensured between ANTEC value and the outcome of the different relational constraints (not represented in (44)) that are lexically associated with each NP – in case it is a lexical NP –, or to the specifier of the NP – in case this is a non-lexical one. Consequently, the value of ANTEC is a list that records the grammatically admissible antecedents of the corresponding anaphor, sorted out by those relational constraints, whose function is to enforce the conditions on interpretation imposed by binding constraints.

As a final remark, it is thus worth noting that the overall design of the fragment of grammar now developed brings to light the fact that an NP can be envisaged as a "binding machine": it takes a representation of the context, possibly updates its anaphoric potential in accordance with its binding constraint, and contributes to the context, against which the other NPs are also interpreted. This rationale is clearly in line with the seminal insights of Johnson and Klein (1990) concerning the processing of the semantics of nominals and also the spirit (though by no means the letter) of the framework of dynamic semantics (vd. Chierchia, 1995).

6.4 Summary

In this chapter, we began by introducing HPSG, the constraint-based framework for language knowledge representation and processing with which we aimed to fully integrate binding constraints into grammar.

Next we discussed previous attempts to accommodate these constraints either in HPSG or in other frameworks, such as GB or LFG. We based our discussion on learning from the many difficulties and drawbacks found by these attempts and bringing to light two crucial facets that should be ensured in order to achieve an adequate accommodation of binding constraints.

On the one hand, given the non-local nature of the relations involved, and the formal devices available for representing and processing grammatical knowledge, we found necessary to ensure that the relevant aspects of the (non local) context receive a representation at the local level of each anaphor.

On the other hand, in order to avoid costly proliferation of representations and to escape from resorting to extra grammatical levels of processing, it was also found necessary to ensure a packaged representation of the anaphoric potential of anaphors. As discussed, this has also the positive effect of allowing for neat interface points between grammar and reference processing systems.

With this in place, a full specification of binding constraints in the constraint-based setup of HPSG was designed, which is to the best of our knowledge, the first one proposed to date. This was achieved basically by means of extending two areas of the feature geometry. On the one hand, the representation of non-local material was extended by creating the new feature BINDING of the sort nonloc. On the other hand, the semantic representation was improved with the admission of the new feature ANAPHORA for the sort udrs. The first keeps record of the relevant lists of reference markers. The latter encodes the reference marker contributed by the anaphor at stake as well as its anaphoric potential under the form of the list of candidate antecedents complying with the corresponding binding constraint, stated under the form of a relational constraint entered at the lexicon level.

In addition, two extra principles were added. The Binding Domains Principles is responsible for correctly constraining the creation and handling of the different lists of reference markers, and in particular, for ensuring that non-local reference markers are available at the local level of the anaphor. Furthermore, the COSUBCAT Principle helps to ensure that the relevant lists of reference markers in the local domain of the trace are also locally present “at the distance” in the feature structure of the filler.

7 Computational Implementation

7.1 Implementation Formalism 146
Implementation systems 146
ProFIT 146
7.2 Implemented Grammar 146
Grammar fragment 146
Control 146
Performance 146
7.3 Implementation of Binding Constraints 146
Signature 146
Lexicon 146
Clitics and the ambiguity of Portuguese LD reflexive 146
Grammatical principles 146
Complexity issues 146
7.4 Summary 146

As an instrumental step in our research on reference processing, we developed a grammar workbench. This workbench was conceived as a tool to help verify and improve the theoretical consistency and the empirical accuracy of our successive working hypotheses on the formal and computational modeling of binding constraints. At its current stage, in which the specification of binding constraints worked out in previous chapters is implemented, the workbench also represents a demonstration by example of the practical viability of this specification for applied research and development on reference processing.

In this chapter, we will describe the grammar developed and its functionality. First, we will present a brief survey of the computational systems available for implementing HPSG particular grammars, and of the system we opted for. Second, we will discuss the major aspects of implementing the supporting grammar fragment on this system. Finally, we report on the integration of binding constraints into this core grammar.

7.1 Implementation Formalism

Implementation systems

Several different implementation systems for natural language grammars have been construed in the last few years which can be used to implement grammars developed in the constraint-based framework of HPSG. From those, the ones whose development and improvement have been consistently pursued along the years have been documented and comparatively assessed in surveys such as Backofen et al., 1996 or Bolc et al. 1996. These systems are identified below:

(1)

SYSTEMS	AUTHORS	SITE
ALE Attribute Logic Engine	Bob Carpenter and Gerald Penn	Carnegie Mellon Univ.
ALEP Advanced Linguistic Engineering Platform	BIM, Belgium	Cray Systems, Luxembourg
CL-ONE	RGR project	Univ. of Saarland, Saarbrücken, and Univ. of Edinburgh
ConTroll	Seminar für Sprachwissenschaft	Univ. of Tübingen
CUF Comprehensive Unification Formalism	DYANA project	IMS, Univ. of Stuttgart
PAGE/TDL Platform for Advanced Grammar Engineering	DISCO project	DFKI, Saarbrücken
ProFIT Prolog with Features, Inheritance and Templates	Gregor Erbach	Univ. of Saarland, Saarbrücken
TFS Typed Feature Structure	Martin Emele and Rémi Zajac	IMS, Univ. of Stuttgart (Polygloss project)

The above mentioned reports, together with direct experimentation on some of the systems at the Language Technology Laboratory of the DFKI-German Research Center for Artificial Intelligence, Saarbrücken, Germany, provided us with the basis for opting for a specific system on which our workbench could be implemented. One should bear in mind, however, that at the current stage of research "there is no one formalism which covers HPSG completely" (Backofen et al., 1996, p.116). Consequently, whatever the formalism one adopts, eventual adjustments will be required to progress with implementing a grammar fragment specified under the framework of HPSG.

Signature

If one looks at the formalisms from the viewpoint of their implementational capabilities regarding the signature of the grammar, a clear distinction can be made between two sets of systems.

On the one hand, there is ALEP, whose limited type system and absence of suitable inheritance mechanisms does not allow a convenient implementation of HPSG sort hierarchy and corresponding appropriateness conditions.

On the other hand, all the remaining systems have rich enough type systems allowing a thorough implementation of the signature of HPSG grammars. The authors of Genabith et al., 1994, who have extensively experimented with the ALEP system in several EU funded projects, reported that "one can write [in ALEP] not HPSG grammars but 'HPSG inspired' grammars".

PS- vs. SH-based parsing

A second major trait that differentiates the implementation systems listed above has to do with the gist of the strategy adopted for the parsing algorithm. The systems can be differentiated into those whose parsing algorithm is centered around phrase structure (PS-based) and those whose algorithm is centered around the sort hierarchy (SH-based). This distinction underpins important differences both in terms of expressive capacity of the systems, and in terms of their efficiency.