University of Lisbon

Department of Informatics

A Computational Grammar for Deep Linguistic Processing of Portuguese: LXGram, version A.4.1

António Branco Francisco Costa

July 2008

Contents

List of Figures

List of Tables

Acknowledgments

We are very grateful for all the support and encouragement from Hans Uszkoreit, both during the phase of project proposal definition and during the development phase of the GramaXing project.

We are very grateful for all the invaluable help and for the conditions made available at the University of Saarland, Department of Computational Linguistics at Saarbruecken, that permitted the internship of Francisco Costa for 7 months, in 2005-2006.

We want to thank also Frederik Fouvry, who helped in the setting up of the project proposal, and to Valia Kordoni for the warm hospitality and support at Saarbruecken.

Chapter 1

Preamble

This document is the report on the implementation of LXGram, in its version A.4.1 of March 2008.

LXGram is a grammar for the computational processing of Portuguese.

It is being developed under the following major design features:

- precision: it is a precision grammar delivering accurate, linguistically grounded information of natural language sentences
- deep processing: it is a grammar for deep linguistic processing in as much as besides information on the major syntactic dimensions of grammatical constituency and dependency, it delivers (and generates from) fully-fledged logical representation of the meaning of natural language sentences.
- large-scale: it is planned not to leave out any sort of regular grammatical construction or phenomena.
- multi-purpose: it is intended to make available as much linguistic information as it can possible be made explicit by automatic means, given the current state of the art in language technology, with the goal of offering itself to support the largest possible range of language technology applications.

LXGram is being developed at the University of Lisbon, by NLX—Natural Language and Speech Group of the Department of Informatics, Faculty of Sciences, under the coordination of António Branco. Major coding work has been performed by Francisco Costa. The development activities benefited from support or contributions from Mariana Avelãs, Filipe Gil, Marco Gonzalez, Clara Pinto and David Raposo.

The development of LXGram has been undertaken with the support and in the scope of the Delph-in international consortium.

The research and development activities of LXGram were partially supported by FCT—Fundação para a Ciência e Tecnologia, of the Portuguese Ministry of Science, and to Instituto Camões, of the Portuguese Ministry of Foreign Affaires, under the research grant PLUS/PLP/50301/2003 for the project GramaXing. The former institution is also acknowledged for its partial support to the development of LXGram under the research grant RIPD/PSI/63557/2005 for the project SemanticShare.

For releases and licensing consult `http://nlx.di.fc.ul.pt/lxgram`.

Chapter 2

Introduction

2.1 Formalisms and Tools

LXGram is developed under the grammatical framework of Head-Driven Phrase Structure Grammar (HPSG: [Pollard and Sag, 1987], [Pollard and Sag, 1994], [Sag et al., 2003]) and uses Minimal Recursion Semantics (MRS: [Copestake et al., 2005]) for the representation of meaning.

This grammar implementation is undertaken with the LKB ([Copestake, 2002]) grammar development environment and its evaluation and regression testing is done via [incr tsdb()] ([Oepen, 2001]). It is also intended to be compatible with the PET parser ([Callmeier, 2000]).

The LinGO Grammar Matrix (version 0.9 [1]; [Bender et al., 2002]), an open-source kit for the rapid development of grammars based on HPSG and MRS, was used as the initial infrastructure upon which to build LXGram.

2.2 Implementation Agenda

The implementation of LXGram follows a list of phenomena aimed at being included in the grammar coverage. The implementation agenda is the following:

Phase A - Structure

1. Auxiliaries and basic phrase structure: S, VPs, PPs, APs, AdvPs
2. NPs (without relatives)
3. Predication structure and agreement
4. Modification structure

Phase B - Core Recursion

5. Completive subordination
6. Adverbial subordination
7. Punctuation
8. Coordination
9. Comparatives

Phase C - Reshuffling

10. Alternations: passive, middle, ...

[1] Initially, version 0.8 was used; at a later moment the coordination module provided in `http://depts.washington.edu/uwcl/HPSG2005/modules.html` was explored and then kept.

11. Raising
12. Control
13. Clitics I

Phase D - Long-distance Dependencies

14. Non-canonical word-order
15. Relatives
16. Interrogatives
17. Exclamatives and Imperatives

Phase E - Advanced Semantics

18. Negation
19. Clitics II
20. Tense and aspect
21. Determination and quantification

At this moment, Phase A is complete and some of Phase B has been implemented as well. Work is progressing to towards the implementation of further items in the agenda.

Chapter 3

Getting Started

The main files of LXGram are the following:

- Configuration files:
  - All files in the `lkb` directory
    Configuration files for the LKB. Loading the file `script` in the LKB loads the grammar.
  - `portuguese.tdl` and all files in the `pet` directory
    Configuration files for the PET parser.
- Type definitions:
  - `matrix-redefined.tdl`
    Type definitions adapted from the LinGO Grammar Matrix.
  - `features-types.tdl`
    Definitions of many types and features.
  - `head-types.tdl`
    Definitions of the type hierarchy of types appropriate for the attribute HEAD. It is not the `head-types.tdl` file that comes with the LinGO Grammar Matrix.
  - `lexical-types.tdl`
    Definitions of lexical types.
  - `syntax.tdl`
    Definitions of syntactic rules.
  - `lexical-rules.tdl`
    Definitions of non-spelling-changing lexical rules.
  - `morphology.tdl`
    Definitions of spelling-changing lexical rules.
  - `roots.tdl`
    Definition of start symbols.
  - `punctuation.tdl`
    Type definitions to control punctuation (see Section 5.9).
  - `lexicon.tdl`
    Lexical entries.

- Rules:
  - `rules.tdl`
    Names and inventory of syntactic rules.
  - `lrules.tdl`
    Names and inventory of non-spelling-changing lexical rules.
  - `irules.tdl`
    Names and inventory of spelling-changing lexical rules, as well as specification of those changes.
  - `irregs.tab`
    List of irregularities to spelling changes produced by the lexical rules.
- Additional files:
  - `labels.tdl`
    Definition of the labels associated with syntactic tree nodes.
  - `tree-decorations.tdl`
    Supertypes of the types in `labels.tdl`. These supertypes are used to factor out common behavior.
  - `preprocessor.fsr`
    Preprocessor rules (e.g. to undo contractions).
  - `lkb/mtr.tdl`
    Definitions of types used in the generator and paraphraser rules.
  - `trigger.mtr`
    Generator rules to select lexical items with no semantics in generation.
  - `trigger-abstract-types.tdl`
    Supertypes of the types in `trigger.mtr`. These supertypes are used to factor out common behavior.
  - `idioms.mtr`
    List of idiomatic expressions, used by the idiom detection mechanism (see Section 5.8).
  - `idioms-abstract-types.tdl`
    Supertypes of the types in `idioms.mtr`. These supertypes are used to factor out common behavior.
  - `paraphraser.mtr`
    Examples of rules that would enable the grammar to be used as a paraphraser (e.g. treating lexical synonymy).
  - All files in the `smaf` directory
    Configuration files for integrating the grammar with the shallow tools referred in Section 6.1, using SMAF. There are also several test examples in this directory.

[Copestake, 2002] is an introduction to the LKB system. The reader should consult it in order to know how to load and run an LKB grammar.

Chapter 4

Linguistic Coverage and Performance

At this point, LXGram size can be described with the numbers presented in Table 4.1.

Grammar Version	Lines of Code	Types (with GLBs)	Syntactic Rules	Lexical Entries	Lexical Types (Leaves)	Lexical Rules	Preprocessor Rules	Maintenance	Progress
A.1.2	10600	3069	28	1112	174	30	169		Phase A.1
A.4.1	24484	7787	53	2718	409	40	534		Phase A.4

Table 4.1: LXGram implementation progress.

The remaining tables describe LXGram's coverage, overgeneration and performance on two disjoint test suites: the test suite for the phase A.1 (see Section 2.2), containing 202 examples, and the test suite for the phases A.2, A.3 and A.4, with 851 items. These test suites are hand-crafted and contain many examples of the relevant phenomena, as well as many negative examples, that are to be rejected by the grammar.

We present the results for the version A.1.2 of the grammar (corresponding to the end of Phase A.1) and for the version A.4.1 of LXGram (at the end of phase A.4), for comparative purposes. The tables presenting these results are adapted from [incr tsdb()] output (the leftmost columns were removed and the table titles abridged).

Table 4.2 describes the coverage of LXGram on the data set for phase A.1, both under version A.1.2 and under version A.4.1. Coverage was maintained from version A.1.2 to version A.4.1. Ambiguity rose slightly (from an average of 2.48 parses per sentence to 2.93), increasing by 18%. Lexical ambiguity increased by 54% (from 59.11 to 90.84). Table 4.3 presents the coverage of version A.4.1 on the test suite for the phase A.4.

Table 4.4 and Table 4.5 display the values for overgeneration. The version A.4.1 is overgenerating for the phase A.1 test suite. The main source of overgeneration is sequences that are impossible as an adjective phrase receiving a parse as a noun phrase with a missing noun. One example is the following, where "mais bom" is intended to mean "melhor" (*better*):

(1)	* Este	computador	é	mais	bom.
	this	computer	is	more	good

Phase A.1 Test Suite
LXGram Version A.1.2

Coverage						
total items ♯	**positive items** ♯	**word string** ϕ	**lexical items** ϕ	**distinct analyses** ϕ	**total results** ♯	**overall coverage** %
202	**122**	**6.86**	**59.11**	**2.48**	**122**	**100.0**

Phase A.1 Test Suite
LXGram Version A.4.1

Coverage						
total items ♯	**positive items** ♯	**word string** ϕ	**lexical items** ϕ	**distinct analyses** ϕ	**total results** ♯	**overall coverage** %
202	**122**	**6.86**	**90.84**	**2.93**	**122**	**100.0**

Table 4.2: LXGram coverage evolution on the phase A.1 test suite.

Phase A.4 Test Suite
LXGram Version A.4.1

Coverage						
total items ♯	**positive items** ♯	**word string** ϕ	**lexical items** ϕ	**distinct analyses** ϕ	**total results** ♯	**overall coverage** %
851	**422**	**5.31**	**33.55**	**1.15**	**422**	**100.0**

Table 4.3: LXGram coverage evolution on the phase A.4 test suite.

Although the expression "mais bom" is ungrammatical with "mais" as an adverb, it can receive a parse as an NP (with "mais" as a determiner), because of examples like:

(2) Queremos menos vinho mau e mais bom.
we want less wine bad and more good
We want less bad wine and more of the good one.

With the version A.1.2 of the grammar, sentences like (1) were ruled out. The version A.4.1 contains an implementation of noun ellipsis (see Section 8.7) that produces a parse for these sentences. However, phrases like "mais bom" keep not being analyzed as adjective phrases.

Efficiency measures are also provided in Table 4.6 and Table 4.7. In Table 4.6 we can see that the average time needed to analyze a sentence in the phase A.1 test suite went up from 0.06 seconds in version A.1.2 to 0.22 seconds in version A.4.1. The amount of memory required also increased from 14.2MB to 27.5MB. These numbers reflect the higher number of rules in the version A.4.1 of LXGram as well as the increased lexical ambiguity.

Phase A.1 Test Suite						
LXGram Version A.1.2						
Overgeneration						
total items ♯	**negative items** ♯	**word string** ϕ	**lexical items** ϕ	**distinct analyses** ϕ	**total results** ♯	**overall coverage** %
202	**80**	**8.90**	**82.59**	**0.00**	**0**	**0.0**

Phase A.1 Test Suite						
LXGram Version A.4.1						
Overgeneration						
total items ♯	**negative items** ♯	**word string** ϕ	**lexical items** ϕ	**distinct analyses** ϕ	**total results** ♯	**overall coverage** %
202	**80**	**8.90**	**128.00**	**1.14**	**14**	**17.5**

Table 4.4: LXGram overgeneration evolution on the phase A.1 test suite.

Phase A.4 Test Suite						
LXGram Version A.4.1						
Overgeneration						
total items ♯	**negative items** ♯	**word string** ϕ	**lexical items** ϕ	**distinct analyses** ϕ	**total results** ♯	**overall coverage** %
851	**429**	**5.38**	**33.74**	**0.00**	**0**	**0.0**

Table 4.5: LXGram overgeneration evolution on the phase A.4 test suite.

Phase A.1 Test Suite						
LXGram Version A.1.2						
Performance						
items ♯	**etasks** ϕ	**filter** %	**edges** ϕ	**first** ϕ (s)	**total** ϕ (s)	**space** ϕ (kb)
202	**282**	**92.5**	**172**	**0.07**	**0.06**	**14516**

Phase A.1 Test Suite						
LXGram Version A.4.1						
Performance						
items ♯	**etasks** ϕ	**filter** %	**edges** ϕ	**first** ϕ (s)	**total** ϕ (s)	**space** ϕ (kb)
202	**1374**	**95.1**	**314**	**0.23**	**0.22**	**28136**

Table 4.6: LXGram efficiency evolution on the phase A.1 test suite.

Phase A.4 Test Suite
LXGram Version A.4.1

Performance						
items	**etasks**	**filter**	**edges**	**first**	**total**	**space**
♯	ϕ	%	ϕ	ϕ (s)	ϕ (s)	ϕ (kb)
848	**733**	**95.0**	**167**	**0.12**	**0.11**	**15843**

Table 4.7: LXGram efficiency evolution on the phase A.4 test suite.

Chapter 5

General Aspects of the Implementation

In the present chapter some of the strategies followed in the implementation of LXGram are described. For instance, there is often a concern to reduce feature structures (Section 5.1). Section 5.2 and Section 5.3 describe departures from the LinGO Grammar Matrix. The remaining sections describe some aspects of the implementation that have an impact on the general organization of the features employed in the grammar.

5.1 Reduction of Feature Structures via Type Inference and Polymorphism

Throughout the implementation of LXGram a technique described in [Flickinger, 2000] and referred to as *minimal types* is used in order to reduce the size of the feature structures that the parser will operate on at runtime. The use of minimal types is employed in the ERG and in the LinGO Grammar Matrix.

An example follows. The features SLASH, REL and QUE are needed under NON-LOCAL, so an initial approach could be to create a type *non-local* for which these features are declared appropriate and, since NON-LOCAL is a feature of synsems, declare the type *synsem* to have a feature NON-LOCAL of type *non-local*.

The minimal types approach would be different. One still has type *non-local*, where all these features are declared, but one would also create an additional type *non-local-min*, which has no features, and declare *non-local* to inherit from *non-local-min*. Furthermore, the feature NON-LOCAL would now be declared in *synsem* to be of type *non-local-min*. The following figures show the contrast between not using (Figure 5.1) and using (Figure 5.2 and Figure 5.3) the *minimal types* approach.

The LKB supports polymorphism, so the feature NON-LOCAL can be instantiated with a feature structure of the type *non-local*, since it is a subtype of *non-local-min*.

$$\begin{bmatrix} \textit{synsem} & \\ \text{LOCAL} & \textit{local-min} \\ \text{NON-LOCAL} & \begin{bmatrix} \textit{non-local} & \\ \text{SLASH} & \textit{0-1-dlist} \\ \text{QUE} & \textit{0-1-dlist} \\ \text{REL} & \textit{0-1-dlist} \end{bmatrix} \end{bmatrix}$$

Figure 5.1: *Synsem* without a minimal type for attribute NON-LOCAL

$$\begin{bmatrix} \textit{synsem} & \\ \text{LOCAL} & \textit{local-min} \\ \text{NON-LOCAL} & \textit{non-local-min} \end{bmatrix}$$

Figure 5.2: *Synsem* with a minimal type for attribute NON-LOCAL

$$\begin{tikzcd} \textit{non-local-min} \arrow[d, dash] \\ \begin{bmatrix} \textit{non-local} & \\ \text{SLASH} & \textit{0-1-dlist} \\ \text{QUE} & \textit{0-1-dlist} \\ \text{REL} & \textit{0-1-dlist} \end{bmatrix} \end{tikzcd}$$

Figure 5.3: Example type hierarchy under *non-local-min.*

Furthermore, the LKB requires for each feature there to be a single most general type for which it is appropriate; if such a constraint is not respected there is an error at grammar load time. This requirement enables the system to support type inference. With type inference, one does not need to mention explicitly that a particular instance of the feature NON-LOCAL is of the type *non-local* when one wants to use these features. Whenever a constraint mentions the feature SLASH, QUE or REL, the type *non-local* is inferred for that instance of the feature NON-LOCAL.

The main advantage of using minimal types is that it can make feature structures smaller. Continuing with the NON-LOCAL example, the features SLASH, QUE and REL will appear in feature structures only if they are mentioned in some constraint (e.g. in general they will not appear in synsems in a MOD or SPEC list, as these are not used in the amalgamation of non-local features).

This fact will have a positive impact on performance, as it spares some unification operations.

A nice side effect is that, since features that are not being used are not present in the resulting AVMs, these become more readable as they are smaller and redundant information is hidden.

As noted above, the ERG and the LinGO Grammar Matrix use this technique. In fact, the types *non-local-min* and *non-local* are defined in both of them in a way similar to the one presented.

The PET parser manipulates feature structures with the purpose of removing redundant features automatically at run-time (*unfilling*, see [Callmeier, 2000]), which is in fact equivalent to using minimal types in the grammar. However, [Flickinger, 2000] reports that the use of minimal types in a grammar can still improve parser performance even when the grammar is used with a parser that *unfills* feature structures.

In LXGram, this technique is also employed. Furthermore, instead of creating a single abstract type and a more specific one where all the features that are needed are introduced at once, sometimes an extra abstract type for each feature is created (where that feature is introduced), and all possible combinations of subtypes of these are also created.

Therefore, for each feature in a feature structure, it will be present there if and only if some constraint mentions it, and since there is a subtype for each combination of features, the technique will never cause unification failures by itself, thus having no impact on correctness.

This technique can be illustrated with the type hierarchy under *png*, where the features PERSON, NUMBER and GENDER are necessary (Figure 5.4). These features are the place where the information about morphological gender, number and person is represented in the grammar. For instance, the type hierarchy presented allows for a *png* object underspecified for gender not to have an attribute GENDER, instead of having one with its most general type *gender*.

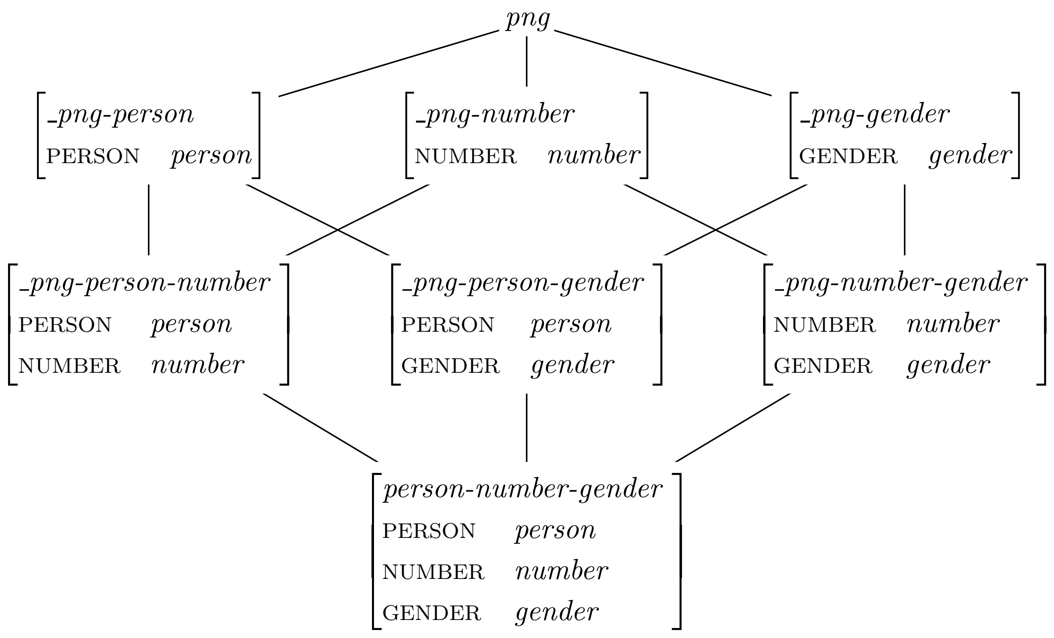

png
_png-person PERSON person
_png-number NUMBER number
_png-gender GENDER gender
_png-person-number PERSON person NUMBER number
_png-person-gender PERSON person GENDER gender
_png-number-gender NUMBER number GENDER gender
person-number-gender PERSON person NUMBER number GENDER gender

Figure 5.4: Type hierarchy under *png* with expanded constraints. Features PERSON, NUMBER and GENDER are declared in the types *_png-person*, *_png-number* and *_png-gender* respectively.

5.2 Changes to the LinGO Grammar Matrix

The initial policy in the implementation of LXGram was to not change the code in the LinGO Grammar Matrix. As the grammar grew, however, certain changes happened to be required.

The main motivation for not changing the Matrix setup is that it makes it easier to upgrade the grammar to new versions of the Matrix. Also, not changing it means that it is guaranteed that the construction of semantics is done as in the other grammars that use the LinGO Grammar Matrix, which helps keep semantic representations consistent and uniform across grammars.

LXGram is growing faster than the Matrix is being updated, so the first advantage has had no practical consequences so far, and one can also make semantic representations similar to the ones created by Matrix grammars without having to use the Matrix unchanged.

As the changes accumulated over time, we decided to abandon the original `matrix.tdl` file and include the changed type definitions in a new file, called `matrix-redefined.tdl`.

The next section documents an important departure from the LinGO Grammar Matrix.

5.3 Functors

In LXGram we replace the Head-Specifier and Head-Adjunct configurations with the Functor-Head schemata of [Allegranza, 1998a], [Allegranza, 1998b], [Van Eynde, 2003a] and [Van Eynde, 2003b].

In [Allegranza, 1998a], [Allegranza, 1998b], [Van Eynde, 2003a] and [Van Eynde, 2003b], Head-Functor relations are a cover of Head-Specifier and Head-Adjunct configurations. Functors, like adjuncts, select their head via a dedicated feature. All treatments put this feature that encodes selection requirements under the attribute HEAD, but the name varies (here it is SELECT). As a consequence, this feature SELECT percolates in all headed constructions. Like Head-Specifier configurations, information about saturation of the resulting node (i.e. its combinatorial potential) may be different from the combinatorial potential of the head daughter. So, for instance, a noun can combine with a determiner on the left to form a phrase with a determiner and noun, and this phrase cannot combine with another determiner on the left.

In Head-Specifier configurations, information about this kind of combinatorial potential is determined by the head noun, not by the specifier. Nouns select their specifier in their SPR attribute, which is list-valued (and usually either the empty list or a singleton list). Head-Specifier constructions unify the synsem in that attribute with the synsem of the other daughter, and the SPR of the mother node is the tail of the SPR list in the head daughter.

In Head-Functor schemata, saturation of the mother node is determined by the functor daughter, not the head (details are below).

With functors replacing specifiers and adjuncts, the features MOD, SPEC and SPR are no longer necessary, and neither are Head-Specifier phrases and Head-Adjunct phrases.

LXGram uses the features SELECT, MARKING and MARK to implement Head-Functor schemata. MARKING is used to describe combinatorial potential other than saturation of a head's arguments. For instance, since cardinals cannot iterate, the top node of a phrase like *three cars* would have a different value of MARKING from the one of *cars*.

The features MARK and SELECT are relevant for functors. The value of MARKING of the mother node in Head-Functor phrases comes from the MARK attribute of the functor daughter. The attribute SELECT is where functors state the heads they can attach to.

The main properties of Head-Functor schemata are presented in Figure 5.5.

$$\begin{bmatrix} \textit{basic-head-functor-phrase} \\ \text{SYNSEM}|\text{LOCAL}|\text{CAT} \begin{bmatrix} \text{HEAD} & \boxed{1} \\ \text{VAL}|\text{SUBJ} & \boxed{2} \\ \text{MARKING} & \boxed{3} \end{bmatrix} \\ \text{HEAD-DTR}|\text{SYNSEM } \boxed{4} \begin{bmatrix} \text{LOCAL}|\text{CAT} \begin{bmatrix} \text{HEAD} & \boxed{1} \\ \text{VAL}|\text{SUBJ} & \boxed{2} \end{bmatrix} \end{bmatrix} \\ \text{NON-HEAD-DTR}|\text{SYNSEM}|\text{LOCAL}|\text{CAT}|\text{HEAD}|\text{MARKER} \begin{bmatrix} \text{MARK} & \boxed{3} \\ \text{SELECT} & \boxed{4} \end{bmatrix} \end{bmatrix}$$

Figure 5.5: Outline of Head-Functor schemata

It is a headed construction, so the HEAD of the mother is token-identical to the HEAD of the head daughter. This constraint — the Head Feature Principle — is inherited from a supertype coming from the LinGO Grammar Matrix, *headed-phrase*, which implements this principle.

The subject of the mother and the subject of the head daughter are also shared since functors do not discharge the subject of the head. In LXGram the COMPS feature of the mother node can be the COMPS of the head daughter or the COMPS of the non-head daughter (see below).

Because functors have access to information about valence and marking of the head (the functor daughter's SELECT feature is unified with the head daughter's SYNSEM), control on the level of saturation of the head in these constructions is reduced to lexical specifications in functors.

Functors can cause the saturation described by the feature MARKING on the mother node to be different from the one on the head daughter — the mother node's MARKING feature is structure-shared with the functor's MARK feature.

LXGram contains three functor-head rules inheriting from *basic-head-functor-phrase*. One projects a functor to the right of the head (*head-functor-phrase*), which identifies the COMPS attribute of the resulting phrase with that of the head daughter. The other two project a functor to the left of its head. One of these identifies the complements of the mother with those of the head daughter (*functor-head-hcomps-phrase*), the other percolates the complements of the functor (*functor-head-fcomps-phrase*). Sentence (3), where the "(do) que" (*than*) phrase is analyzed as a complement of "mais" (*more*), illustrates the need for these three rules. A partial parse tree

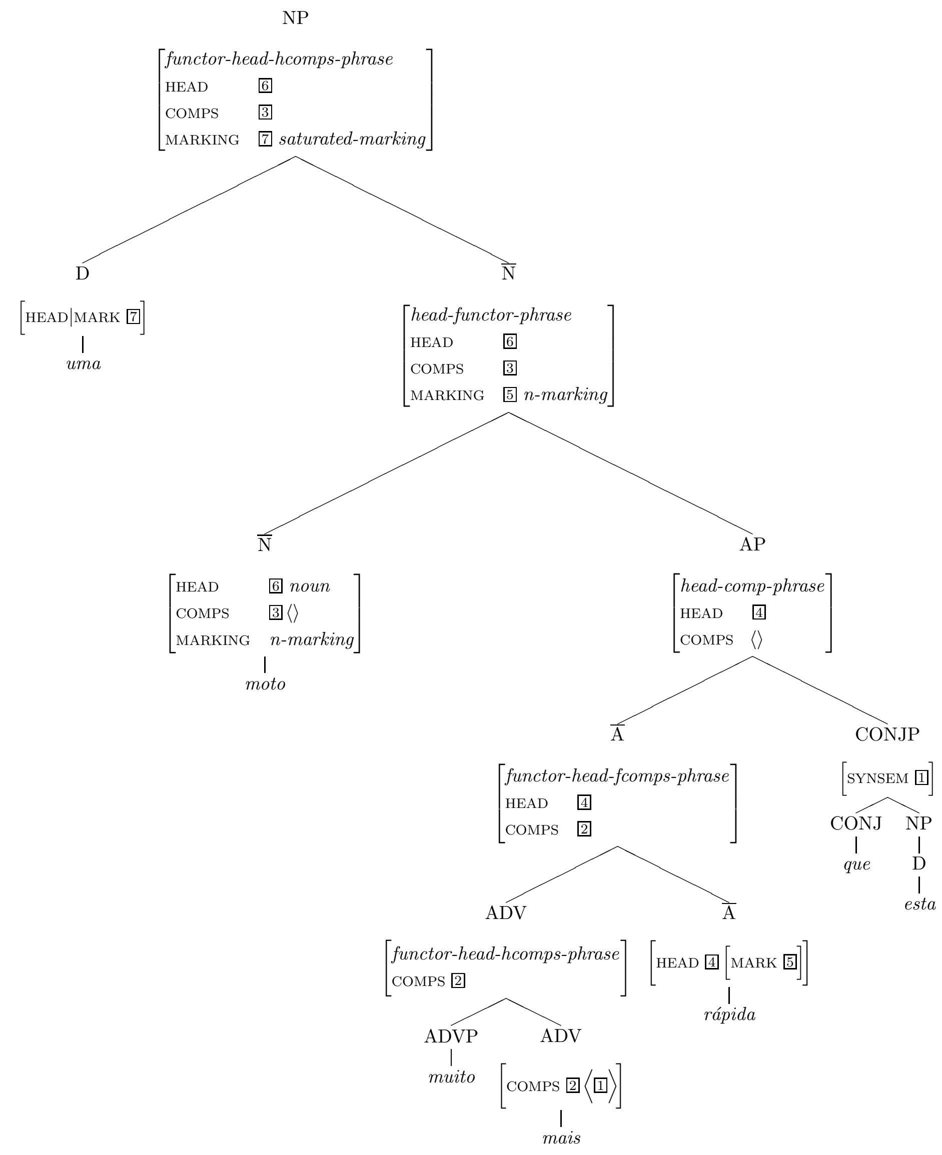


Figure 5.6: Parse tree for a phrase with all functor-head rules. The phrase is “uma moto muito mais rápida que esta” (*a motorcycle much faster than this one*). Feature paths are shortened.

is presented in Figure 5.6, where each node is decorated with the corresponding rule (the values that are presented here for the attribute MARKING are explained in Chapter 8).

(3)

Tenho	uma	moto	muito	mais	rápida	(do) que	esta.
I have	a	mortorcycle	much	more	fast	than	this one

I have a motorcycle much faster than this one.

This example also illustrates the purpose of the features MARKING and MARK. What distinguishes the NP node from the $\overline{\text{N}}$ nodes in this example is the value of the feature MARKING, which comes from the MARK feature of the functor daughter.

Also, there are actually two versions for each of these three rules (so a total of six rules): a version for intersective modifiers, another version for scopal modifiers and specifiers. The purpose of this splitting is explained in the next subsection.

5.3.1 Composition of Semantics in Head-Functor Phrases

The composition of semantics comes as expected: as all semantic information comes from the daughters, the RELS and HCONS of the mother are simply the difference list append of the homonymous features of both daughters.

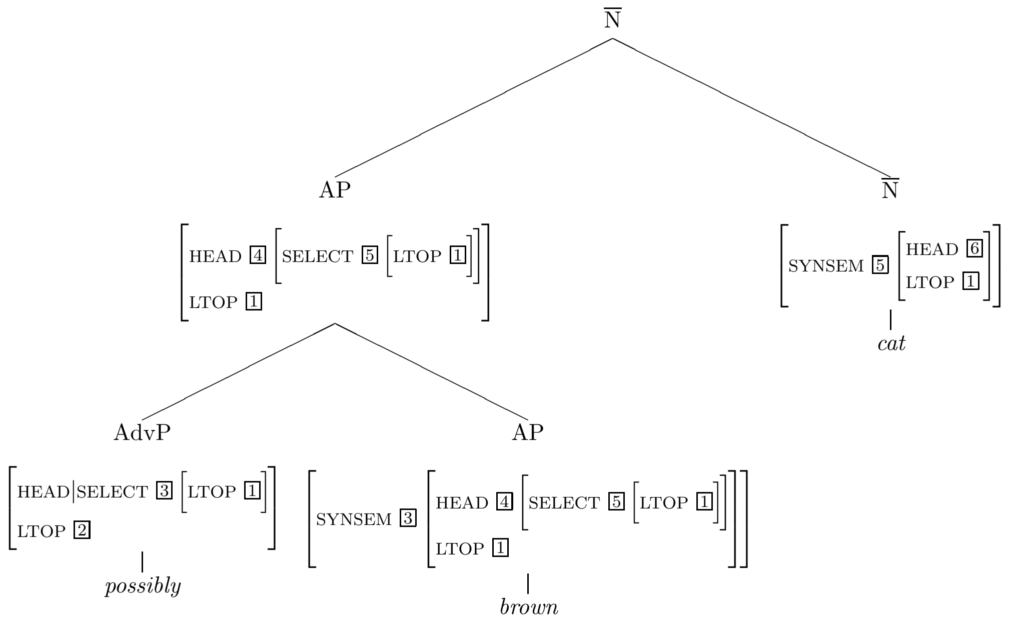
As discussed elsewhere [Kasper, 1996], [Copestake et al., 2005], there are issues (concerning the feature LTOP) regarding the interaction between intersective and scopal modifiers. The next paragraphs describe the problem at hand.

Consider an example like *possibly brown cat*, where the adjective *brown* is an intersective modifier of the noun *cat*, and the adverb *possibly* is a scopal modifier of the adjective *brown*.

Assume that intersective modifiers unify their LTOP feature with the LTOP of the synsem they select via their SELECT attribute in their lexical entries, so that the relations for the modifying element and the modified one end up with the same LBL in the MRS (as this situation denotes conjunction of the two relations). Note that in general the LTOP of a lexical entry will be unified with the LBL of a relation in that item's RELS, so unifying LTOPs amounts to unifying LBLs in MRSs.

As for scopal modifiers (as well as determiners, which now, as functors, combine via the same rules), they do not identify these values but rather use the LTOP of the selected constituent as the value for one of the arguments of the relation they introduce, often mediated by a *qeq* constraint in the functor's HCONS — and all of this is done in the lexical entries for functors.

This yields the wrong semantics for phrases like *possibly brown cat*. What is produced is equivalent to $\lambda x.possible(brown(x) \wedge cat(x))$, but what is correct is $\lambda x.possible(brown(x)) \wedge cat(x)$ (feature paths are shortened in the derivation tree):


N̄
AP
HEAD 4 [SELECT 5 [LTOP 1]]
LTOP 1
N̄
SYNSEM 5 [HEAD 6, LTOP 1]
cat
AdvP
HEAD|SELECT 3 [LTOP 1]
LTOP 2
possibly
AP
SYNSEM 3 [HEAD 4 [SELECT 5 [LTOP 1]], LTOP 1]
brown

As can be seen, because LTOPs are unified in the lexicon, semantic scope and syntactic scope do not match.

There are two solutions in the literature. One is in [Copestake et al., 2005]: LTOP features are not unified in the lexicon, but in the syntax rules. This requires separate rules for intersective modification (where LTOP attributes are unified) and scopal modification (where these features are not unified). The other is in [Kasper, 1996]: in order to obtain the desired result, the number of features used for the composition of semantics is enriched.

In LXGram we follow the first solution. Rule application is controlled by a boolean feature under MARKER, called SCOPAL. Scopal modifiers come in the lexicon with the value + for this

$$\begin{bmatrix} \textit{head-initial} & \\ \text{HEAD-DTR} & \boxed{1} \\ \text{NON-HEAD-DTR} & \boxed{2} \\ \text{ARGS} & \left\langle \boxed{1}, \boxed{2} \right\rangle \end{bmatrix} \begin{bmatrix} \textit{head-final} & \\ \text{HEAD-DTR} & \boxed{1} \\ \text{NON-HEAD-DTR} & \boxed{2} \\ \text{ARGS} & \left\langle \boxed{2}, \boxed{1} \right\rangle \end{bmatrix}$$

Figure 5.7: Implementation of word order in binary headed phrases

feature, and intersective modifiers have the type - here. A set of Head-Functor rules requires functor daughters with a SCOPAL feature of type - and identifies the LTOP feature of both daughters. Another set of rules constrains the functor daughter with the other value for this feature and places no constraints on the LTOP features.

5.3.2 Word Order in Head-Functor Phrases

In the LKB, the actual daughters of a rule are configured with the parameter ***args-path***. Its value is usually ARGS, a feature of *sign* instances.[1] It must be list-valued (it is a list of *sign*s), and the position of an element in that list correlates with linear precedence. In many computational grammars and in the LinGO Grammar Matrix, features like the attribute HEAD-DTR used in Figure 5.5 can be viewed as pointers to specific elements of that list.

It is often the case that abstract types for phrases are employed to constrain these pointer features, and then different subtypes implement different word order possibilities by linking them to different elements in ARGS. Figure 5.7 shows the constraints for two abstract types that define word order between the head daughter and the non-head daughter in binary headed phrases: *head-initial* constrains the head daughter to precede the non-head daughter, and *head-final* defines the non-head daughter to precede the head daughter. These types are in the LinGO Grammar Matrix.

The essential properties of Head-Functor phrases are stated in the abstract type *basic-head-functor-phrase*, already presented in Figure 5.5. Two subtypes implement the two word order possibilities between the daughters, by inheriting from *head-initial* or *head-final*. The result is shown in Figure 5.8. All headed phrases with two daughters inherit from *basic-binary-headed-phrase*.

As stated, all functors can feed both sets of rules. This situation is not desirable, since specific pairings of head and functor can be restricted to occur in a specific order.

For instance, a preposition comes in the lexicon with the information that it must attach to an NP on its right (forming a PP)[2]and then modify nouns and verbs. PPs can attach to either side of a verb headed constituent (4a, 4b), but only to the right of nouns (4c, 4d).

(4) a.

Isso	sai	[$_{PP}$ com	benzina.]
that	goes away	with	benzine

That goes away with benzine.

b.

Isso	[$_{PP}$ com	benzina]	sai.
that	with	benzine	goes away

That goes away with benzine.

[1]The feature ARGS is declared in the LinGO Grammar Matrix as appropriate for *signs*, but it does not make sense to speak of the ARGS of lexical items, so in LXGram ARGS is declared in type *phrase-or-lexrule* instead. This type also comes in the LinGO Grammar Matrix. It is the supertype of all phrases and lexical rules (which for instance account for morphological inflection), but not of lexical items.

[2]This is achieved by constraining the element in the COMPS of prepositions to be of type *canonical-synsem*, as in Portuguese complements of prepositions cannot be null (hence they cannot unify with *unexpressed-synsem*), and they cannot be extracted, either (hence they must be incompatible with *gap*).

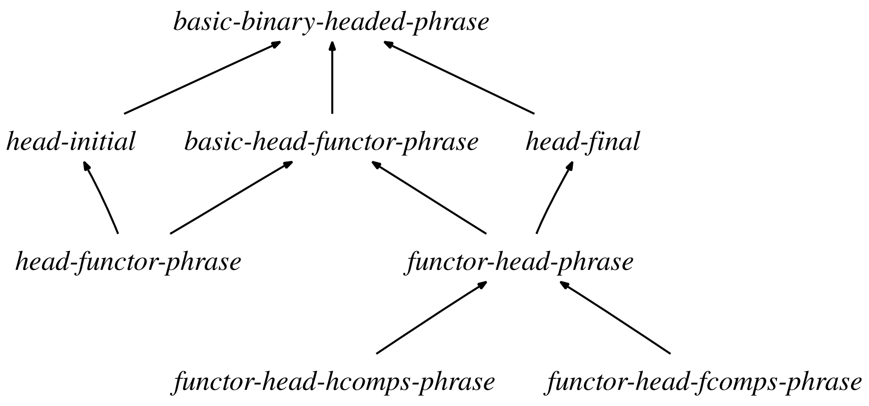

basic-binary-headed-phrase
head-initial
basic-head-functor-phrase
head-final
head-functor-phrase
functor-head-phrase
functor-head-hcomps-phrase
functor-head-fcomps-phrase

Figure 5.8: Organization of Head-Functor phrases.

c. Era um chapéu [PP com uma antena.]
was a hat with an antenna

It was a hat with an antenna.

d. * Era um [PP com uma antena] chapéu.
was a with an antenna hat

The way word order is controlled is by using more features to denote word order restrictions. These restrictions are seen as properties of functors, i.e. it is assumed that word order restrictions are lexical properties of functors.

The two features that are used are also under MARKER: PREHEAD and POSTHEAD. They contain constraints that must be satisfied when a functor precedes or follows the head daughter, respectively. These constraints are put on SELECT and MARK attributes under PREHEAD and POSTHEAD. The basic organization of features under MARKER is in Figure 5.9. The AVM in this figure does not correspond to a type definition. It is rather a schematic view of the features that are used.

$$\begin{bmatrix} \text{MARKER} & \begin{bmatrix} \text{SELECT} & \textit{synsem} \\ \text{MARK} & \textit{marking} \\ \text{PREHEAD} & \begin{bmatrix} \text{SELECT} & \textit{synsem} \\ \text{MARK} & \textit{marking} \end{bmatrix} \\ \text{POSTHEAD} & \begin{bmatrix} \text{SELECT} & \textit{synsem} \\ \text{MARK} & \textit{marking} \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

Figure 5.9: Organization of the features under the HEAD of functors.

The head type for prepositions has the constraints in Figure 5.10, where *noun-or-verb* is a supertype of *verb* and *noun*.

In order for these constraints to play the intended role, the two types *head-functor-phrase* and *functor-head-phrase*, depicted in Figure 5.8, must be further refined. Their definitions are in Figure 5.11.

Because the higher SELECT attribute is already unified with the SYNSEM of the head daughter, and the higher MARK with the MARKING of the mother node, the homonymous features under PREHEAD and POSTHEAD will also be unified with these, but only when the relevant syntax rule is used.

$$\begin{bmatrix} \textit{preposition} & \\ \text{MARKER} & \begin{bmatrix} \text{SELECT|LOCAL|CAT|HEAD} & \textit{noun-or-verb} \\ \text{PREHEAD|SELECT|LOCAL|CAT|HEAD} & \textit{verb} \end{bmatrix} \end{bmatrix}$$

Figure 5.10: Constraints on the HEAD of prepositions. *noun-or-verb* is a supertype of *noun* and *verb*, the head types of nouns and verbs, respectively.

$$\begin{bmatrix} \textit{head-functor-phrase} & \\ \text{NON-HEAD-DTR|SYNSEM|LOCAL|CAT|HEAD|MARKER} & \begin{bmatrix} \textit{post-marker} & \\ \text{SELECT} & \boxed{1} \\ \text{MARK} & \boxed{2} \\ \text{POSTHEAD} & \begin{bmatrix} \text{SELECT} & \boxed{1} \\ \text{MARK} & \boxed{2} \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

$$\begin{bmatrix} \textit{functor-head-phrase} & \\ \text{NON-HEAD-DTR|SYNSEM|LOCAL|CAT|HEAD|MARKER} & \begin{bmatrix} \textit{pre-marker} & \\ \text{SELECT} & \boxed{1} \\ \text{MARK} & \boxed{2} \\ \text{PREHEAD} & \begin{bmatrix} \text{SELECT} & \boxed{1} \\ \text{MARK} & \boxed{2} \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

Figure 5.11: Constraints on *head-functor-phrase* and *functor-head-phrase*

In LXGram the type of the feature MARKER controls whether specific functors can feed head-initial and head-final Head-Functor constructions. An element with this feature constrained to be of the type *pre-only-marker* cannot feed the head-initial Head-Functor rule, because in this rule the functor daughter is constrained to have a feature MARKER with a type incompatible with *pre-only-marker*. This incompatible type is a type subsumed by *post-marker*. An element with the value *post-only-marker* for this feature is not elligible to be the functor daughter of head-final Head-Functor phrases, either, for similar reasons (the head-final Head-Functor phrases constrain the MARKER of their functor daughter to be of a type subsumed by *pre-marker*). The type hierarchy for these values is in Figure 5.12.

The hierarchy presented in this figure is simplified in that minimal types, as presented in Section 5.1, are also used in the grammar, but they are not shown in this figure.

$$\begin{tikzcd}
 & & \textit{marker} & & \\
 & \textit{post-marker} \arrow[ur] & & \textit{pre-marker} \arrow[ul] & \\
\textit{post-only-marker} \arrow[ur] & & \textit{pre-or-post-marker} \arrow[ul] \arrow[ur] & & \textit{pre-only-marker} \arrow[ul]
\end{tikzcd}$$

Figure 5.12: Values for the feature MARKER (simplified hierarchy).

5.4 Sentence Force

Like other DELPH-IN grammars, LXGram no longer produces message relations in MRSs. Instead, a feature SF (SENTENCE FORCE) in events holds information on illocutionary force. Declarative sentences correspond to MRSs where the main event has an SF feature with value *proposition*, interrogative sentences give rise to MRSs with an event with a *question* SF and so on.

At the begininng of 2007 a new version of the LinGO Grammar Matrix appeared where message relations had been replaced with this encoding of illocutionary force. This change had already been implemented in LXGram.

The places of the grammar that were affected by this change are similar to the ones reported in the message-free LinGO Grammar Matrix announcement. The main differences are the following. First, subject-verb inversion does not distinguish between declarative sentences and yes-no questions in Portuguese and realization of the subject does not separate imperatives from other sentences, so we do not need to constrain this feature in subject-head constructions. The second difference is that we are introducing a *qeq* constraint between the topmost LTOP (assuming it denotes the global top) and the main verb's label (LBL), following the literature on MRS ([Copestake et al., 2001]), while the LinGO Grammar Matrix implementation relies on a modification of the scope resolution algorithm in the LKB. Our approach requires a unary syntactic rule to produce the root node of all syntactic trees, for the sole purpose of adding this *qeq* constraint, but the grammar is producing MRSs that conform to the MRS literature.

The type hierarchy used in LXGram for the possible values of the feature SF is very similar to the one in the LinGO Grammar Matrix. We use more verbose names, and there is an extra type that allows underspecification between propositions and commands (for sentences that end with a period). Figure 5.13 shows this hierarchy.

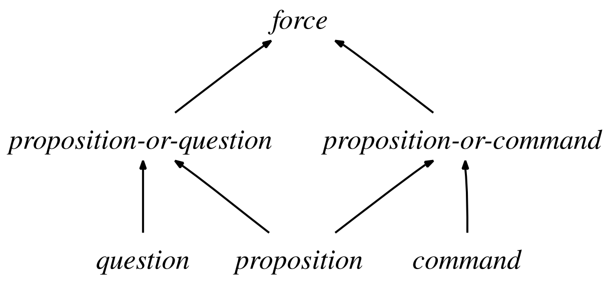


Figure 5.13: Type hierarchy under *force*.

5.5 Variation Between European and Brazilian Portuguese

In LXGram variation between European Portuguese (EP) and Brazilian Portuguese (BP) is taken into account. A mechanism is implemented in order to control grammar behaviour regarding variation between EP and BP.

Currently, only a two-fold distinction is made among the varieties of Portuguese, namely a coarse-grained distinction between EP and BP, although regional differences can obviously be found in the Portuguese spoken in Africa or the rest of the world or in different regions of Portuguese speaking countries. This will be expanded as needed.

The mechanism can be used for several purposes:

1. to restrict the grammar to parse or generate EP sentences only or BP sentences only, or to accept EP and BP but reject sentences showing specific features of both;

2. to use the grammar to detect which variety is used in some text (by parsing it and reading from the resulting feature structures).

It is important that the grammar can work with both EP and BP because of coverage, but accepting the two will necessary increase ambiguity. The ability to control variation is important in that it is a way to control the ambiguity generated from accepting both varieties. It is also very desirable when generating.

The implementation is very simple and is inspired on the idiom detection mechanism implemented in the Jacy grammar and described at http://wiki.delph-in.net/moin/JacyIdiom (see also Section 5.8). A brief introduction to it is presented below. For a detailed account, see [Branco and Costa, 2007].

We make use of a feature VARIANT, which encodes the variety of Portuguese being used. It is appropriate for all signs and declared to be of the type *variant*. Its possible values are presented in Figure 5.14.

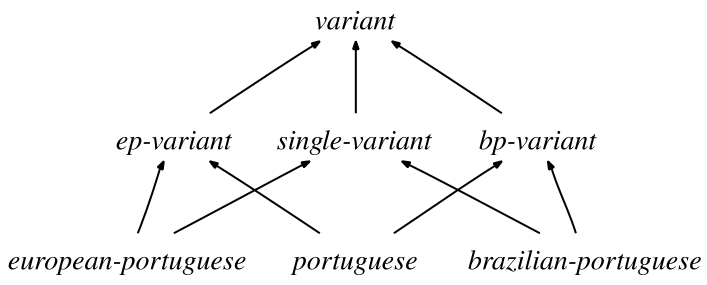


Figure 5.14: Type hierarchy under *variant*.

This attribute is constrained to take the appropriate value in lexical items and constructions specific to one of the two main Portuguese varieties. For example, the lexical entry for the noun *autocarro* (*bus*, exclusive to EP) includes the constraint that the attribute VARIANT has the value *ep-variant* and the corresponding BP entry for *ônibus* constrains the same feature to bear the value *bp-variant*. The only two types that are used to mark signs are *ep-variant* and *bp-variant*. The remaining types presented in Figure 5.14 are used to perform computations or to constrain grammar behaviour, as explained below.

The feature VARIANT is structure-shared among all signs that comprise a full parse tree. This is achieved by having all lexical rules unify their VARIANT feature with the VARIANT feature of their daughter, and all syntax rules identify their VARIANT feature with the VARIANT feature of every daughter.

If two signs (lexical items, syntax rules) in the same parse tree have different values for feature VARIANT (one has *ep-variant* and the other *bp-variant*), they will unify to *portuguese*, as can be seen in Figure 5.14. This type means that lexical items or constructions specific to two different varieties are used together. Furthermore, since this feature is shared among all signs, it will be visible everywhere, for instance in the root node.

It is possible to constrain the feature VARIANT in the root condition of the grammar (i.e. in the definition for the type that corresponds to the LKB's ***start-symbol*** parameter and specifies the constraints that the root node of a valid parse must satisfy). If this feature is constrained to be of the type *single-variant*, the grammar will accept either EP or BP, but the sentences with properties of both will be blocked. As explained in the previous paragraph, the feature VARIANT will have the value *portuguese* in this case, a value incompatible with *single-variant*. If this feature is constrained to be of the type *european-portuguese*, the grammar will not accept any sentence with features of BP, since they will be marked to have a VARIANT of the type *bp-variant*, which is incompatible with *european-portuguese*. It is also possible to have the grammar reject EP (using type *brazilian-portuguese*) or to ignore variation completely by not constraining this feature in the start symbol.

It is not practical to have to change the grammar code in order to be able to switch among variation policies, and that would also require reloading the grammar for changes to take effect. For this reason, the most convenient way of changing the variation control policy followed by the grammar is to change the LKB parameter ***start-symbol***. To make it simple to change, we provide four root types that correspond to the four types of behavior we anticipate. They are all defined in the file `roots.tdl` and are presented in Figure 5.15. They have the common supertype *root-clause*, which encapsulates the remaining constraints on valid root nodes.

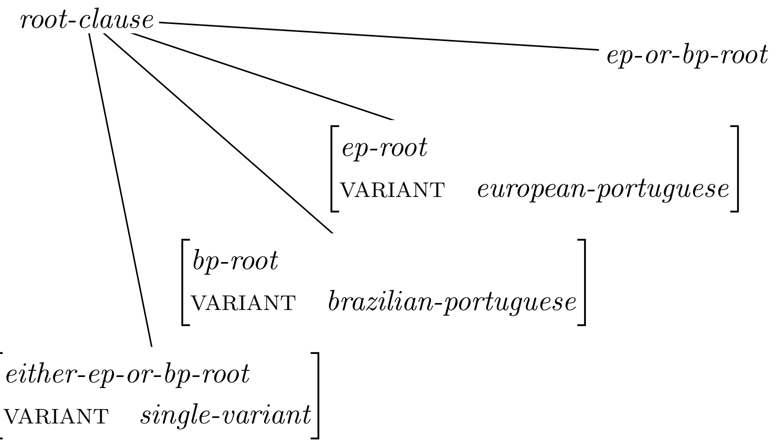


Figure 5.15: Types defining variation control policies.

Choosing *ep-and-bp-root* as root condition has the effect that no constraint is put on variants. Choosing the type *only-ep-root* makes the grammar reject all sentences with features of BP; *only-bp-root* makes the grammar reject all sentences with features of EP. Type *only-ep-or-only-bp-root* constrains it to accept EP and BP, but block sentences with properties of both.

In the LKB configuration file `lkb/globals.lsp`, ***start-symbol*** is initialized with the value *only-ep-or-only-bp*, so by default the grammar will accept either EP or BP but reject sentences with variant switching.

It is possible to change the value of the parameter ***start-symbol*** in the Lisp buffer by invoking **(defparameter *start-symbol* '(only-ep-root))**, for instance. Any of the four types just presented will be a valid choice. It is not necessary to reload the grammar for the change to take effect.

It is important to note that the LKB does not unify a type defined as a ***start-symbol*** with the feature structures resulting from parsing or generation. Instead, it only checks if they are unifiable. This means that if the default value of ***start-symbol*** is used (thereby constraining the VARIANT attributes of valid structures to be unifiable with *single-variant*), and a sign with features of EP is present in the input, the resulting feature structure will have VARIANT features of type *ep-variant*, not *european-portuguese*, although *european-portuguese* is the most general unifier of *ep-variant* and *single-variant*.

As mentioned in the beginning of this section, it is possible to use the grammar to detect to which variety input text belongs to. This is done by parsing that text using the default start symbol and reading the value of attribute VARIANT from the resulting feature structure: the value *variant* indicates that no variant-specific marked signs were detected and the text could be from both; the values *ep-variant* and *bp-variant*, in turn, result from parsing text with features specific to EP and BP respectively.

5.5.1 Example

With this setup (and using the default variation policy) and the following lexical entries:

- Specific to EP
  - *dezasseis* (*sixteen*)
  - *bebé* (*baby*)
  - *canadiano* (*Canadian*)
- Specific to BP
  - *dezesseis* (*sixteen*)
  - *bebê* (*baby*)
  - *canadense* (*Canadian*)

the following sentences are accepted:

(5) Ontem nasceram dezasseis bebés canadianos.
yesterday were born sixteen.EP babies.EP Canadian.EP
Sixteen Canadian babies were born yesterday.

(6) Ontem nasceram dezesseis bebês canadenses.
yesterday were born sixteen.BP babies.BP Canadian.BP
Sixteen Canadian babies were born yesterday.

Furthermore, (5) is detected to be EP, and (6) to be BP. The sentences in (7) are rejected.

(7) a. Ontem nasceram dezasseis bebês canadenses.
yesterday were born sixteen.EP babies.BP Canadian.BP
Sixteen Canadian babies were born yesterday.

b. Ontem nasceram dezesseis bebés canadianos.
yesterday were born sixteen.BP babies.EP Canadian.EP
Sixteen Canadian babies were born yesterday.

5.6 Pragmatics

LXGram includes some information about pragmatics. Currently, the information encoded there is the information that used to be placed under message relations, identifying:

- topicalized constituents,
- subjects of passives,
- post-verbal subjects.

Pragmatic information is represented under the path SYNSEM|LOCAL|CTXT. The kind of pragmatic information that LXGram includes is encoded in a subfeature of CTXT called BACKGROUND.

Pragmatics is built the same way as semantics, but using these different features. We use the attributes RELS and HCONS under the path SYNSEM|LOCAL|CTXT|BACKGROUND. The information about topicalized constituents, subjects of passives and post-verbal subjects is placed under a relation in SYNSEM|LOCAL|CTXT|BACKGROUND|RELS. This relation is called *theme-rheme_d_rel*, and it has one argument for each of these pieces of information:

$$\begin{bmatrix} \text{LBL} & \textit{handle} \\ \text{PRED} & \textit{theme-rheme\_d\_rel} \\ \text{ARG0} & \textit{event} \\ \text{TOPICALIZED} & \textit{non-vacuous} \\ \text{POSTPOSED} & \textit{handle-or-ref-index} \\ \text{PASSIVIZED} & \textit{handle-or-ref-index} \end{bmatrix}$$

In LXGram, the type *non-vacuous* is the supertype of handles (type *handle*), events (*event*) and referential indices (*ref-ind*). This is the type of the feature TOPICALIZED (for topicalized constituents). The attributes POSTPOSED (for post-verbal subjects) and PASSIVIZED (for subjects of passives) are of the type *handle-or-ref-index*, a supertype of *handle* and *ref-ind*. Each of these features can be left uninstantiated, or filled with an appropriate value.

For instance, in sentences with post-verbal NP subjects, the referential index associated to that NP is also the value of the feature POSTPOSED. In this case, this value is filled in by the Subject-Head construction that projects subjects to the right of the verb. For sentences with pre-verbal or null subjects, this feature is left underspecified. The use of the other features is similar.

Note that the current version of LXGram does not implement passives: the value of the feature PASSIVE is therefore always underspecified.

The only kind of long distance dependencies that is implemented is relative clauses, so, following the LinGO Grammar Matrix, the value of the attribute TOPICALIZED is specified only when relative clauses are processed.

This *theme-rheme_d_rel* relation is introduced in the lexical entry for verbs, and there will be one such relation per verb form (at the moment, the exception is with auxiliary verbs, that do not introduce one of these relations, as the past participle that they select for already introduces one). The LBL feature of the *theme-rheme_d_rel* relation is identified with the LBL attribute of the corresponding verb's relation. The event that is the value of the ARG0 attribute of this relation is also identified with the event of the verb.

The composition of pragmatics is parallel to the composition of semantics: the RELS and HCONS of a rule (under BACKGROUND) is the difference-list append of the same features in the daughters.

An important note relates to the filling in of these attributes TOPICALIZED, POSTPOSED and PASSIVIZED. As mentioned above, this *theme-rheme_d_rel* relation is introduced in the lexical entries, but some of these three features are instantiated at a higher syntactic position (in the rule for post-verbal subjects, in the case of the feature POSTPOSED). Therefore, this relation needs to be visible in the feature structures in some place different than the attribute RELS (because in general it is not possible to know the exact position in this list where the relevant *theme-rheme_d_rel* can be found, as there will be one *theme-rheme_d_rel* relation per verb). We put this relation under the path SYNSEM|LOCAL|CTXT|BACKGROUND|KEYS|THEME-RHEME-REL. This feature is percolated from the head daughter to the mother node in all headed syntactic constructions (and it is also percolated from the daughter in all morphological rules): therefore the *theme-rheme_d_rel* relation associated to the verb of the current clause can always be found as the value of this feature in the feature structures for verb-headed constituents.

We present an example for the sentence "chegou o presidente" (*the president has arrived*), with a post-verbal subject, in Figure 5.16.

This information on pragmatics can also be incorporated in the MRSs produced by LXGram.

In LXGram, the root nodes of syntactic trees are produced by a unary syntax rule that takes as daughter a full sentence (the result of this is not shown in the previous example). The main motivation for this rule is related to the composition of semantics: namely it is to introduce in

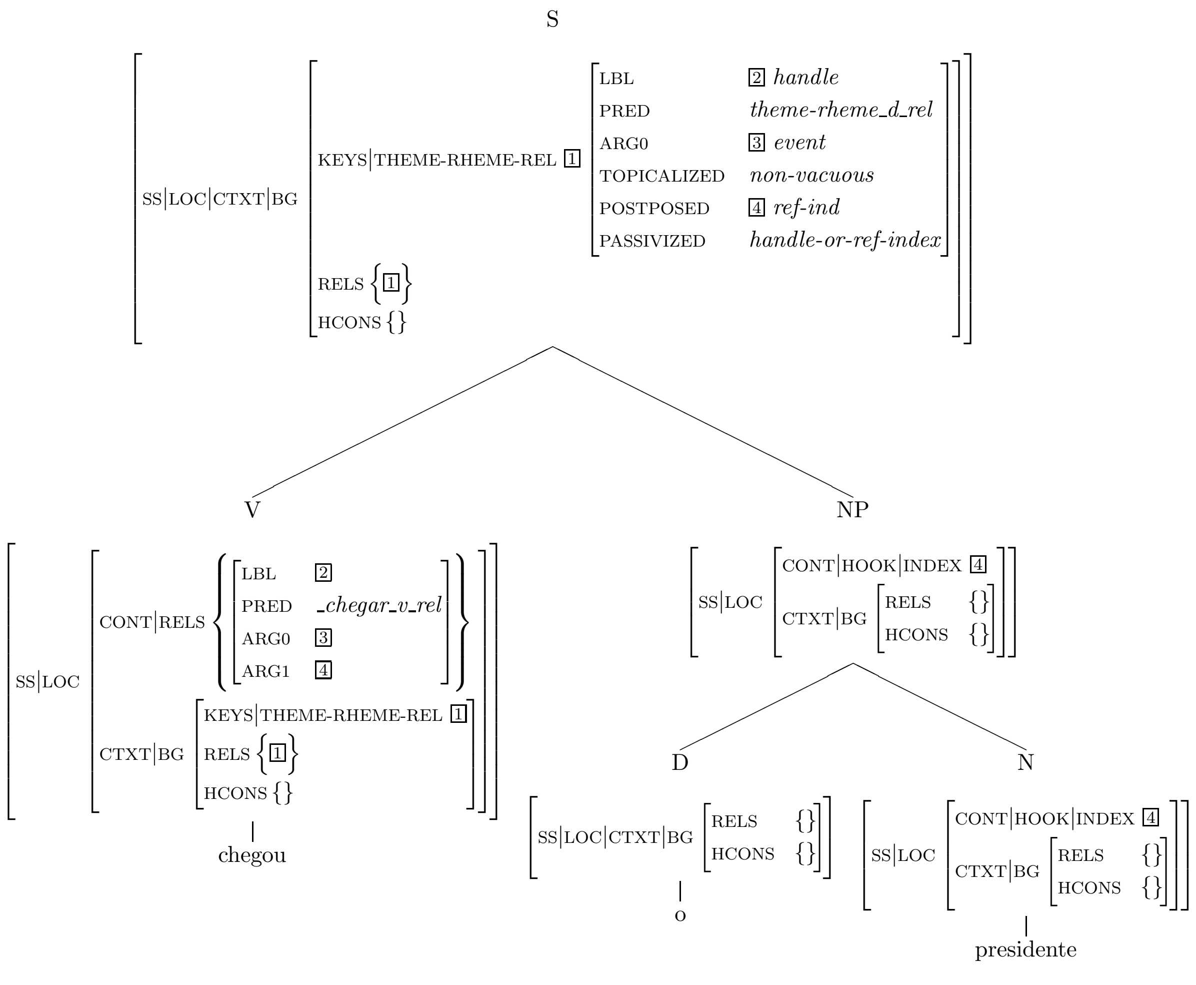


Figure 5.16: Example of the composition of pragmatics. The sentence is "chegou o presidente" (*the president has arrived*). SS abbreviates SYNSEM, LOC abbreviates LOCAL, and BG abbreviates BACKGROUND.

$$\begin{bmatrix} \textit{mrs} & \\ \text{LTOP} & \boxed{h1}\ h \\ \text{INDEX} & \boxed{e2}\ e \\ \text{RELS} & \left\langle \begin{bmatrix} \textit{theme-rheme\_d\_rel} & \\ \text{LBL} & \boxed{h3}\ h \\ \text{ARG0} & \boxed{e2} \\ \text{PASSIVIZED} & \boxed{u6}\ u \\ \text{POSTPOSED} & \boxed{x5}\ x \\ \text{TOPICALIZED} & \boxed{u4}\ u \end{bmatrix}, \begin{bmatrix} \textit{tam\_rel} & \\ \text{LBL} & \boxed{h3} \\ \text{ARG0} & \boxed{e2} \end{bmatrix}, \begin{bmatrix} \textit{\_chegar\_v\_rel} & \\ \text{LBL} & \boxed{h3} \\ \text{ARG0} & \boxed{e2} \\ \text{ARG1} & \boxed{x5} \end{bmatrix}, \begin{bmatrix} \textit{\_o\_q\_rel} & \\ \text{LBL} & \boxed{h7}\ h \\ \text{ARG0} & \boxed{x5} \\ \text{RSTR} & \boxed{h9}\ h \\ \text{BODY} & \boxed{h8}\ h \end{bmatrix}, \begin{bmatrix} \textit{\_presidente\_n\_-de-\_rel} & \\ \text{LBL} & \boxed{h10}\ h \\ \text{ARG0} & \boxed{x5} \\ \text{ARG1} & \boxed{r11}\ r \end{bmatrix} \right\rangle \\ \text{HCONS} & \left\langle \begin{bmatrix} \textit{qeq} & \\ \text{HARG} & \boxed{h1} \\ \text{LARG} & \boxed{h3} \end{bmatrix}, \begin{bmatrix} \textit{qeq} & \\ \text{HARG} & \boxed{h9} \\ \text{LARG} & \boxed{h10} \end{bmatrix} \right\rangle \end{bmatrix}$$

Figure 5.17: MRS for the sentence "chegou o presidente" (*the president has arrived*). Information about pragmatics is also included in the MRS.

the MRS representation a *qeq* relation between the global top and the handle of the relation corresponding to the main verb.

This syntax rule is implemented with the type *unary-root-clause*, defined in the file `syntax.tdl`. This type *unary-root-clause* is defined as inheriting from *basic-unary-root-clause-no_ctxt_in_mrs*. The constraints on the type *basic-unary-root-clause-no_ctxt_in_mrs* keep the semantic and the pragmatic representations separate. However, there is another type, *basic-unary-root-clause-ctxt_in_mrs*, defined in the same file, that can be used as the supertype of *unary-root-clause* instead. If the supertype of *unary-root-clause* is changed to *basic-unary-root-clause-ctxt_in_mrs*, the information under SYNSEM|LOCAL|CTXT|BACKGROUND will be appended to the MRS representation in SYNSEM|LOCAL|CONT.

In Figure 5.17 we show the MRS that is obtained by including pragmatic information in the semantics. The sentence is the same as that of the previous example: "chegou o presidente" (*the president has arrived*).

This separation of pragmatics and semantics allows the grammar to produce MRS with information that only affects truth conditions. However, if desired, it is also possible to include other sorts of information (namely pragmatic information) that can be useful for applications.

There is however an issue with this implementation concerning generation. When the grammar includes pragmatics in the MRSs, the *theme-rheme_d_rel* relations are introduced in the semantic representations at the topmost syntactic rule based on what is in SYNSEM|LOCAL|CTXT|BACKGROUND. These relations are not in the C-CONT of any rule or in the SYNSEM|LOCAL|CONT of any lexical item. This means that the generator cannot index these relations, and it is not possible to generate from MRSs containing *theme-rheme_d_rel* relations.

5.7 Constructional Content and Constructional Context

As presented in the previous section, the building up of pragmatic information is very much like the composition of semantics.

For the composition of semantics, a feature C-CONT is usually used in order to represent the contribution of lexical and syntactic rules to the semantics. For instance, in syntax rules, the RELS of the mother is the union of the RELS of all daughters and the RELS under the feature C-CONT.

For pragmatics, we could also think of a feature C-CTXT. At the moment, no rule in LXGram contributes to the pragmatic representation, but we would like to keep this possibility open.

However, instead of using a feature C-CONT and another feature C-CTXT, in LXGram we use two features CONT and CTXT grouped under a feature C. The semantic content of a rule is thus represented under C|CONT (instead of C-CONT), and its pragmatic information under C|CTXT.

5.8 Idioms

LXGram implements an idiom detection mechanism similar to what can be found at `http://wiki.delph-in.net/moin/JacyIdiom`. Our implementation is similar to the one described there, and the basic machinery concerning the configuration files that is described there is also used in LXGram. We explain some of the details briefly in the following paragraphs.

A boolean attribute IDIOM is used. It is present in all feature structures for words and phrases. The value of this feature is unified for all signs comprising a syntax tree. This is accomplished by having all lexical rules identify the value of their attribute IDIOM with that of the same attribute in its daughter and all syntax rules unify their IDIOM feature with that of every daughter.

Because both the features IDIOM and VARIANT (see Section 5.5) are unified across all signs of a syntax tree, we place them under a feature GLOBAL and unify this feature GLOBAL instead, taking advantage of the fact that unification is recursive. We also exploit minimal types (see Section 5.1) in order to avoid the presence of any of these features (IDIOM and VARIANT) if they are left unconstrained.

More specifically, the feature GLOBAL is appropriate for signs and of the type *global-min*:

$$\begin{bmatrix} \textit{sign} & \\ \textsc{global} & \textit{global-min} \end{bmatrix}$$

It can take as values the types in the following hierarchy:

$$\begin{tikzcd}
 & \textit{global-min} & \\
\textit{global-idiom} \arrow[ur] & & \textit{global-variant} \arrow[ul] \\
 & \textit{global} \arrow[ul] \arrow[ur] &
\end{tikzcd}$$

The type *global-idiom* is the type where the attribute IDIOM is declared. It has the following definition:

$$\begin{bmatrix} \textit{global-idiom} & \\ \textsc{idiom} & \textit{bool} \end{bmatrix}$$

In a similar fashion, the type *global-variant* is the most general type for which the feature VARIANT is appropriate:

$$\begin{bmatrix} \textit{global-variant} & \\ \textsc{variant} & \textit{variant} \end{bmatrix}$$

As mentioned before, this feature must be unified across all signs in a parse tree. Therefore, the type *lex-rule* (the supertype of all lexical rules) includes this constraint:

$$\begin{bmatrix} \textit{lex-rule} \\ \textsc{global}\ \boxed{1} \\ \textsc{dtr}|\textsc{global}\ \boxed{1} \end{bmatrix}$$

Also, the types *basic-unary-phrase* (the supertype of all unary constructions) and *basic-binary-phrase* (the supertype of all binary phrases) also include the constraints necessary to propagate these values:

$$\begin{bmatrix} \textit{basic-unary-phrase} \\ \textsc{global}\ \boxed{1} \\ \textsc{args}\ \left\langle \begin{bmatrix} \textsc{global}\ \boxed{1} \end{bmatrix} \right\rangle \end{bmatrix} \begin{bmatrix} \textit{basic-binary-phrase} \\ \textsc{global}\ \boxed{1} \\ \textsc{args}\ \left\langle \begin{bmatrix} \textsc{global}\ \boxed{1} \end{bmatrix}, \begin{bmatrix} \textsc{global}\ \boxed{1} \end{bmatrix} \right\rangle \end{bmatrix}$$

Consider the example idiom in the following sentence:

(8)	Ele	pregou-lhes	um	susto.	
	he	nailed them	a	scare	
	He scared them.				

In order to accommodate such an expression, in LXGram we create a lexical entry for the verb “pregar” according to which it is a ditransitive verb. This lexical entry is also marked with the feature GLOBAL|IDIOM taking the value $+$.

The machinery described in `http://wiki.delph-in.net/moin/JacyIdiom` makes the LKB parser check for an idiom rule whenever a root node is produced with the value $+$ for the IDIOM feature (as in this example). The parse is accepted only if there is a rule that can match the MRS for the sentence. For this expression, we also include the following rule in the file `idioms.mtr`:

$$\begin{bmatrix} \textsc{input}|\textsc{rels}\ \left\{ \begin{bmatrix} \textsc{pred} & \text{“\_pregar\_v\_i\_rel”} \\ \textsc{arg2} & \boxed{1} \end{bmatrix}, \begin{bmatrix} \textsc{pred} & \text{“\_susto\_n\_rel”} \\ \textsc{arg0} & \boxed{1} \end{bmatrix} \right\} \end{bmatrix}$$

This rule matches the MRS for this sentence, and the sentence is accepted and marked as containing an idiomatic expression.

A similar sentence with a different noun will not be parsed, because it will display an IDIOM attribute with the value $+$, but no idiom rule will be found that can match the MRS for that sentence.

Several other expressions can be parsed with this rule and this special entry for the verb “pregar”: “pregar dois sustos” (*scare twice*, lit. *inflict two scares*), “pregar um grande susto” (*scare for real*, lit. *inflict a big scare*), etc.

5.9 Punctuation

LXGram contains a very incipient implementation of punctuation. At the moment only periods (`.`), question marks (`?`), commas (`,`) and ellipses (`...`) are recognized, all other punctuation characters being ignored. Except for commans, they are allowed at the end of sentences only. Also, at the moment, commas are allowed only in a few places (for instance, after sentence initial interjections — see Section 7.7).

As in the ERG, punctuation is implemented as affixation. This strategy is possibly more involved than treating punctuation as separate tokens, but it produces more familiar parse trees (modulo the extra nodes for the spell-changing rules that attach punctuation characters to base forms), since punctuation does not give rise to independent branches.

The implementation is as follows. A feature PUNCT is used in all signs. There are two subfeatures of PUNCT: R-PUNCT and L-PUNCT. The attribute R-PUNCT contains information about the punctuation mark appearing right after a word. In all lexical entries it has the value *no-punctuation-mark*, denoting the fact that words come in the lexicon with no punctuation marks attached to them. The attribute L-PUNCT of a word contains information about the punctuation mark following the word that precedes that word in the sentence. It is underspecified in all lexical entries and filled in via the syntax rules. There is one morphological rule for each punctuation mark. These rules constrain the R-PUNCT feature of the mother node with one of several values: *period* (for the rule inserting a period), *question-mark*, etc. They also constrain the daughter to have the value *no-punctuation-mark* for its feature R-PUNCT, which prevents these rules from applying to their own output.

The syntax rules percolate the information in PUNCT. The R-PUNCT of the mother node of a syntax rule is always the R-PUNCT of its rightmost daughter. The L-PUNCT of the mother node of a syntax rule is always the L-PUNCT of the leftmost daughter, also. The L-PUNCT of the rightmost daughter of binary phrases is the R-PUNCT of the leftmost daughter.

We make sentence final punctuation have an impact on the semantics. In particular we want to distinguish among propositions, imperatives and questions. Note that with the setup described in the previous paragraph, the R-PUNCT of the root node is the R-PUNCT of the final word of the sentence. If that word has a period attached to it, the value of this feature will be *period*. We use a subfeature S-FORCE under these types for punctuation, where the associated sentence force is represented. In the case of the type *period*, the value of its feature S-FORCE is *proposition-or-command* (this type is made more specific with information coming from the verb form). This feature is unified in root nodes with the feature SF that appears in MRSs under the event for the main verb (see Section 5.4).

Most binary phrases also constrain the R-PUNCT of the leftmost daughter to have the value *no-punctuation-mark*, thus blocking most of the sentence internal punctuation.

LXGram also accepts sentences with no punctuation mark on the last word. In this case, the sentence is considered as if it ended with a period.

5.10 Agreement

In LXGram agreement is controlled via the unification of the features AGR, where the information relevant for agreement is encoded (person, number and gender). In LXGram the feature AGR is put under SYNSEM|LOCAL|CAT|HEAD.

Agreement is enforced by unifying the AGR features of the relevant elements. For instance, determiners select the noun via their SYNSEM|LOCAL|CAT|MARKER|SELECT feature. The masculine singular form of the definite article, "o", therefore constrains the AGR feature of the noun it selects in the expected way:

$$\left[\text{SYNSEM|LOCAL|CAT|HEAD|MARKER|SELECT|LOCAL|CAT|HEAD|AGR|PNG}\ \begin{bmatrix}\text{NUMBER} & \textit{singular}\\ \text{GENDER} & \textit{masculine}\end{bmatrix}\right]$$

However, constraints like these are to be produced by inflectional rules, namely the ones responsible for gender and number inflection. Note that for nouns, these rules constrain the values of the features NUMBER and GENDER under the path SYNSEM|LOCAL|CAT|HEAD|AGR|PNG. In LXGram we use the same set of rules to produce gender and number variants of nouns, determiners, adjectives and other elements. To this end, we make nouns specifiers and noun modifiers also contain an AGR feature under their head, which is unified in the lexical entries for these items with the AGR feature of the noun they select. So for instance, determiners have constraints like the following:

$$\left[\textsc{synsem}|\textsc{local}|\textsc{cat}|\textsc{head}\ \begin{bmatrix}\textsc{agr}\ \boxed{1}\\ \textsc{select}|\textsc{local}|\textsc{cat}|\textsc{head}|\textsc{agr}\ \boxed{1}\end{bmatrix}\right]$$

The inflectional rules simply constrain the person and number features under SYNSEM|LOCAL|CAT|HEAD|AGR. Because this AGR feature of determiners, adjectives, etc. is unified with the AGR feature of the noun they select, the appropriate constraints are put in a place where they will be unified with the AGR of nouns when syntax rules are applied. This way, the same set of morphological rules can apply to a wide range of syntactic categories: nouns, adjectives, determiners, etc.

5.11 Additional Features for the Composition of Semantics

In LXGram we use an additional feature SARG under the path SYNSEM|LOCAL|CONT|HOOK. This feature is either unified with the attribute LTOP or the attribute INDEX under the same path.

The purpose of this feature is to allow for a single lexical entry for verbs that can take an NP or CP complement. Consider the examples with the verb "dizer" below.

(9) a. Ele disse [$_{NP}$ isso.]
he said that
He said that.

b. Ele disse [$_{CP}$ que ia chover.]
he said that went rain
He said that it was going to rain.

For the first example, the second argument of the verbal relation is a referential index, found under the INDEX feature of the NP complement. For the second example, this argument is a handle, found in the LTOP attribute of the CP complement.

In LXGram we include a constraint in the lexical type of verbs like this one according to which the second argument of the verb's relation is the SARG feature of its complement. In the lexical entries of nouns, the SARG feature is unified with the attribute INDEX. These features are percolated from the head daughter in syntax rules. In the lexical entries of complementizers, the SARG feature is identified with the feature LTOP.

5.12 Modification

All modification structures are handled via the Head-Functor rules (see Section 5.3).

In LXGram we use a feature under SYNSEM|LOCAL|CAT called MODIFICATION. There are several features under MODIFICATION, each with a particular purpose.

One attribute under MODIFICATION is MODIFIABLE. This feature is used to control spurious attachment ambiguity. Consider the following examples:

(10) a. Chegou ontem um carro.
arrived yesterday a car
A car arrived yesterday.

b. Chegou um carro ontem.
arrived a car yesterday
A car arrived yesterday.

c. Chegou ontem.
arrived yesterday
It arrived yesterday.

In order to parse the first two sentences, in LXGram we allow adverbs to modify verb phrases (as in the first example) and also sentences (second example). Null subjects, like in the last example, are processed with a unary syntactic rule that discharges the SUBJ feature of the daughter and adds pronominal semantics. This creates two possible attachments for the last sentence: the adverb can attach to the verb phrase and the resulting node feed the unary rule for null subjects, or it can attach to the node produced by the unary syntax rule for null subjects. The attribute MODIFIABLE is used to block modification of the mother node of null subject phrases, in order to avoid this potential spurious ambiguity.

Another attribute under MODIFICATION, MODIFIERS, contains information about whether the current syntactic node includes modifiers. This information is used to control the co-occurrence of definite articles and proper names: some proper names cannot be preceded by definite articles unless they are modified (see Section 8.5.2).

5.13 Miscelaneous

This section is meant to describe some implementation decisions that are occasionally referred to in the rest of this document.

5.13.1 Copular verbs

Copular verbs are analysed as raising the subject of their complement. Every element that can be a predicate selected by a copula must thus have a non empty SUBJ list. This allows these elements to constrain the type of subject they can be a predicate of.

An example follows. Sentences (11) indicate that the CP that co-occurs with an adjective of the kind of "óbvio" (*obvious*) must be its subject, as it cannot co-occur with another subject. A color adjective, however, cannot have a sentential subject (12). Adjectives can thus control the type of subjects they take in their SUBJ list.

(11) a. Isso é óbvio.
that is obvious
That is obvious.

b. É óbvio que isso é assim.
is obvious that that is so
It is obvious that that is like that.

c. * Isso é óbvio que isso é assim.
that is obvious that that is so

(12) * É verde que isso é assim.
is green that that is so

Chapter 6

Preprocessing and Morphology

This chapter deals with issues related to LXGram input and treatment of morphology.

6.1 Input

LXGram uses the UTF-8 character encoding. Running it in an environment with a different character encoding will make the use of special characters problematic.

Implementation of morphology is only demonstrative. It is not exhaustive. It should, however, be sufficient to parse and generate many interesting sentences. The plan is to interface LXGram with external components that deal with morphology ([LXS, Web Page; LXL, Web Page; LXC, Web Page; Branco and Silva, 2002; Branco and Henriques, 2003; Branco and Silva, 2003a; Branco and Silva, 2003b; Branco and Silva, 2003c; Branco and Silva, 2003d; Branco and Silva, 2003e; Branco and Silva, 2004a; Branco and Silva, 2004b]), so this aspect will not be further developed.

The file `preprocessor.fsr` includes several preprocessing rules that allow the use of contracted forms in the input. Figure 6.1 shows an example. The formalism for these rules allows the use of regular expressions and references to portions of the matched elements. However, it does not allow the disambiguation of the items involved. For instance, "a" is ambiguous (preposition, article, clitic), but the contraction "à" is not (the expansion "a a" must be a preposition followed by an article), which cannot be expressed in these rules.

6.2 Verbal Morphology

Tense and aspect information conveyed by verbal inflection may affect truth conditions. Therefore, this information should be reflected in the semantic representations in the grammar. For instance, if the grammar is used for generation, one wants to control these aspects in the grammar input (semantic representations).

However, the initial strategy of coding this information with features and types that reflect morphology directly was adopted. That is, encodings of the semantic values expressed in each occurrence by tense, aspect and modality are not provided in the MRSs produced by the grammar. Instead, the names given to the inflected forms of verbs are used in the semantic representations. This has the advantages of simplifying the MRSs and of postponing a more elaborate and time-consuming analysis due in implementation phase E.20. This way, the full inflected forms of verbs

```
-ness([ea]s?)([\.\?,]?)    em ess\1\2
```

Figure 6.1: Preprocessor rule to expand the contractions "nesse", "nesses", "nessa", "nessas" into "em esse" (*on that.MASC-SING*), "em esses" (*on those.MASC-PLU*), "em essa" (*on that.FEM-SING*), "em essas" (*on those.FEM-PLU*) respectively.

$$\begin{bmatrix} \textit{mrs} & \\ \text{LTOP} & \boxed{h1}\ h \\ \text{INDEX} & \boxed{e2}\begin{bmatrix} \textit{e} & \\ \text{SF} & \textit{proposition} \\ \text{ELLIPTICAL-PUNCT} & - \\ \text{E.TENSE} & \textit{pretérito-perfeito} \\ \text{E.ASPECT.PERF} & - \\ \text{E.MOOD} & \textit{indicativo} \end{bmatrix} \\ \text{RELS} & \left\langle \begin{bmatrix} \textit{tam\_rel} & \\ \text{LBL} & \boxed{h3}\ h \\ \text{ARG0} & \boxed{e2} \end{bmatrix}, \begin{bmatrix} \textit{\_chover\_v\_rel} & \\ \text{LBL} & \boxed{h3} \\ \text{ARG0} & \boxed{e2} \end{bmatrix} \right\rangle \\ \text{HCONS} & \left\langle \begin{bmatrix} \textit{qeq} & \\ \text{HARG} & \boxed{h1} \\ \text{LARG} & \boxed{h3} \end{bmatrix} \right\rangle \end{bmatrix}$$

Figure 6.2: MRS for the sentence "Choveu" (*It rained*). Tense, aspect and mood information is encoded using the usual designations appearing in conjugation tables.

are available from an early stage without sacrificing precision when generating.

An example of an MRS where such encoding is visible is given in Figure 6.2 (the feature PERF is explained in Section 7.1). The type hierarchy of mood values is also presented (Figure 6.3).

6.3 Nominal Morphology

LXGram contains a set of inflectional rules that produce gender and number variants for nouns, adjectives, determiners and pronouns. The same rules apply to items of all of these grammatical categories. They are listed in Chapter 10. There are two sets of rules: one set for gender variants (one rule for masculine forms and one rule for feminine forms) and another set of rules for number variants (one rule for singular forms and one rule for plural forms).

For items that have gender and number variants, the gender rules apply directly to lexical items and the number rules apply to the output of the gender rules. This is the case of the adjective "alto" (*tall*): "alto" masculine singular, "alta" feminine singular, "altos" masculine plural, "altas" feminine plural.

For items that do not show any gender variants but have number morphology, the number rules apply directly to the lexical items. The adjective "grande" (*large*) is an example: "grande" masculine or feminine singular, "grandes" masculine or feminine plural. The value of the GENDER feature will be underspecified.

For items that do not have gender or number variants, none of these rules apply. The adjective "simples" (*simple*) falls in this case. This form is underspecified for gender and number.

The application of the inflectional rules is controlled by the type hierarchy under *sign*. Words that need to undergo both sets of rules have a lexical type inheriting from the type *nominal-uninflected-elem*. The rules for gender inflection constrain their daughter to be of this same type, and inherit from the type *nominal-gend_infl-num_uninfl-elem*. Words that have no gender variants also inherit from *nominal-gend_infl-num_uninfl-elem*. The rules for number morphology constrain their daughter to be of this type, *nominal-gend_infl-num_uninfl-elem*, and inherit from the type *syntactic-sign*. Words with no gender or number distinctions also inherit from

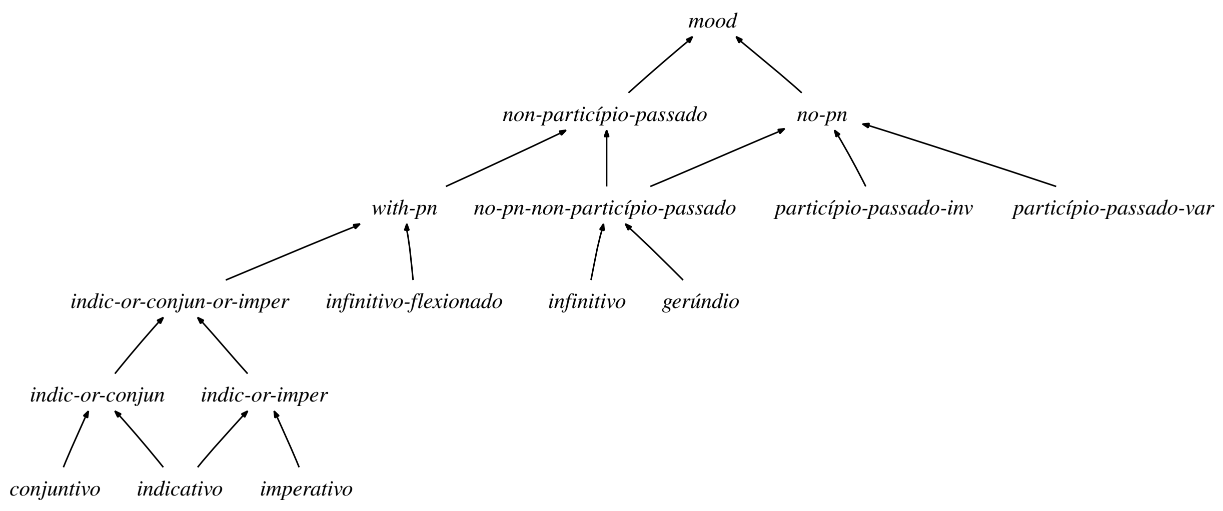


Figure 6.3: Type hierarchy under *mood*. The types *with-pn* and *no-pn* separate verb forms into those that show subject agreement and those that do not, but they never show up in semantic representations.

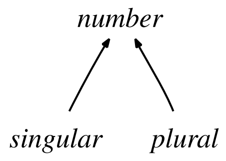


Figure 6.4: Type hierarchy under *number*.

syntactic-sign. Neither *nominal-uninflected-elem* or *nominal-gend_infl-num_uninfl-elem* inherit from *syntactic-sign*, and these three types are all incompatible. All daughters of syntactic rules must be *syntactic-sign*, because the elements of the feature ARGS are constrained to be of this type (see Section 7.2).

This way, only words that show no morphological variants or words that have undergone the morphology rules can feed the syntax rules.

Just like with verbal morphology, the information about morphological person, number and gender is visible in MRS representations. This information is encoded under the features PNG|-PERSON, PNG|NUMBER and PNG|GENDER of referential indices (type *ref-index*). The type hierarchy for the values of PERSON is presented in Section 8.3. It also encodes information about Tu-Vous distinctions. The type hierarchies of gender and number values are presented in Figure 6.4 and Figure 6.5 respectively.

The morphological rules for gender and number unify the PNG feature of the mother with the PNG feature of the daughter.

The hierarchy under *number* comes as expected in as much as Portuguese has a two-fold distinction for number.

The hierarchy under *gender* deserves an explanation. When it comes to adjectives, there is only a two-way distinction for gender values: masculine and feminine. In LXGram, a third value is included, *neuter*, so that the following co-occurrence patterns can be treated as agreement for gender:

$$\begin{tikzcd}
 & \textit{gender} & \\
\textit{masculine-or-neuter} \arrow[ur] & & \textit{masculine-or-feminine} \arrow[ul] \\
\textit{neuter} \arrow[u] & \textit{masculine} \arrow[ul] \arrow[ur] & \textit{feminine} \arrow[u]
\end{tikzcd}$$

Figure 6.5: Type hierarchy under *gender*.

(13) a.

Beberam	todo	aquele	vinho.
they drank	all	that	wine

They drank all that wine.

b.

Beberam	toda	aquela	cerveja.
they drank	all	that	beer

They drank all that beer.

c.

Beberam	tudo	aquilo.
they drank	all	that

They drank all that.

There are no other combination possibilities between the forms "todo", "toda", "tudo" and forms like "aquele". "aquela", "aquilo". "Todo" and "aquele" are analyzed as having the masculine gender, "toda" and "aquela" the feminine gender and "tudo" and "aquilo" the neuter gender.

Note that adjectives agreeing with these neuter elements appear in their masculine forms. Also, nouns can never co-occur with "aquilo" or "tudo".

For these two reasons, we use the type *masculine-or-neuter* in the lexical rules that produce masculine forms (recall that the same rules apply to all classes). We further stipulate that all nouns must have the gender value *masculine-or-feminine*.

As a result, masculine nouns will have the gender value *masculine* (the unifier of *masculine-or-feminine* and *masculine-or-neuter*). Therefore, they cannot co-occur with neuter forms.

On the other hand, the masculine forms of adjectives will have the gender value *masculine-or-neuter*. This way, they can be in an agreement relation with masculine nouns and these neuter elements.

The neuter forms are listed in the lexicon, with the GENDER feature constrained to be *neuter*, and do not undergo the gender inflection rules. Also, the neuter items do not have plural forms. The inflectional rule that produces plural nominal forms constrains the PNG|GENDER feature with the value *masculine-or-feminine*. The neuter elements are therefore not elligible to feed the rule for plurals. Furthermore, masculine forms of adjectives end up with the value *masculine* for the GENDER feature (the unifier for *masculine-or-neuter*, the type that this feature is constrained to be by the masculine inflectional rule, and *masculine-or-feminine*, the type for GENDER in the plural inflecional rule).

Syntactic agreement is handled by unifying the PNG attribute of the elements involved in an agreement relation.

Chapter 7

Auxiliaries and Basic Phrase Structure

This chapter presents information on the implementation of the auxiliaries of compound tenses and the basic phrase structure with verbs, adjectives, adverbs, prepositions and conjunctions as heads.

7.1 Auxiliaries

For the encoding of the information contributed by the auxiliaries of compound tenses, a boolean attribute PERF under ASPECT is used, as is done in the ERG. Simple tenses are marked PERF - and compound tenses PERF +.

The marking on the simple tenses is required, since otherwise each of them would be underspecified and would denote both the simple and the corresponding compound tense.

This choice is however best viewed as a compromise or a temporary solution. A first problem is that the attribute PERF (perfect) is arguably not semantically valid for Portuguese. The meaning of the compound forms involving these auxiliaries is not like in English. They carry rather an anteriority meaning, in general. For instance the, compound tense formed by the past imperfect form of the auxiliary is a pluperfect and, in contrast to English, sentences like (14) are not semantically anomalous in Portuguese.

(14)	Ele	tinha	soluçado.
	he	had	hiccuped

On the other hand, the meaning they contribute is dependent on the auxiliary's tense and mood. For example, the compound form with the auxiliary in the present indicative is not semantically equivalent to the simple perfective preterit form, but the simple pluperfect and the compound pluperfect are semantically equivalent to each other. The encoding used fails to represent these last two tenses in an equivalent way.

A second problem is that mixing the traditional names of simple tenses with the PERF attribute waters down our choice of representing tense and aspect via their usual names, and actually creates apparently contradictory information (not really contradictory in that the feature PERF denotes something different from the perfective-imperfective distinction that is present in the inflectional names), as is illustrated in Figure 7.1. Hence, abandoning the PERF attribute will be considered in the next phases of the implementation.

Most of the implementation is straightforward. The lexical types of the auxiliaries do not need to be much different from those of regular verbs. They have an empty RELS list, since they do not contribute relations to the semantics of the sentence, and they behave like subject raising

$$\left[\begin{array}{ll}
\textit{mrs} & \\
\text{LTOP} & \boxed{h1}\ h \\
\text{INDEX} & \boxed{e2}\left[\begin{array}{ll} e & \\ \text{SF} & \textit{proposition} \\ \text{E.TENSE} & \textit{pretérito-imperfeito} \\ \text{E.ASPECT.PERF} & + \\ \text{E.MOOD} & \textit{indicativo}\end{array}\right] \\
\text{RELS} & \left\langle \left[\begin{array}{ll} \textit{tam\_rel} & \\ \text{LBL} & \boxed{h3}\ h \\ \text{ARG0} & \boxed{e2}\end{array}\right], \left[\begin{array}{ll}\textit{\_chover\_v\_rel} & \\ \text{LBL} & \boxed{h3} \\ \text{ARG0} & \boxed{e2}\end{array}\right]\right\rangle \\
\text{HCONS} & \left\langle \left[\begin{array}{ll} \textit{qeq} & \\ \text{HARG} & \boxed{h1} \\ \text{LARG} & \boxed{h3}\end{array}\right]\right\rangle
\end{array}\right]$$

Figure 7.1: MRS representation of a pluperfect form. The Portuguese sentence is "Tinha chovido" (*It had rained*).

$$\left[\begin{array}{l}
\textit{aspect-tam-verb-rule} \\
\text{SYNSEM}|\text{LOCAL}|\text{CONT}|\text{HOOK}|\text{INDEX}|\text{E}|\text{ASPECT}|\text{PERF}\ \boxed{1} \\
\text{ALTS}|\text{SIMPLE-TENSE-PERF}\ \boxed{1}
\end{array}\right]$$

Figure 7.2: Type definition of a supertype of all verbal inflectional rules except the one to produce past participles.

verbs in that they have a single VP complement corresponding to the main verb and share their subject's index with that of their complement's. Since they are semantically empty, they copy their complement's HOOK.

But two problems have to be addressed. The lexical rules that produce the inflected forms of non-auxiliary verbs should constrain tense to be PERF - (except for the main past participle form which occurs in the compound forms), but the auxiliaries cannot be constrained in this way and also need to undergo inflection. The second problem is that these auxiliaries must be kept from iterating.

The latter requirement has been solved by following an indication in [Sag et al., 2003]: the auxiliaries of compound tenses are not allowed to form a past participle. For this solution, the feature ALTS from the LinGO Grammar Matrix is used. This feature is employed to control application of lexical rules. A new attribute under ALTS was created, PAST-PARTICIPLE, of type *luk* (defined in the LinGO Grammar Matrix) and constrained to be *na* for the auxiliaries. The past participle lexical rule constrains its daughter to be ALTS|PAST-PARTICIPLE +.

The first problem was solved taking the solution to the second issue as inspiration. Verbs specify whether their inflected forms (apart from the past participle) should be PERF + or -. For this a new feature SIMPLE-TENSE-PERF in ALTS is used. Auxiliaries are specified to be ALTS|SIMPLE-TENSE-PERF + and all others are ALTS|SIMPLE-TENSE-PERF -. The lexical rules just structure-share this value with the attribute used to encode aspect (Figure 7.2).

In order to avoid proliferation of features, the technique described in Section 5.1 was employed. The feature ALTS is thus declared in the LinGO Grammar Matrix to be of type *alts-min*, and the type hierarchy under *alts-min* is depicted in Figure 7.3 (where *alts-min*, *alts* and *no-alts*

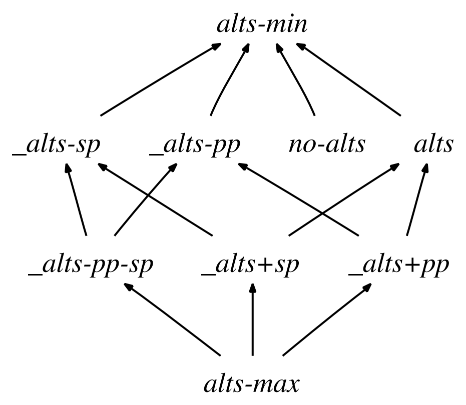


Figure 7.3: Type hierarchy under *alts-min*.

come from the LinGO Grammar Matrix, and types *alts*, *_alts-sp* and *_alts-pp* introduce features PASSIVE, SIMPLE-TENSE-PERF and PAST-PARTICIPLE respectively).

Finally, the tense of the auxiliary verbs is constrained to be *non-pretérito-perfeito* (a supertype of all tense types except *pretérito-perfeito*), so that sentences like the one in (15) are ruled out.

(15)	* Teve	chovido.
	it had.PRETÉRITO-PERFEITO	rained

7.2 Constraints on Syntactic Elements

LXGram uses two incompatible subtypes of *sign*, *syntactic-sign* and *morphological-sign* to control the application of lexical rules. Any lexical item that must undergo inflection inherits from the type *morphological-sign* and is thus blocked from feeding the syntactic rules directly. All lexical items that are not inflected and all lexical rules that produce fully inflected items inherit from *syntactic-sign*. These two types are employed in a way similar to the boolean feature INFLECTED that comes in the LinGO Grammar Matrix and the types *uninflected-lexeme* and *fully-inflected-lexeme* where this feature is constrained, but this feature and these two types are not used in LXGram.

The type *synsem-min* is the most general value that the attribute SYNSEM can take, and *optional-synsem-min* corresponds to the LinGO Grammar Matrix type *unexpressed*. The type *optional-synsem-min* is incompatible with the synsem type *canonical-synsem-min* (the equivalent of the LinGO Grammar Matrix type *canonical*), which is the type of SYNSEM that all realized elements are constrained to have (the Principle of Canonicality of [Ginzburg and Sag, 2000]).

The Principle of Canonicality is enforced in LXGram via a constraint on the type of the feature ARGS. The LKB and PET use this feature to encode the daughters of a syntactic or lexical rule. It is list valued, and the order of the elements in this list corresponds to surface word order.

For the implementation of lists, the types *list*, *cons* and *null* come from the LinGO Grammar Matrix. The type *null* represents an empty list, and the type *cons* represents a non-empty list, with the features FIRST (where the first element of that list is) and REST (with the tail of that list, which is also a *list*). The type hierarchy and constraints for lists and lists of optional synsems are similar to the definitions in the LinGO Grammar Matrix for *olist*, but we also use the type *list-of-synsems* for lists whose elements are synsems (the type of the features SUBJ and COMPS in general). The relevant part of the type hierarchy is in Figure 7.4.

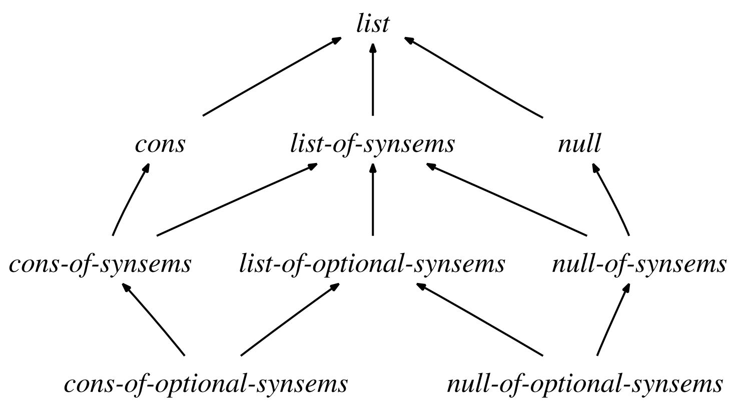


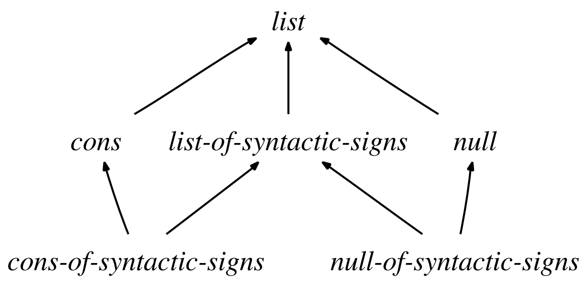
Figure 7.4: Types of lists of synsems

The types for non-empty lists of synsems and non-empty lists of optional synsems further constrain these two features in the expected way:

$$\begin{bmatrix} \textit{cons-of-synsems} & \\ \textsc{first} & \textit{synsem-min} \\ \textsc{rest} & \textit{list-of-synsems} \end{bmatrix}$$

$$\begin{bmatrix} \textit{cons-of-optional-synsems} & \\ \textsc{first} & \textit{optional-synsem-min} \\ \textsc{rest} & \textit{list-of-optional-synsems} \end{bmatrix}$$

In the type *phrase* (the supertype of all syntactic constructions), ARGS is constrained to be of the type *list-of-syntactic-signs*. The relevant part of the type hierarchy is:



The type *cons-of-syntactic-signs* constrains the features FIRST (the head of the list) and REST (its tail), both inherited from *cons*:

$$\begin{bmatrix} \textit{cons-of-syntactic-signs} & \\ \textsc{first} & \textit{syntactic-sign} \\ \textsc{rest} & \textit{list-of-syntactic-signs} \end{bmatrix}$$

The type *syntactic-sign* is constrained to have a synsem of type *canonical-synsem-min*.

As mentioned before, the feature ARGS of the type *phrase* is constrained to be of the type *list-of-syntactic-signs*. In the subtypes where its size is constrained (it will never be empty), the type for ARGS will be inferred to be *cons-of-syntactic-signs*, as this type is the most general unifier of *list-of-signs-syntactic-signs*, a constraint inherited from *phrase*, and *cons*, the most general type for which FIRST and REST are appropriate.

For instance, the constraints defined in the supertype of phrases with two daughters can be very simple:

$$\begin{bmatrix} \textit{basic-binary-phrase} & \\ \textsc{args} & \left\langle \textit{*top*},\ \textit{*top*} \right\rangle \end{bmatrix}$$

Since *basic-binary-phrase* inherits from *phrase*, where ARGS is declared to be of the type *list-of-syntactic-signs*, the full constraints on *basic-binary-phrase* will be as desired (the definition just presented is notationally equivalent to the left operand):

$$\begin{bmatrix} \textit{basic-binary-phrase} & \\ \textsc{args} & \begin{bmatrix} \textit{cons} & \\ \textsc{first} & \textit{*top*} \\ \textsc{rest} & \begin{bmatrix} \textit{cons} & \\ \textsc{first} & \textit{*top*} \\ \textsc{rest} & \textit{null} \end{bmatrix} \end{bmatrix} \end{bmatrix} \sqcap \begin{bmatrix} \textit{phrase} & \\ \textsc{args} & \textit{list-of-syntactic-signs} \end{bmatrix} =$$

$$\begin{bmatrix} \textit{basic-binary-phrase} & \\ \textsc{args} & \begin{bmatrix} \textit{cons-of-syntactic-signs} & \\ \textsc{first} & \begin{bmatrix} \textit{sign} & \\ \textsc{synsem} & \textit{canonical-synsem-min} \end{bmatrix} \\ \textsc{rest} & \begin{bmatrix} \textit{cons-of-syntactic-signs} & \\ \textsc{first} & \begin{bmatrix} \textit{sign} & \\ \textsc{synsem} & \textit{canonical-synsem-min} \end{bmatrix} \\ \textsc{rest} & \textit{null-of-syntactic-signs} \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

The Principle of Canonicality is then simply a constraint on this feature ARGS in a very general type (*phrase*).

7.3 Verb Phrases and Basic Sentence Structure

LXGram provides basic support to VP and sentence structure. VPs are produced from verb items with head-complement schemata that inherit from the type *basic-head-comp-phrase* from the LinGO Grammar Matrix.

Subjects of finite clauses are produced with two schemata that both inherit from *basic-head-subj-phrase* also in the LinGO Grammar Matrix. One is head final, another is head initial.

There are also two rules for null subjects: one for expletives, another for semantically non-empty subjects. In the last case the semantics of a personal pronoun is added to the sentence MRS representation, via the C-CONT attribute of the rule. Figure 7.5 shows an example.

There is also a unary syntactic rule applying to full sentences, which introduces a *qeq* relation between the global top and the handle labeling the relation for the main verb. It produces root nodes.

Subject-verb agreement is controlled in the lexical rules that produce inflected forms of verbs, since these can have access to the subject via the SUBJ valence list.

$$\begin{bmatrix} \textit{mrs} & \\ \text{LTOP} & \boxed{h1}\ h \\ \text{INDEX} & \boxed{e2}\ e \\ \text{RELS} & \left\langle \begin{bmatrix}\textit{pronoun\_q\_rel} & \\ \text{LBL} & \boxed{h3}\ h \\ \text{ARG0} & \boxed{x4}\ x\\ \text{RSTR} & \boxed{h5}\ h \\ \text{BODY} & \boxed{h6}\ h\end{bmatrix}, \begin{bmatrix}\textit{pronoun\_n\_rel} & \\ \text{LBL} & \boxed{h7}\ h\\ \text{ARG0} & \boxed{x4}\end{bmatrix}, \begin{bmatrix}\textit{tam\_rel} & \\ \text{LBL} & \boxed{h8}\ h \\ \text{ARG0} & \boxed{e2}\end{bmatrix}, \begin{bmatrix}\textit{\_chegar\_v\_rel} & \\ \text{LBL} & \boxed{h8}\\ \text{ARG0} & \boxed{e2} \\ \text{ARG1} & \boxed{x4}\end{bmatrix}\right\rangle \\ \text{HCONS} & \left\langle \begin{bmatrix}\textit{qeq} & \\ \text{HARG} & \boxed{h1}\\ \text{LARG} & \boxed{h8}\end{bmatrix}, \begin{bmatrix}\textit{qeq} & \\ \text{HARG} & \boxed{h5} \\ \text{LARG} & \boxed{h7}\end{bmatrix}\right\rangle \end{bmatrix}$$

Figure 7.5: MRS for a sentence with a null subject. The sentence is "Chegou" (*He/she/it arrived*).

7.3.1 Postponed Coverage or Known Limitations

Subcategorization patterns of verbs are left to phase C. Clitics will be properly implemented in phases C.13 and E.19.

7.4 Degree Specifiers

In LXGram almost all adjectives, prepositions and adverbs are allowed to combine with degree specifiers/modifiers. The exceptions are only lexical items that are saliently ungradable, such as some types of scopal adjectives discussed below in Section 7.9.

The co-occurrence of degree specifiers is however controlled on the basis of gradability categories, which are discussed below, but in general degree specification of all prepositions and adverbs and practically all adjectives is allowed. (16) shows examples of degree specification of prepositions ("a"), adjectives ("interessante") and adverbs/degree specifiers ("tarde", "mais").

(16) a. Estacionei o carro num lugar mais à esquerda.
I parked the car in a place more to the left
I parked the car in a place more to the left.

b. Isso é muito mais interessante.
that is much more interesting
That is much more interesting.

c. Ele chegou muito mais tarde.
he arrived much more late
He arrived much later.

In LXGram intersective semantics is given to degree specification, as can be seen in Figure 7.6, and, as in the ERG, an ARG0 is provided for the relations of the degree specifiers, which allows unambiguous representations of recursive degree specification as depicted in the referred figure. Both "mais" (*more*) and"do que" (*than*) are assigned a single relation (*mais_x_do-que_rel*).

7.5 Conjunction Phrases

At this stage, the "do que" phrase is allowed to introduce nominative NPs only. A more general implementation would also allow full or elliptical sentences, but that is postponed to phase B.9.

$$\begin{bmatrix} \textit{mrs} & \\ \text{RELS} & \left\langle \begin{array}{l} \ldots, \begin{bmatrix} \textit{\_chegar\_v\_rel} & \\ \text{LBL} & \boxed{h10}\ h \\ \text{ARG0} & \boxed{e2} \\ \text{ARG1} & \boxed{x6} \end{bmatrix}, \begin{bmatrix} \textit{\_muito\_x\_rel} & \\ \text{LBL} & \boxed{h10} \\ \text{ARG0} & \boxed{e11}\ e \\ \text{ARG1} & \boxed{e12}\ e \end{bmatrix}, \begin{bmatrix} \textit{\_mais\_x\_do-que\_rel} & \\ \text{LBL} & \boxed{h10} \\ \text{ARG0} & \boxed{e12} \\ \text{ARG1} & \boxed{e14}\ e \\ \text{ARG2} & \boxed{x13}\ x \end{bmatrix}, \\ \begin{bmatrix} \textit{\_tarde\_a\_rel} & \\ \text{LBL} & \boxed{h10} \\ \text{ARG0} & \boxed{e14} \\ \text{ARG1} & \boxed{e2} \end{bmatrix}, \begin{bmatrix} \textit{pronoun\_q\_rel} & \\ \text{LBL} & \boxed{h15}\ h \\ \text{ARG0} & \boxed{x13} \\ \text{RSTR} & \boxed{h16}\ h \\ \text{BODY} & \boxed{h17}\ h \end{bmatrix}, \begin{bmatrix} \textit{pronoun\_n\_rel} & \\ \text{LBL} & \boxed{h18}\ h \\ \text{ARG0} & \boxed{x13} \end{bmatrix} \end{array} \right\rangle \end{bmatrix}$$

Figure 7.6: Excerpt from the MRS for the sentence "Eu cheguei muito mais tarde do que tu" (*I arrived much later than you* - lit. *much more late*).

Syntactically, the "do que" (*than*) phrase is analyzed as a complement of the degree specifiers "mais" and "menos", and it must be passed up so that it is projected to the right of the specified head, as the examples in (17) shows.

Furthermore, word order is strict in that the "do que" complement cannot occur between the degree word and its head, so one needs to prevent head-complement rules to apply directly to these items. This cannot be analyzed by constraining the specifier to be a single word, since a degree word is allowed to have degree dependents of its own.

A third requirement is that, when there are multiple degree specifiers, the appropriate complement must be passed up. So in the example (17c) below it is the "do que" complement of "mais" (*more*) that needs to be copied, not the empty COMPS list of "muito" (*much*).

The example sentences in (17) illustrate these observations, and Figure 7.7 and Figure 7.8 present the parse trees that LXGram assigns to these (where ADVP labels an adverb-headed constituent with empty complements, and ADV an adverb-headed one with non-empty complements).

(17) a.

Eu	cheguei	mais	tarde	do que	tu.
I	arrived	more	late	than	you

I arrived later than you.

b.

*	Eu	cheguei	mais	do que	tu	tarde.
	I	arrived	more	than	you	late

c.

Eu	cheguei	muito	mais	tarde	do que	tu.
I	arrived	much	more	late	than	you.

I arrived much later than you.

As mentioned in Section 5.3, in LXGram the notions of specifier and adjunct have been replaced by functors, so degree specifiers are not treated as specifiers but rather as functors. In any case, what is required is that some instances of functor-head schemata will pass the functor complements up, while others pass those of the head. Additionally, one needs to control which functors can undergo which types of head-functor rules, since not all pre-head specifiers and modifiers can have their complements projected to the right of the head they attach to. This will also solve the third desideratum: "mais" (*more*) must have its complements percolated, "muito" (*much*) must not.

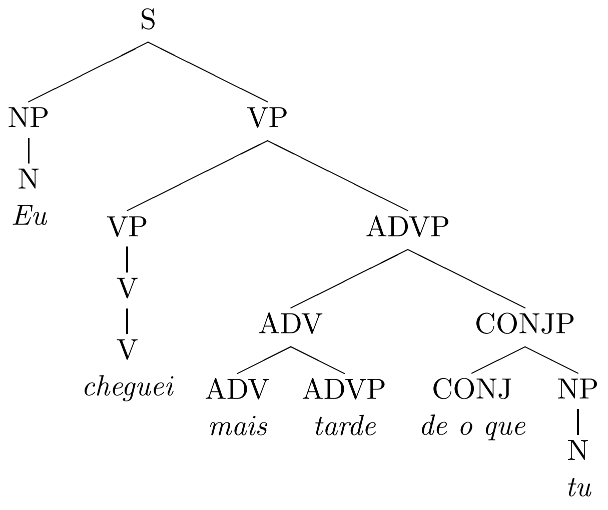


Figure 7.7: Parse tree for sentence (17a).

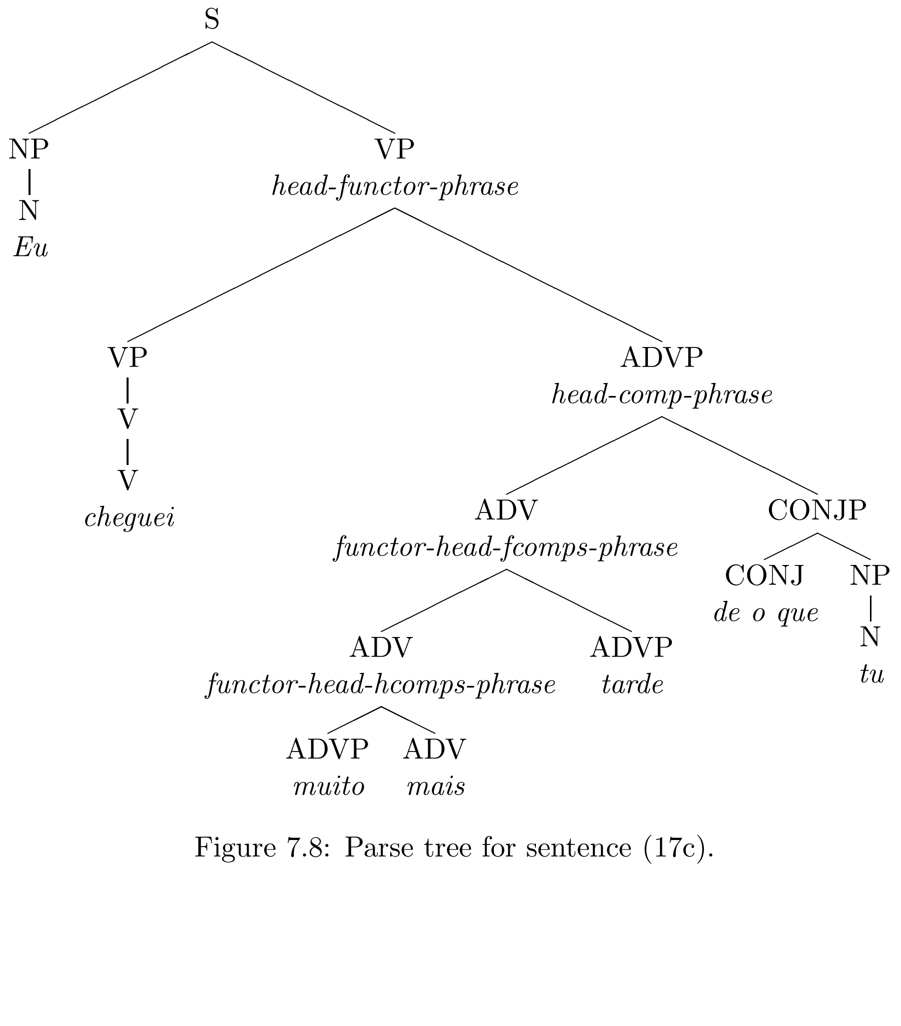


Figure 7.8: Parse tree for sentence (17c).

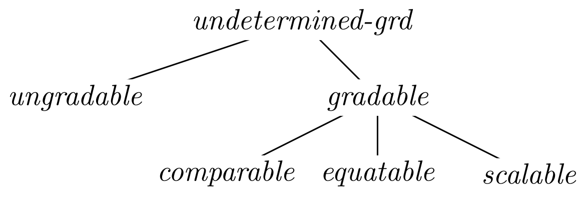


Figure 7.9: Simplified type hierarchy under *undetermined-grd.*

The analysis that is implemented makes available two versions of head-final Head-Functor phrases. One of these versions percolates the complements of the head (*functor-head-hcomps-phrase*), the other version percolates the complements of the functor (*functor-head-fcomps-phrase*). Head-initial Head-Functor constructions always percolate the complements of the head. The relevant nodes in the tree in Figure 7.8 are decorated with the names of the phrases involved. Rule application is controlled via an additional feature: COMPS-POSITION. The value of this feature in the functor daughter determines whether *functor-head-hcomps-phrase* or *functor-head-fcomps-phrase* is used.

The sentence in (18) is also considered ungrammatical (it would require some functor phrase that appends the COMPS of both of its daughters).

(18) *O Pedro é mais mais alto do que a Maria do que a Ana.
the Pedro is more more tall than the Maria than the Ana
intended: *Pedro is taller than Maria more than Ana is.*

As was mentioned, constrained recursion of degree specification is allowed. On the one hand, degree specifiers can have a degree specifier of their own. On the other hand, their co-occurrence is constrained. An attribute GRADABLE [1] is employed with a type *undetermined-grd* with the values presented in Figure 7.9.

For instance, to block the co-occurrence of two “mais” (*more*), as in example (18), “mais” selects a head with GRADABLE *comparable* but its own GRADABLE feature is of type *scalable* (specifiable by e.g. “muito” - *much*). An element that can be modified by “tão ... como/quanto” (*as ... as*) has a GRADABLE feature with a value unifiable with *equatable.*

The hierarchy of *undetermined-grd* presented is actually simpler than the one implemented, because some details have been omitted. For example, “mais” must actually constrain the head it selects to be *comparable by “mais”* (i.e. there are subtypes of *comparable*), because some adjectives that have synthetic comparatives cannot form analytical ones (“*mais bom”), but can nevertheless be specified by “menos” (*less*), which also selects a head with a GRADABLE feature of (a subtype of) type *comparable.*

Forms of adjectives that are referred to in traditional Portuguese grammar as absolute superlatives (forms ending in “-íssimo”: “contentíssimo” — *very happy*) are treated in a similar manner. They are of course generated by a lexical rule. It adds a semantic relation similar to that of “muito” (*very*), but it is different from degree specification by “muito” in two mais respects: LXGram does not commit to saying that “-íssimo” suffixation and modification by “muito” are semantically equivalent — the relation introduced by this rule has the name *íssimo_x_rel* (Figure 7.10 shows the MRS for sentence (20a) below) —; its output is constrained to have a GRADABLE feature with the value *ungradable*, whereas “muito” has the value *scalable* (see (19–20)).

(19) a. Estou muito contente.
I am very happy
I am very happy.

[1] GRADABLE is a feature of HEAD.

$$\begin{bmatrix} \textit{mrs} & \\ \text{LTOP} & \boxed{h1}\ h \\ \text{INDEX} & \boxed{e2}\ e \\ \text{RELS} & \left\langle \begin{bmatrix} \textit{pronoun\_q\_rel} & \\ \text{LBL} & \boxed{h3}\ h \\ \text{ARG0} & \boxed{x4}\ x \\ \text{RSTR} & \boxed{h5}\ h \\ \text{BODY} & \boxed{h6}\ h \end{bmatrix}, \begin{bmatrix} \textit{pronoun\_n\_rel} & \\ \text{LBL} & \boxed{h7}\ h \\ \text{ARG0} & \boxed{x4} \end{bmatrix}, \begin{bmatrix} \textit{tam\_rel} & \\ \text{LBL} & \boxed{h8}\ h \\ \text{ARG0} & \boxed{e2} \end{bmatrix}, \begin{bmatrix} \textit{íssimo\_x\_rel} & \\ \text{LBL} & \boxed{h8} \\ \text{ARG0} & \boxed{e9}\ e \\ \text{ARG1} & \boxed{e2} \end{bmatrix}, \begin{bmatrix} \textit{\_contente\_a\_-com-\_rel} & \\ \text{LBL} & \boxed{h8} \\ \text{ARG0} & \boxed{e2} \\ \text{ARG1} & \boxed{x4} \\ \text{ARG2} & \boxed{u10}\ u \end{bmatrix} \right\rangle \\ \text{HCONS} & \left\langle \begin{bmatrix} \textit{qeq} & \\ \text{HARG} & \boxed{h1} \\ \text{LARG} & \boxed{h8} \end{bmatrix}, \begin{bmatrix} \textit{qeq} & \\ \text{HARG} & \boxed{h5} \\ \text{LARG} & \boxed{h7} \end{bmatrix} \right\rangle \end{bmatrix}$$

Figure 7.10: MRS for a sentence with an "-íssimo" form. The sentence is "Estou contentíssimo" (*I am very happy*).

b. Estou muito muito contente.
I am very very happy
I am very very happy.

c. * Estou mais muito contente.
I am more very happy

d. * Estou tão muito contente.
I am as very happy

(20) a. Estou contentíssimo.
I am very happy
I am very happy.

b. * Estou muito contentíssimo.
I am very very happy

c. * Estou mais contentíssimo.
I am more very happy

d. * Estou tão contentíssimo.
I am as very happy

7.5.1 Postponed Coverage or Known Limitations

The analysis of degree specifiers began before the functor schemata were implemented, and the first implementation treated them as specifiers, so they are not allowed to modify verbs. With the functor analysis, they are no longer specifiers and modification of verbs by these items can now be implemented without requiring multiple lexical entries. This has not been completed yet.

The current implementation of the "-íssimo" lexical rule applies only to adjectives, but some adverbs can also present these forms (e.g. "cedíssimo", *very early*; "cedo" *early* is never an adjective in Portuguese)).

7.6 Adverb Phrases

In LXGram non degree adverbs are distinguished in several dimensions:

- whether they can modify nouns and verbs or only verbs

(21) a. O Pedro [comprou um livro] assim.
the Pedro bought a book so
Pedro [bought a book] this way.

b. O Pedro comprou [um livro assim].
the Pedro bought a book so
Pedro bought [a book like this].

c. O Pedro [comprou um livro] depressa.
the Pedro bought a book quickly
Pedro [bought a book] quickly.

This is implemented via constraints on the feature SYNSEM|LOCAL|CAT|HEAD|MARKER|SELECT. Namely, for adverbs in general the sole element in this list has a head of type *noun-or-verb* (a supertype of nouns and verbs), but adverbs that can modify only verbs constrain it to be of the type *verb*.

- whether they can be a predicate selected by the copular verbs

(22) a. Isso é assim.
that is so
That is like that.

b. * Isso é rapidamente.
that is quickly

Since the copular verbs raise their complement's subject (Section 5.13.1), the implementation distinguishes between adverbs that have an empty SUBJ valence list and those that select a subject.

- whether they have PP complements

(23) a. Ele saiu depois.
he went out afterwards
He left afterwards.

b. Ele saiu depois da festa.
he went out after of the party
He left after the party.

Being a question of valence, the implementation obviously distinguishes between adverbs with an empty COMPS list from those with a non empty one. As the examples in (23) illustrate, the complement is always optional. It is also always a PP, but the preposition may vary. The current implementation distinguishes among adverbs with a PP headed by "com" ("a compasso com" *simultaneously with*), "de" (e.g. "depois de" *after*) and by "de" or "a" ("junto a/de" *next to*).

- whether their semantics is intersective ("depressa" in the example below) or scopal ("possivelmente")

$$\left[\begin{array}{ll}
\textit{mrs} & \\
\text{LTOP} & \boxed{h1}\ h\\
\text{INDEX} & \boxed{e2}\ e\\
\text{RELS} & \left\langle \left[\begin{array}{ll}\textit{pronoun\_q\_rel} & \\ \text{LBL} & \boxed{h3}\ h\\ \text{ARG0} & \boxed{x4}\ x\\ \text{RSTR} & \boxed{h6}\ h\\ \text{BODY} & \boxed{h5}\ h\end{array}\right], \left[\begin{array}{ll}\textit{pronoun\_n\_rel} & \\ \text{LBL} & \boxed{h7}\ h\\ \text{ARG0} & \boxed{x4}\end{array}\right], \left[\begin{array}{ll}\textit{\_possível\_a\_rel} & \\ \text{LBL} & \boxed{h8}\ h\\ \text{ARG0} & \boxed{e9}\ e\\ \text{ARG1} & \boxed{h10}\ h\end{array}\right], \left[\begin{array}{ll}\textit{tam\_rel} & \\ \text{LBL} & \boxed{h11}\ h\\ \text{ARG0} & \boxed{e2}\end{array}\right], \left[\begin{array}{ll}\textit{\_chegar\_v\_rel} & \\ \text{LBL} & \boxed{h11}\\ \text{ARG0} & \boxed{e2}\\ \text{ARG1} & \boxed{x4}\end{array}\right] \right\rangle\\
\text{HCONS} & \left\langle \left[\begin{array}{ll}\textit{qeq} & \\ \text{HARG} & \boxed{h1}\\ \text{LARG} & \boxed{h8}\end{array}\right], \left[\begin{array}{ll}\textit{qeq} & \\ \text{HARG} & \boxed{h6}\\ \text{LARG} & \boxed{h7}\end{array}\right], \left[\begin{array}{ll}\textit{qeq} & \\ \text{HARG} & \boxed{h10}\\ \text{LARG} & \boxed{h11}\end{array}\right] \right\rangle
\end{array}\right]$$

Figure 7.11: MRS for a sentence with a scopal adverb. The sentence is "Ele possivelmente chegou" (*He possibly arrived*).

(24) a. Ele possivelmente chegou.
he possibly arrived
He has possibly arrived.

b. Ele chegou depressa.
he arrived quickly
He arrived quickly.

Figure 7.11 and Figure 7.12 show the MRSs produced by LXGram for these two sentences.

Intersective modifiers have an empty HCONS difference list and unify the INDEX feature of the synsem inside their SELECT list with the ARG1 of their elementary predication.

Scopal adverbs have an HCONS with a *qeq* that relates their ARG1 with the LTOP of the element in their SELECT attribute.

- whether they trigger proclisis

(25) a. Ele então comprou-o.
he then bought it
He then bought it.

b. Ele já o comprou.
he already it bought
He has already bought it.

Control of clitic placement is still embryonic, but there is a common supertype for elements that trigger proclisis (*proclisis-trigger-verbal-adjunct-item*, defined in the file named `lexical-types.tdl`) and another for those that do not (*non-proclisis-trigger-verbal-adjunct-item*, defined in the same file), whose constraints can be changed in the future when a proper analysis of this phenomenon is implemented (phases C.13 and E.19 of the implementation agenda). But lexical items are already marked according to this property.

- whether they can occur preverbally, postverbally or both

$$\left[\begin{array}{ll}
\textit{mrs} & \\
\text{LTOP} & \boxed{h1}\ h \\
\text{INDEX} & \boxed{e2}\ e \\
\text{RELS} & \left\langle
\left[\begin{array}{ll}\textit{pronoun\_q\_rel} & \\ \text{LBL} & \boxed{h3}\ h \\ \text{ARG0} & \boxed{x4}\ x \\ \text{RSTR} & \boxed{h6}\ h \\ \text{BODY} & \boxed{h5}\ h\end{array}\right],
\left[\begin{array}{ll}\textit{pronoun\_n\_rel} & \\ \text{LBL} & \boxed{h7}\ h \\ \text{ARG0} & \boxed{x4}\end{array}\right],
\left[\begin{array}{ll}\textit{tam\_rel} & \\ \text{LBL} & \boxed{h8}\ h \\ \text{ARG0} & \boxed{e2}\end{array}\right],
\left[\begin{array}{ll}\textit{\_chegar\_v\_rel} & \\ \text{LBL} & \boxed{h8} \\ \text{ARG0} & \boxed{e2} \\ \text{ARG1} & \boxed{x4}\end{array}\right],
\left[\begin{array}{ll}\textit{\_depressa\_a\_rel} & \\ \text{LBL} & \boxed{h8} \\ \text{ARG0} & \boxed{e9}\ e \\ \text{ARG1} & \boxed{e2}\end{array}\right]
\right\rangle \\
\text{HCONS} & \left\langle
\left[\begin{array}{ll}\textit{qeq} & \\ \text{HARG} & \boxed{h1} \\ \text{LARG} & \boxed{h8}\end{array}\right],
\left[\begin{array}{ll}\textit{qeq} & \\ \text{HARG} & \boxed{h6} \\ \text{LARG} & \boxed{h7}\end{array}\right]
\right\rangle
\end{array}\right]$$

Figure 7.12: MRS for a sentence with an intersective adverb. The sentence is "Ele chegou depressa" (*He arrived quickly*).

(26) a. Eu mal tinha estacionado o carro.
I "mal" (scopal) had parked the car.
I had hardly/just parked the car.

b. Eu tinha estacionado o carro mal.
I had parked the car "mal" (intersective)
I had parked the car badly.

This sort of information is also lexically specified. For the example above there are two lexical entries for "mal": an intersective adverb that can occur only postverbally and a scopal adverb that can occur only preverbally.

Word order between head and functor is implemented by constraints on the type of the feature MARKER of functors (see Section 5.3.2).

7.7 Discourse markers, interjections

LXGram currently supports special items that tend to occur at the beginning of sentences, like "bom", "ora" (*well*) in the examples (27) below.

LXGram clusters them with interjections and greeting words (some examples of these also in (27).

(27) a. Bom, isso é assim.
well, that is like that
Well, that is like that.

b. Ah, está bem.
oh, it is all right
Oh, it is all right.

c. Olá, estás bem?
Hi, are you OK
Hi, are you OK?

$$\begin{bmatrix} \textit{mrs} & \\ \text{LTOP} & \boxed{h1}\ h \\ \text{INDEX} & \boxed{e2}\ e \\ \text{RELS} & \left\langle \begin{bmatrix} \textit{discourse\_rel} & \\ \text{LBL} & \boxed{h1} \\ \text{C-ARG} & \boxed{e2} \\ \text{L-HNDL} & \boxed{h4}\ h \\ \text{L-INDEX} & \boxed{e5}\ e \\ \text{R-HNDL} & \boxed{h6}\ h \\ \text{R-INDEX} & \boxed{e3}\ e \end{bmatrix}, \begin{bmatrix} \textit{greeting\_rel} & \\ \text{LBL} & \boxed{h4} \\ \text{ARG0} & \boxed{e5} \\ \text{GREETING} & \textit{olá} \end{bmatrix}, \begin{bmatrix} \textit{pronoun\_q\_rel} & \\ \text{LBL} & \boxed{h7}\ h \\ \text{ARG0} & \boxed{x8}\ x \\ \text{RSTR} & \boxed{h9}\ h \\ \text{BODY} & \boxed{h10}\ h \end{bmatrix}, \begin{bmatrix} \textit{pronoun\_n\_rel} & \\ \text{LBL} & \boxed{h11}\ h \\ \text{ARG0} & \boxed{x8} \end{bmatrix}, \begin{bmatrix} \textit{tam\_rel} & \\ \text{LBL} & \boxed{h12}\ h \\ \text{ARG0} & \boxed{e3} \end{bmatrix}, \begin{bmatrix} \textit{\_bem\_a\_rel} & \\ \text{LBL} & \boxed{h12} \\ \text{ARG0} & \boxed{e3} \\ \text{ARG1} & \boxed{x8} \end{bmatrix} \right\rangle \\ \text{HCONS} & \left\langle \begin{bmatrix} \textit{qeq} & \\ \text{HARG} & \boxed{h6} \\ \text{LARG} & \boxed{h12} \end{bmatrix}, \begin{bmatrix} \textit{qeq} & \\ \text{HARG} & \boxed{h9} \\ \text{LARG} & \boxed{h11} \end{bmatrix} \right\rangle \end{bmatrix}$$

Figure 7.13: MRS of a sentence including a discourse element. The sentence is "Olá, estás bem?" (*Hi, are you OK?*).

These peripheral adverbs, as well as interjections and greeting words, are given the semantics shown in Figure 7.13 (attribute EXCLAMATION is used instead of GREETING for interjections, and there is no corresponding attribute for the peripheral adverbials, these are distinguished by the relation name).

There is a dedicated syntactic rule (*clause_pre-root* in Section 9) to project them to the left of a root sentence. Because of this, they also receive a dedicated head type (*discourse-element* under the type *adv*) so that only they can be the left daughter of this rule (which constrains the left daughter to have this type of HEAD).

7.7.1 Postponed Coverage or Known Limitations

The current implementation is embryonic: it does not support utterances that consist only of a single element of this type and it does not allow for their occurrence in other positions.

7.8 Prepositional Phrases

The lexical types for prepositions in LXGram vary with respect to the following:

- type of complement they take: NP, VP

(28) a. Vou para [o jardim $_{\text{NP}}$].
I go to the garden
I'm going to the garden.

b. Acabei por [ir lá $_{\text{VP}}$].
I ended up by going there
I ended up going there.

At this stage the implementation distinguishes between prepositions taking an NP complement and prepositions with an NP or VP complement.

The VP prepositions that have been implemented so far all lack semantic content (see below), and they can also all take an NP complement as well. Prepositions without content have a HEAD of type *particle*.

This type introduces a feature COMP-HEAD, which is of type *noun-or-verb*, a supertype of *noun* and *verb* and of no other descendant of *head*. The lexical types that implement semantically void prepositions unify this feature with feature HEAD of their complement, so for the time being that complement can only be an NP or a VP.

The type of complement a semantically void preposition takes is thus visible in its HEAD, so that a head that selects a PP argument can distinguish locally between a PP that is semantically an NP and a PP that is semantically a VP.

Two subtypes of *particle* (*particle-np* and *particle-vp*) are used directly by predicators that select PPs with dummy prepositions, since general constraints are enforced in this subtypes. *particle-np* forces its complement to show oblique case, i.e. its COMP-HEAD (which is shared with the HEAD of the complement of the preposition, as stated above) is constrained to have CASE of type *oblique*, where CASE is a feature of *noun* where case is represented. *particle-vp* is constrained to have a COMP-HEAD with feature VFORM of type *infinitivo* (*infinitive*), where VFORM is an appropriate feature of *verb* where mood information is stored; hence other verb forms are blocked.

- whether they have semantic content and whether they can be modifiers

(29) a. Venho do jardim.
I come from the garden
I come from the garden.

b. Gosto do jardim.
I like of the garden
I like the garden.

As far as semantic content is concerned, the only implemented distinction is the one between the case when a preposition contributes no semantic relations and the case when it does. For instance, in the example (29b) the preposition "de" is considered empty and in the resulting semantics its complement is an argument of the verbal relation (see Figure 7.15). In the example (29a) the preposition is considered to denote a relation between the situation denoted by the verb and the denotation of "o jardim", but the exact sense used in particular examples is not determined: "de" is always given a *_de_p_rel* relation. The same applies to all other prepositions. See Figure 7.14 for the full semantic representation given to that sentence by the grammar.

Prepositions that do not contribute any semantic relations are constrained so that they cannot be modifiers, and hence can only appear in a parse tree if they are selected by a head. More specifically, the type *particle* does not inherit from *functor* and hence does not have the feature MARKER.

In order to implement these distinctions, multiple lexical entries are used for prepositions that can sometimes be contentful modifiers and sometimes semantically empty and selected by a head (such as "de"). There are thus two lexical entries for "de": one that has an empty

$$\left[\begin{array}{ll}
\textit{mrs} & \\
\text{LTOP} & \boxed{h1}\ h \\
\text{INDEX} & \boxed{e2}\ e \\
\text{RELS} & \left\langle \begin{array}{l}
\left[\begin{array}{ll} \textit{pronoun\_q\_rel} & \\ \text{LBL} & \boxed{h3}\ h \\ \text{ARG0} & \boxed{x4}\ x \\ \text{RSTR} & \boxed{h5}\ h \\ \text{BODY} & \boxed{h6}\ h \end{array}\right],
\left[\begin{array}{ll} \textit{pronoun\_n\_rel} & \\ \text{LBL} & \boxed{h7}\ h \\ \text{ARG0} & \boxed{x4} \end{array}\right],
\left[\begin{array}{ll} \textit{tam\_rel} & \\ \text{LBL} & \boxed{h8}\ h \\ \text{ARG0} & \boxed{e2} \end{array}\right],
\left[\begin{array}{ll} \textit{\_vir\_v\_rel} & \\ \text{LBL} & \boxed{h8} \\ \text{ARG0} & \boxed{e2} \\ \text{ARG1} & \boxed{x4} \end{array}\right], \\
\left[\begin{array}{ll} \textit{\_de\_p\_rel} & \\ \text{LBL} & \boxed{h8} \\ \text{ARG0} & \boxed{e9}\ e \\ \text{ARG1} & \boxed{e2} \\ \text{ARG2} & \boxed{x10}\ x \end{array}\right],
\left[\begin{array}{ll} \textit{\_o\_q\_rel} & \\ \text{LBL} & \boxed{h11}\ h \\ \text{ARG0} & \boxed{x10} \\ \text{RSTR} & \boxed{h13}\ h \\ \text{BODY} & \boxed{h12}\ h \end{array}\right],
\left[\begin{array}{ll} \textit{\_jardim\_n\_rel} & \\ \text{LBL} & \boxed{h14}\ h \\ \text{ARG0} & \boxed{x10} \end{array}\right]
\end{array} \right\rangle \\
\text{HCONS} & \left\langle
\left[\begin{array}{ll} \textit{qeq} & \\ \text{HARG} & \boxed{h1} \\ \text{LARG} & \boxed{h8} \end{array}\right],
\left[\begin{array}{ll} \textit{qeq} & \\ \text{HARG} & \boxed{h5} \\ \text{LARG} & \boxed{h7} \end{array}\right],
\left[\begin{array}{ll} \textit{qeq} & \\ \text{HARG} & \boxed{h13} \\ \text{LARG} & \boxed{h14} \end{array}\right]
\right\rangle
\end{array}\right]$$

Figure 7.14: MRS for the sentence “Venho do jardim” (*I come from the garden*).

$$\left[\begin{array}{ll}
\textit{mrs} & \\
\text{LTOP} & \boxed{h1}\ h \\
\text{INDEX} & \boxed{e2}\ e \\
\text{RELS} & \left\langle \begin{array}{l}
\left[\begin{array}{ll} \textit{pronoun\_q\_rel} & \\ \text{LBL} & \boxed{h3}\ h \\ \text{ARG0} & \boxed{x4}\ x \\ \text{RSTR} & \boxed{h5}\ h \\ \text{BODY} & \boxed{h6}\ h \end{array}\right],
\left[\begin{array}{ll} \textit{pronoun\_n\_rel} & \\ \text{LBL} & \boxed{h7}\ h \\ \text{ARG0} & \boxed{x4} \end{array}\right],
\left[\begin{array}{ll} \textit{tam\_rel} & \\ \text{LBL} & \boxed{h8}\ h \\ \text{ARG0} & \boxed{e2} \end{array}\right], \\
\left[\begin{array}{ll} \textit{\_gostar\_v\_-de-\_rel} & \\ \text{LBL} & \boxed{h8} \\ \text{ARG0} & \boxed{e2} \\ \text{ARG1} & \boxed{x4} \\ \text{ARG2} & \boxed{x9}\ x \end{array}\right],
\left[\begin{array}{ll} \textit{\_o\_q\_rel} & \\ \text{LBL} & \boxed{h10}\ h \\ \text{ARG0} & \boxed{x9} \\ \text{RSTR} & \boxed{h12}\ h \\ \text{BODY} & \boxed{h11}\ h \end{array}\right],
\left[\begin{array}{ll} \textit{\_jardim\_n\_rel} & \\ \text{LBL} & \boxed{h13}\ h \\ \text{ARG0} & \boxed{x9} \end{array}\right]
\end{array} \right\rangle \\
\text{HCONS} & \left\langle
\left[\begin{array}{ll} \textit{qeq} & \\ \text{HARG} & \boxed{h1} \\ \text{LARG} & \boxed{h8} \end{array}\right],
\left[\begin{array}{ll} \textit{qeq} & \\ \text{HARG} & \boxed{h5} \\ \text{LARG} & \boxed{h7} \end{array}\right],
\left[\begin{array}{ll} \textit{qeq} & \\ \text{HARG} & \boxed{h12} \\ \text{LARG} & \boxed{h13} \end{array}\right]
\right\rangle
\end{array}\right]$$

Figure 7.15: MRS for the sentence “Gosto do jardim” (*I like the garden*).

RELS and a HEAD of type *particle* and a second one with a HEAD of type *preposition* and a non-empty RELS.

Semantically void prepositions also have an extra feature under HEAD, PFORM, where their form is stated. This allows a verb to select for instance a PP whose head is preposition "de" and no other.

The following aspects are also accounted for:

- adjunction sites: nominal vs. verbal
  When a PP adjoins to a nominal category, one needs to say that it must follow the head, but when it modifies a verbal projection it can precede it as well. This is implemented through constraints on the features PREHEAD and POSTHEAD explained in Section 5.3.2.
- predicative use
  Semantically vacuous prepositions are prevented from being modifiers and also from being predicates selectable by copular verbs. Semantically vacuous prepositions are treated as raising the subject of their complement, since they can occur with a VP complement and be the complement of raising verbs. So the copular verbs must impose more constraints on their complement besides requiring them to have a subject — see Section 5.13.1. Incompatible HEAD types are currently employed to constrain this. There is an abstract type *predicational-head* that is a supertype of *preposition* (as well as other types that can be the head of the complement of a copula) but not of *particle*.

7.8.1 Postponed Coverage or Known Limitations

Prepositions taking a VP complement and contributing a semantic relation have not been implemented yet.

7.9 Adjective Phrases

Several types of adjectives have been implemented, differing in the following aspects:

- number and type of complements

  The implementation distinguishes between adjectives with zero complements and a one-place relation (ignoring their event argument) and with one complement and a two place relation. There are lexical types for the cases where the complement is a PP headed by "a", "com", "de" and "para" and selecting a noun phrase, illustrated with the sentences in (30).

(30) a.

Este	computador	é	igual	a	esse.
this	computer	is	equal	to	that one

This computer is equal to that one.

b.

O	cliente	está	feliz	com	a	compra.
the	client	is	happy	with	the	purchase

The client is happy with his purchase.

c.

Este	computador	é	diferente	desse.
this	computer	is	different	from that one

This computer is different from that one.

d.

Os	funcionários	são	simpáticos	para	os	clientes.
the	employees	are	nice	to	the	clients

The employees are nice to the clients.

The implementation resorts to a PFORM attribute of the preposition. For instance, adjectives that select a complement headed by the preposition "de" taking an NP complement (e.g. "diferente", *different*) are assigned a lexical type where their COMPS list is specified to contain a single element with a HEAD of type *particle-np* (see the Section 7.8) with the feature PFORM of type *de*.

This treatment obviously does not cover all subcategorization frames of adjectives.

- relative word order between adjective and noun

  The relative word order between adjective and noun is controlled by constraints in the lexical entries of adjectives. The implementation is explained in Section 5.3.

(31) a. Tenho um carro amarelo.
I have a car yellow
I have a yellow car.

b. * Tenho um amarelo carro.
I have a yellow car

c. Tenho um fantástico carro.
I have an amazing car
I have an amazing car.

- predicative use

  The possibility of an adjective to be used predicatively or not is also controlled in the lexicon, according to the contrast in the examples below. Since the copular verbs are implemented as raising their complement's subject (Section 5.13.1), adjectives that cannot appear in these contexts are lexically specified to have an empty SUBJ list.

(32) a. O meu carro é amarelo.
the my car is yellow
My car is yellow.

b. * O meu carro é mero.
the my car is mere

- choice of copular verb

  There are two copular verbs in Portuguese corresponding to English *to be*. One is for individual predicates ("ser"), another for stage-level predicates ("estar"). The sentences in (33) exemplify the difference in meaning.

(33) a. És tão grande!
you are ("ser") so big
You are so tall!

b. Estás tão grande!
you are ("estar") so big
You are so grown up!

Of course, when an adjective or other element that can be selected by a copular verb occurs attributively, it will show an ambiguity between these two readings, so it is important that this information appears in MRSs in a way that allows underspecification.

$$\begin{bmatrix} \textit{mrs} & \\ \text{LTOP} & \boxed{h1}\ h \\ \text{INDEX} & \boxed{e2}\begin{bmatrix} e & \\ \text{SF} & \textit{proposition} \\ \text{ELLIPTICAL-PUNCT} & - \\ \text{E.TENSE} & \textit{presente} \\ \text{E.ASPECT.PERF} & - \\ \text{E.MOOD} & \textit{indicativo} \\ \text{PRED-TYPE} & \textit{individual-predicate} \end{bmatrix} \\ \text{RELS} & \left\langle \begin{matrix} \begin{bmatrix} \textit{pronoun\_q\_rel} & \\ \text{LBL} & \boxed{h3}\ h \\ \text{ARG0} & \boxed{x4}\ x \\ \text{RSTR} & \boxed{h5}\ h \\ \text{BODY} & \boxed{h6}\ h \end{bmatrix}, \begin{bmatrix} \textit{pronoun\_n\_rel} & \\ \text{LBL} & \boxed{h7}\ h \\ \text{ARG0} & \boxed{x4} \end{bmatrix}, \begin{bmatrix} \textit{tam\_rel} & \\ \text{LBL} & \boxed{h8}\ h \\ \text{ARG0} & \boxed{e2} \end{bmatrix}, \\ \begin{bmatrix} \textit{\_tão\_x\_exclamative\_rel} & \\ \text{LBL} & \boxed{h8} \\ \text{ARG0} & \boxed{e9}\ e \\ \text{ARG1} & \boxed{e2} \end{bmatrix}, \begin{bmatrix} \textit{\_grande\_a\_isect\_rel} & \\ \text{LBL} & \boxed{h8} \\ \text{ARG0} & \boxed{e2} \\ \text{ARG1} & \boxed{x4} \end{bmatrix} \end{matrix} \right\rangle \\ \text{HCONS} & \left\langle \begin{bmatrix} \textit{qeq} & \\ \text{HARG} & \boxed{h1} \\ \text{LARG} & \boxed{h8} \end{bmatrix}, \begin{bmatrix} \textit{qeq} & \\ \text{HARG} & \boxed{h5} \\ \text{LARG} & \boxed{h7} \end{bmatrix} \right\rangle \end{bmatrix}$$

Figure 7.16: MRS for the sentence "És tão grande" (*You are so tall*).

One such way is to represent it under a dedicated attribute under events. LXGram uses a feature PRED-TYPE to this end. Figure 7.16 and Figure 7.17 show the MRSs assigned to the sentences in (33).

This is implemented by having copular verbs structure-share their event (feature INDEX) with that of their complements (which automatically adds to it the tense, aspect and mood information produced by the inflectional rules the verb undergoes). Additionally, "ser" constrains the PRED-TYPE feature of its event to take the value *individual-predicate*, "estar" constrains it to be *stage-level-predicate.*

The relevant issue here is that adjectives are not constrained for this feature: LXGram lets all adjectives co-occur with both copular verbs.

It is a deliberate decision. Adjectives like "eléctrico" (*electric*) have been reported to not co-occur with "estar", but one can easily find examples where these adjectives do co-occur with "estar", albeit with a shift in meaning ("eléctrico" then means *edgy, restless*). This shift in meaning seems productive.

Of course, a sentence like "esta máquina está eléctrica" (*this machine is electric*, with "estar") is thus strange, but parsed by LXGram. Its strangeness is just a matter of lexical semantics though (it means *this machine is restless*), which is not covered in LXGram.

$$\left[\begin{array}{ll}
\textit{mrs} & \\
\text{LTOP} & \boxed{h1}\ h \\
\text{INDEX} & \boxed{e2}\left[\begin{array}{ll}
e & \\
\text{SF} & \textit{proposition} \\
\text{ELLIPTICAL-PUNCT} & - \\
\text{E.TENSE} & \textit{presente} \\
\text{E.ASPECT.PERF} & - \\
\text{E.MOOD} & \textit{indicativo} \\
\text{PRED-TYPE} & \textit{stage-level-predicate}
\end{array}\right] \\
\text{RELS} & \left\langle \begin{array}{l}
\left[\begin{array}{ll}
\textit{pronoun\_q\_rel} & \\
\text{LBL} & \boxed{h3}\ h \\
\text{ARG0} & \boxed{x4}\ x \\
\text{RSTR} & \boxed{h5}\ h \\
\text{BODY} & \boxed{h6}\ h
\end{array}\right],
\left[\begin{array}{ll}
\textit{pronoun\_n\_rel} & \\
\text{LBL} & \boxed{h7}\ h \\
\text{ARG0} & \boxed{x4}
\end{array}\right],
\left[\begin{array}{ll}
\textit{tam\_rel} & \\
\text{LBL} & \boxed{h8}\ h \\
\text{ARG0} & \boxed{e2}
\end{array}\right], \\
\left[\begin{array}{ll}
\textit{\_to\_x\_exclamative\_rel} & \\
\text{LBL} & \boxed{h8} \\
\text{ARG0} & \boxed{e9}\ e \\
\text{ARG1} & \boxed{e2}
\end{array}\right],
\left[\begin{array}{ll}
\textit{\_grande\_a\_isect\_rel} & \\
\text{LBL} & \boxed{h8} \\
\text{ARG0} & \boxed{e2} \\
\text{ARG1} & \boxed{x4}
\end{array}\right]
\end{array} \right\rangle \\
\text{HCONS} & \left\langle
\left[\begin{array}{ll}
\textit{qeq} & \\
\text{HARG} & \boxed{h1} \\
\text{LARG} & \boxed{h8}
\end{array}\right],
\left[\begin{array}{ll}
\textit{qeq} & \\
\text{HARG} & \boxed{h5} \\
\text{LARG} & \boxed{h7}
\end{array}\right]
\right\rangle
\end{array}\right]$$

Figure 7.17: MRS for the sentence “Estás tão grande” (*You are so grown up*).

$$\begin{bmatrix} \textit{mrs} & \\ \text{LTOP} & \boxed{h1}\ h \\ \text{INDEX} & \boxed{e2}\ e \\ \text{RELS} & \left\langle \begin{array}{l} \begin{bmatrix} \textit{pronoun\_q\_rel} & \\ \text{LBL} & \boxed{h3}\ h \\ \text{ARG0} & \boxed{x4}\ x \\ \text{RSTR} & \boxed{h6}\ h \\ \text{BODY} & \boxed{h5}\ h \end{bmatrix}, \begin{bmatrix} \textit{pronoun\_n\_rel} & \\ \text{LBL} & \boxed{h7}\ h \\ \text{ARG0} & \boxed{x4} \end{bmatrix}, \begin{bmatrix} \textit{tam\_rel} & \\ \text{LBL} & \boxed{h8}\ h \\ \text{ARG0} & \boxed{e2} \end{bmatrix}, \begin{bmatrix} \textit{identity\_v\_rel} & \\ \text{LBL} & \boxed{h8} \\ \text{ARG0} & \boxed{e2} \\ \text{ARG1} & \boxed{x4} \\ \text{ARG2} & \boxed{x9}\ x \end{bmatrix}, \\ \begin{bmatrix} \textit{\_um\_q\_rel} & \\ \text{LBL} & \boxed{h10}\ h \\ \text{ARG0} & \boxed{x9} \\ \text{RSTR} & \boxed{h12}\ h \\ \text{BODY} & \boxed{h11}\ h \end{bmatrix}, \begin{bmatrix} \textit{\_pianista\_n\_rel} & \\ \text{LBL} & \boxed{h13}\ h \\ \text{ARG0} & \boxed{x9} \end{bmatrix}, \begin{bmatrix} \textit{\_louro\_a\_rel} & \\ \text{LBL} & \boxed{h13} \\ \text{ARG0} & \boxed{e14}\ e \\ \text{ARG1} & \boxed{x9} \end{bmatrix} \end{array} \right\rangle \\ \text{HCONS} & \left\langle \begin{bmatrix} \textit{qeq} & \\ \text{HARG} & \boxed{h1} \\ \text{LARG} & \boxed{h8} \end{bmatrix}, \begin{bmatrix} \textit{qeq} & \\ \text{HARG} & \boxed{h6} \\ \text{LARG} & \boxed{h7} \end{bmatrix}, \begin{bmatrix} \textit{qeq} & \\ \text{HARG} & \boxed{h12} \\ \text{LARG} & \boxed{h13} \end{bmatrix} \right\rangle \end{bmatrix}$$

Figure 7.18: MRS for the sentence "Ele é um pianista louro" (*He is a blond pianist*).

In any case, the semantic representation given by LXGram captures, however indirectly, the shift in meaning that these adjectives can suffer. Since PRED-TYPE will end up as an attribute of the adjective event (in its ARG0), one can just say that the *electric* sense of "eléctrico" is represented as an *_eléctrico_a_rel* relation with an ARG0|PRED-TYPE of type *individual-predicate* and it *restless* sense is represented by a relation with the same name but an ARG0|PRED-TYPE of type *stage-level-predicate.*

Note also that the *restless* sense of "eléctrico" is availabe with "ser" if the subject is animate. So once again there is an indication that lexical semantics is involved.

- scopal and intersective semantics

  There is also an implemented distinction between intersective ("louro" in examples (34)) and scopal ("competente" in (34)) adjectives.

(34) a.

Ele	é	um	pianista	louro.
he	is	a	pianist	blond

He is a blond pianist.

b.

Ele	é	um	pianista	competente.
he	is	a	pianist	competent

He is a competent pianist.

Figure 7.18 shows the MRS of a sentence with an intersective adjective, and Figure 7.19 the MRS of one containing a scopal adjective.

The examples below (35) show that sometimes this difference is correlated with word order constraints, which is also accounted for in LXGram, by having lexical types for intersective adjectives that can only occur postnominally (the *evil* meaning of "mau") and others for

$$\begin{bmatrix} \textit{mrs} & \\ \text{LTOP} & \boxed{h1}\ h \\ \text{INDEX} & \boxed{e2}\ e \\ \text{RELS} & \left\langle \begin{array}{l} \begin{bmatrix} \textit{pronoun\_q\_rel} & \\ \text{LBL} & \boxed{h3}\ h \\ \text{ARG0} & \boxed{x4}\ x \\ \text{RSTR} & \boxed{h6}\ h \\ \text{BODY} & \boxed{h5}\ h \end{bmatrix}, \begin{bmatrix} \textit{pronoun\_n\_rel} & \\ \text{LBL} & \boxed{h7}\ h \\ \text{ARG0} & \boxed{x4} \end{bmatrix}, \begin{bmatrix} \textit{tam\_rel} & \\ \text{LBL} & \boxed{h8}\ h \\ \text{ARG0} & \boxed{e2} \end{bmatrix}, \begin{bmatrix} \textit{identity\_v\_rel} & \\ \text{LBL} & \boxed{h8} \\ \text{ARG0} & \boxed{e2} \\ \text{ARG1} & \boxed{x4} \\ \text{ARG2} & \boxed{x9}\ x \end{bmatrix}, \\ \begin{bmatrix} \textit{\_um\_q\_rel} & \\ \text{LBL} & \boxed{h10}\ h \\ \text{ARG0} & \boxed{x9} \\ \text{RSTR} & \boxed{h12}\ h \\ \text{BODY} & \boxed{h11}\ h \end{bmatrix}, \begin{bmatrix} \textit{\_pianista\_n\_rel} & \\ \text{LBL} & \boxed{h13}\ h \\ \text{ARG0} & \boxed{x9} \end{bmatrix}, \begin{bmatrix} \textit{\_competente\_a\_rel} & \\ \text{LBL} & \boxed{h14}\ h \\ \text{ARG0} & \boxed{e15}\ e \\ \text{ARG1} & \boxed{h16}\ h \end{bmatrix} \end{array} \right\rangle \\ \text{HCONS} & \left\langle \begin{bmatrix} \textit{qeq} & \\ \text{HARG} & \boxed{h1} \\ \text{LARG} & \boxed{h8} \end{bmatrix}, \begin{bmatrix} \textit{qeq} & \\ \text{HARG} & \boxed{h6} \\ \text{LARG} & \boxed{h7} \end{bmatrix}, \begin{bmatrix} \textit{qeq} & \\ \text{HARG} & \boxed{h12} \\ \text{LARG} & \boxed{h14} \end{bmatrix}, \begin{bmatrix} \textit{qeq} & \\ \text{HARG} & \boxed{h16} \\ \text{LARG} & \boxed{h13} \end{bmatrix} \right\rangle \end{bmatrix}$$

Figure 7.19: MRS for the sentence "Ele é um pianista competente" (*He is a competent pianist*).

scopal adjectives that can occur prenominally or postnominally (the *bad* meaning of "mau"). There will thus be two lexical entries for an adjective like "mau" in the examples below.

(35) a. Ele é um mau pianista.
he is a bad pianist
He is a bad pianist.

b. Ele é um pianista mau.
he is a pianist bad
He is a bad/evil pianist.

But there are also intersective adjectives that can occur prenominally (as in (36)), as well as scopal ones that occur postnominally (as in (34b)), so there is the need to cross-classify adjectives in these two dimentions, which is implemented by making use of multiple inheritance in the hierarchy of lexical types.

(36) Ele é um famélico pianista.
he is a starving pianist
He is a starving pianist.

For scopal adjectives in predicative contexts, it is necessary to add a relation that can be the argument of these scopal relations, so that the resulting MRSs can be scope resolved. In LXGram this is implemented by forcing these adjectives to feed a unary rule that adds an *ellipsis_n_1_rel* relation, before they combine as the complement of a copular verb. An extra feature MOD-SEM under SYNSEM|LOCAL|CONT|HOOK is used to block scopal adjectives from merging as the complement of copular verbs directly. Figure 7.20 shows the MRS for a sentence with a scopal adjective as predicate.

$$\begin{bmatrix} \textit{mrs} & \\ \text{LTOP} & \boxed{h1}\ h \\ \text{INDEX} & \boxed{e2}\ e \\ \text{RELS} & \left\langle \begin{matrix} \begin{bmatrix}\textit{pronoun\_q\_rel} & \\ \text{LBL} & \boxed{h3}\ h \\ \text{ARG0} & \boxed{x4}\ x \\ \text{RSTR} & \boxed{h6}\ h \\ \text{BODY} & \boxed{h5}\ h \end{bmatrix}, \begin{bmatrix}\textit{pronoun\_n\_rel} & \\ \text{LBL} & \boxed{h7}\ h \\ \text{ARG0} & \boxed{x4} \end{bmatrix}, \\ \begin{bmatrix}\textit{tam\_rel} & \\ \text{LBL} & \boxed{h8}\ h \\ \text{ARG0} & \boxed{e2} \end{bmatrix}, \begin{bmatrix}\textit{ellipsis\_n\_1\_rel} & \\ \text{LBL} & \boxed{h9}\ h \\ \text{ARG0} & \boxed{x4} \end{bmatrix}, \begin{bmatrix}\textit{\_competente\_a\_rel} & \\ \text{LBL} & \boxed{h8} \\ \text{ARG0} & \boxed{e2} \\ \text{ARG1} & \boxed{h9} \end{bmatrix} \end{matrix} \right\rangle \\ \text{HCONS} & \left\langle \begin{bmatrix}\textit{qeq} & \\ \text{HARG} & \boxed{h1} \\ \text{LARG} & \boxed{h8} \end{bmatrix}, \begin{bmatrix}\textit{qeq} & \\ \text{HARG} & \boxed{h6} \\ \text{LARG} & \boxed{h7} \end{bmatrix} \right\rangle \end{bmatrix}$$

Figure 7.20: MRS for the sentences"Ele é competente" (*He is competent*).

- synthetic comparatives

  There are only a few synthetic comparatives in Portuguese: "maior" — *bigger*, "menor" — *smaller*, "melhor" — *better*, "pior" — *worse*. Therefore, they are simply listed in the lexicon.

  They are analyzed as carrying two semantic relations, one adjective relation and another similar to the one of the comparative particle "mais" – *more*.

  For example, the semantics of the form "maior" (*bigger*) is said to be *_mais_x_do-que_rel(e0, e1, u2)* ∧ *_grande_a_rel(e1, x3)*, i.e. "more big". Figure 7.21 shows an example MRS.

  The corresponding positive forms are also blocked from producing analytical comparative forms, except for "pequeno" (*small*), for which both comparative forms "menor" and "mais pequeno" are possible. An *ad hoc* feature of HEAD is used for this, since it is in fact exceptional and in no way general (it prevents the occurrence of three adjectives with the item "mais").

  Like the comparative "mais", these adjectives also have a "do que" (*than*) complement. The positions where this complement can be projected are described and analyzed in Section 5.3.

- "-íssimo" superlative forms

  For the "-íssimo" superlative forms (e.g. "pequeníssimo" – *very (very) small*, "cansadíssimo" – *very (very) tired*) a lexical rule from lexemes to lexemes is implemented.

  This rule adds a relation similar to the one of the item "muito" (*very*) and produces an ungradable adjective. However, LXGram does not commit to saying that "muito (muito) pequeno" and "pequeníssimo" are exact synonyms - a different name for the added relation is used: *íssimo_x_rel*. See also Section 7.4 and Chapter 10.

- gradability

$$\begin{bmatrix} \textit{mrs} & \\ \text{LTOP} & \boxed{h1}\ h \\ \text{INDEX} & \boxed{e2}\ e \\ \text{RELS} & \left\langle \begin{array}{l} \begin{bmatrix} \textit{generic\_n\_rel} & \\ \text{LBL} & \boxed{h3}\ h \\ \text{ARG0} & \boxed{x4}\ x \end{bmatrix}, \begin{bmatrix} \textit{def\_q\_rel} & \\ \text{LBL} & \boxed{h5}\ h \\ \text{ARG0} & \boxed{x4} \\ \text{RSTR} & \boxed{h7}\ h \\ \text{BODY} & \boxed{h6}\ h \end{bmatrix}, \begin{bmatrix} \textit{\_este\_a\_rel} & \\ \text{LBL} & \boxed{h3} \\ \text{ARG0} & \boxed{e8}\ e \\ \text{ARG1} & \boxed{x4} \end{bmatrix}, \\ \begin{bmatrix} \textit{tam\_rel} & \\ \text{LBL} & \boxed{h9}\ h \\ \text{ARG0} & \boxed{e2} \end{bmatrix}, \begin{bmatrix} \textit{\_mais\_x\_-que-\_rel} & \\ \text{LBL} & \boxed{h9} \\ \text{ARG0} & \boxed{e10}\ e \\ \text{ARG1} & \boxed{e2} \\ \text{ARG2} & \boxed{u11}\ u \end{bmatrix}, \begin{bmatrix} \textit{\_pequeno\_a\_rel} & \\ \text{LBL} & \boxed{h9} \\ \text{ARG0} & \boxed{e2} \\ \text{ARG1} & \boxed{x4} \end{bmatrix} \end{array} \right\rangle \\ \text{HCONS} & \left\langle \begin{bmatrix} \textit{qeq} & \\ \text{HARG} & \boxed{h1} \\ \text{LARG} & \boxed{h9} \end{bmatrix}, \begin{bmatrix} \textit{qeq} & \\ \text{HARG} & \boxed{h7} \\ \text{LARG} & \boxed{h3} \end{bmatrix} \right\rangle \end{bmatrix}$$

Figure 7.21: MRS for the sentences “Isto é menor” and “Isto é mais pequeno” (*This is smaller*).

A distinction between ungradable adjectives and gradable ones is also present in the lexical types for adjectives.

As far as this aspect is concerned, however, LXGram is rather permissive, as many adjectives traditionally deemed to be ungradable can be forced to occur with degree specifiers.

But for illustration purposes, there are types for ungradable adjectives, and the examples below show sentences the grammaticality judgments of which are mirrored in the grammar.

(37) a. Ele é um mero pianista.
he is a mere pianist
He is a mere pianist.

b. * Ele é um muito mero pianista.
he is a very mere pianist

Ungradable adjectives are implemented as having a feature GRADABLE with the value *ungradable* (see Section 7.4).

7.9.1 Postponed Coverage or Known Limitations

A few subcategorization frames of adjectives have been implemented, but LXGram is still missing many possible subcategorization patterns. This work is to be concluded in phases B and C of the implementation.

Chapter 8

Noun Phrases

As far as noun phrases are concerned, LXGram currently implements case distinctions in personal pronouns (described in Section 8.1), some subcategorization frames of nouns (Section 8.2), pragmatic distinctions in second person forms (Section 8.3), non-atomic proper names (Section 8.5), word order constraints between nouns and many of their dependents and NP constituency (Section 8.6) and noun ellipsis (Section 8.7).

8.1 HEAD Types for Nouns

A simplified type hierarchy of *head* values for nouns and pronouns is presented in Figure 8.1.

Common nouns, proper names and personal pronouns are given dedicated HEAD types (*common-noun*, *proper-noun* and *pronoun* respectively). Proper names have a special feature, NAME-SORT, that is not necessary for common nouns and pronouns (see Section 8.5.3 for details).

In the lexical entries of common nouns, proper names and pronouns, only the types *common-noun*, *proper-name* and *pronoun* are used to constrain their HEAD feature.

The head type *noun-or-complementizer* allows a verb like "dizer" (*say*) to select for a complement that is either headed by a complementizer or a noun. The type hierarchy does not make a unifier available for *proper-name* and *noun-or-complementizer*, and therefore proper names are blocked from occurring as the complement of these verbs.

The type *common-or-proper-noun* is a convenient place to put some generalizations: namely common nouns and proper names cannot occur in the contexts where dative pronouns occur, and their CASE feature is constrained to be *non-dative* (see Section 8.1.1).

8.1.1 Case

The feature CASE is a feature of *noun*, a type under *head* that is the value of the feature HEAD of all nominal constituents. It is responsible for constraining the distribution of NPs. The type hierarchy of case values is presented in Figure 8.2.

In Portuguese only personal pronouns overtly manifest case. The leaf types under *case* correspond to the five-fold partition of syntactic contexts induced by these morphological differences. The distinction between *oblique-com* and *oblique-sem* deserves an explanation below.

The remaning types are used to allow underspecification of ambiguous/homonymous forms. Table 8.1 shows the correspondence between case values and some personal pronoun forms.

The type *non-dative* describes the syntactic contexts where NPs headed by a common noun or a proper name can occur: the feature CASE of the head type *common-or-proper-noun* (see the type hierarchy of head types for nouns in Figure 8.1) is constrained to be *non-dative*. Since all nouns (common or proper) have this head type (pronouns have a different one, *pronoun*) no other NP can commute with the dative "lhe" and "lhes" (a PP must be used instead).

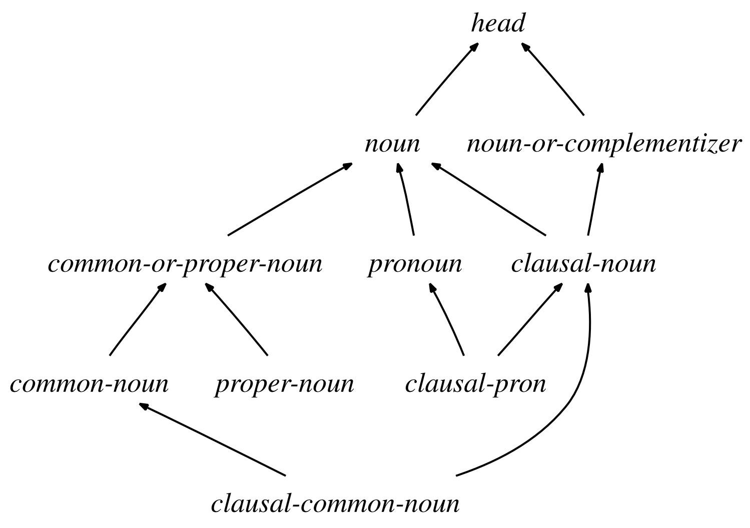


Figure 8.1: Simplified type hierarchy under *noun*.

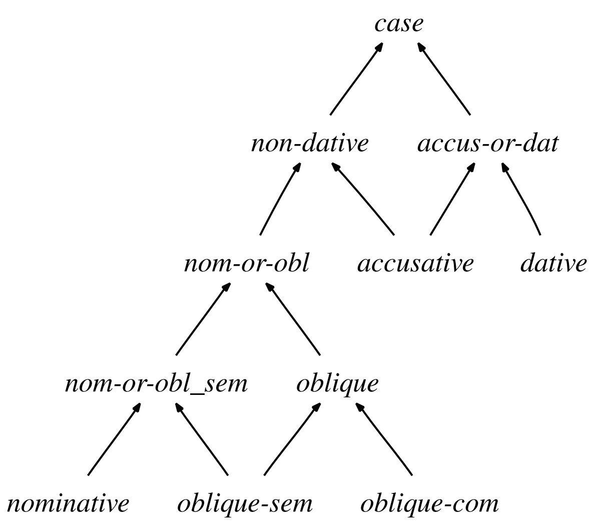


Figure 8.2: Type hierarchy under *case*.

Case	Forms
accusative	o a os as
accus-or-dat	me te vos nos
dative	lhe lhes
nominative	eu tu
nom-or-obl	você vocês
nom-or-obl_sem	nós vós
oblique-sem	mim ti si
oblique-com	migo tigo sigo nosco vosco

Table 8.1: Case values of some personal pronouns.

Four cases are usually recognized for Portuguese: nominative, accusative, dative and oblique. Oblique case is the case of the personal pronoun forms that occur as the complement of prepositions. In LXGram oblique case is divided into *oblique-com* and *oblique-sem*. The former corresponds to the forms "comigo, contigo, consigo, connosco, convosco" (*with me, with you, ...*). The latter case value, *oblique-sem*, is the value for the personal pronoun forms that occur as the complement of other prepositions, among them the preposition "sem" (*without*).

Preprocessor rules are used to split the forms "comigo, contigo, consigo, connosco, convosco" (*with me, with you, ...* into the preposition "com" (*with*) and the forms "migo, ..." (see Table 8.1 for the complete set) and this preposition is constrained to select for NPs with *oblique-com* case. An alternative would be to split these forms into preposition plus the corresponding, standard oblique forms "mim, ti, si, nós, vós", in which case the distinction between *oblique-com* and *oblique-sem* would no longer be necessary (the usual position). This is however not correct, because the forms "vós" and "vosco" have different T-V values (see Section 8.3), and so a distinction must be made between the two, requiring two separate lexical entries with different constraints on case and T-V values.

With this setup, the expected constraints on case can be implemented: finite verbs constrain the CASE feature of their subject to be *nominative* (in LXGram this is implemented in the inflectional rules responsible for subject-verb agreement), copular verbs taking two NP arguments constrain them both to be nominative, transitive verbs constrain the element in their COMPS list to bear a CASE of type *accusative*, and so on.

8.2 Noun Complementation

As far as noun complementation is concerned, LXGram currently implements nouns with the two subcategorization frames:

- Nouns with a PP complement headed by "de"

(38) O Pedro é pai da Maria.
the Pedro is father of the Maria
Pedro is Maria's father.

- Nouns with two PP arguments, one headed by "de" and another headed by "de" or "por" The first one will correspond to the complement of a lexically related verb, the second one to its subject, as the example (39) shows.

(39) a. A destruição da cidade pelo exército foi brutal.
the destruction of the city by the army was brutal

The army's destruction of the city was brutal.

b. O exército destruiu a cidade brutalmente.
the army destroyed the city brutally
The army brutally destroyed the city.

Other selection frames of nouns will be implemented in the subcategorization phase.

8.3 T-V Distinctions

Tu-Vous (T-V) distinctions, like French *tu* vs. *vous* (when addressing a single person), are represented in the MRSs produced by LXGram. This information does not affect truth conditions and is of a pragmatic nature, but we include it in the semantics.

There are two reasons to make T-V distinctions visible in the semantics: (1) it is important to maintain this sort of information in some applications (e.g. machine translation), and (2) it is very difficult to separate it from the semantics (when the semantics includes person information) if one wants to be able to underspecify certain combinations of person and T-V values. For instance the verb form "é" (*is*) can be a third person singular form or a pragmatically marked second person singular form. If one wants to underspecify this information, so that it does not give rise to two different analyses, person and T-V information must be placed together (i.e. in the same feature) and if person is to be visible in the semantics, so will T-V distinctions.

In LXGram the ability to underspecify forms like "é" was chosen over multiplying analyses, with the disadvantage that some pragmatic information will be visible in the produced MRSs.

The implemented level of T-V differences is course-grained. Here are some distinctions currently ignored:

- First person majestic plurals ("Dizemos", lit. *we say* but meaning *I say*). They were used by royalty in the past, and can also be used by authors.

- Second person plural verb forms and corresponding personal pronouns (like "vós") used as singular. This use is archaic. The grammar considers all these forms plural.

- First person NPs headed by nouns (e.g. someone says "o avô" (lit. *the grandfather*) instead of "eu" (*I*) when addressing his grandchildren). They are mostly confined to nouns describing family relations. They present complications in that they go with third person verb forms, but a third person verb form with a null subject is never interpreted as first person.

The following phenomena are accounted for:

- Three way distinction for second person singular forms: "tu" (akin to French *tu* and German *du*), "você" (intermediate) and NPs headed by a common noun, like "o senhor" (lit. *the Mr.*), akin to French *vous* and German *Sie*.

- Lexical marking of whether a noun can have second person readings (along with third person readings), like "pai" (*father*), "senhor" (*Mr.*), "arquitecto" (*architect*), "Pedro" (a person's name), or is confined to third person interpretations, like "pianista" (*pianist*), "homem" (*man*), "piano" (*piano*), "França" (*France*).

- Unavailability of second person readings in indefinite NPs headed by nouns for which those readings are otherwise available: "o pai" (lit. *the father*) can be second person or third person, but "um pai" (*a father*) is third person.

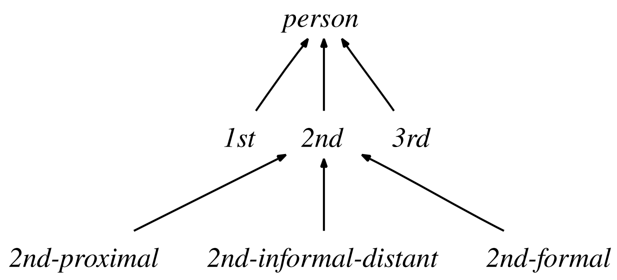


Figure 8.3: Simplified hierarchy for values of person and T-V distinctions.

- Unavailability of second person readings in restrictively modified NPs headed by nouns for which those readings are otherwise available: "o senhor" (lit. *the Mr.*) can be second person or third person, but "o senhor do fato escuro" (*the gentleman in the dark suit*) is third person.

The implementation consists of elaborating the type hierarchy for person values so that it includes both person and T-V categories, and using the same feature PERSON that was previously used only for person to encode T-V distinctions as well. This choice is motivated by the fact that person markings and T-V marking are highly connected in Portuguese: e.g. only second person shows T-V differences at a general level of granularity.

A first approach to encoding a three-way T-V distinction for second person forms would thus posit a type hierarchy for person values like the one in Figure 8.3.

All these types are necessary:

- "Eu" (*I*) or the SUBJ element of verb forms like "sou" (*am*) must be constrained to be PNG|PERSON *1st*.
- "Ele" (*he*) or the SUBJ element of verb forms like "chove" (*rains*) must be constrained to be PNG|PERSON *3rd*.
- "Tu" (*you*, singular) or the SUBJ element of verb forms like "és" (*are*) must be constrained to be PNG|PERSON *2nd-proximal*.
- "Você" (*you*, singular) must be constrained to be PNG|PERSON *2nd-informal-distant*.
- The constraints on the subject NP and the verb form in a sentence like "O senhor venha cá" (*You come here*, lit. *The Mr. come here*) must be such that they unify to *2nd-formal* in this and similar sentences.
- An imperative plural form like "vejam" (*see*) does not carry any T-V information, despite being second person. The type *2nd* is justified, and not merely an organizational device.

Merging T-V information with person information means that T-V distinctions are involved in agreement relations, like subject-verb agreement. This is desired, but it requires elaborations on this type hierarchy, since many verb forms can go with subjects displaying various person and T-V combinations.

Complications have to be made to this hierarchy, because there are several degrees of underspecification that must be dealt with. The final type hierarchy for person and T-V values is presented in Figure 8.4.

The data motivating this hierarchy are:

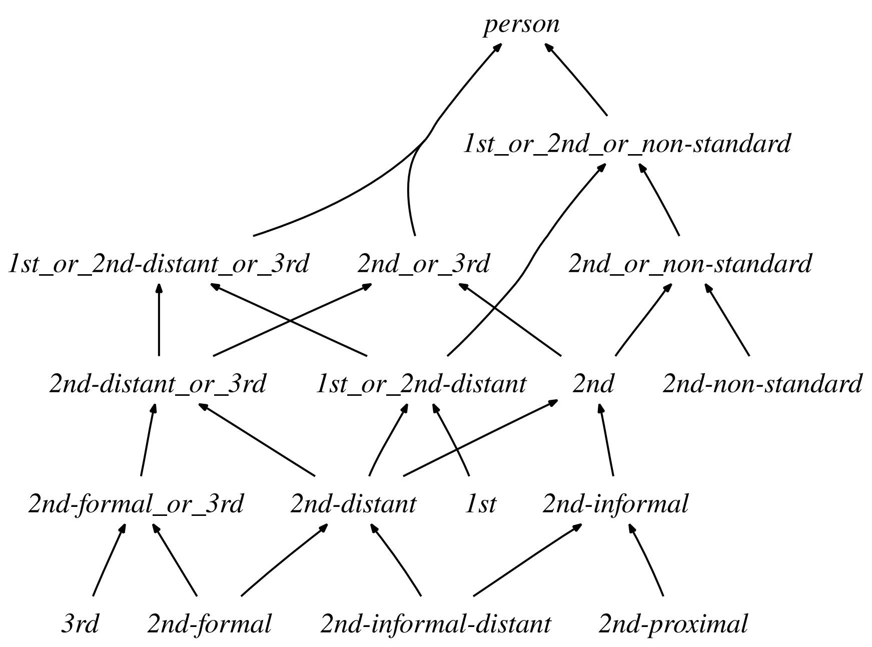


Figure 8.4: Hierarchy for values of person and T-V distinctions.

- The plural item "vocês" corresponds to singular "tu" and "você", so a supertype of *2nd-proximal* and *2nd-informal-distant* is needed: *2nd-informal*.

- An imperative singular form like "veja" (*see*) can be *2nd-formal* or *2nd-informal-distant*, so a supertype *2nd-distant* is needed in order for such verb forms to be underspecified. Non-reflexive "si" is also ambiguous this way.

- The personal pronoun "vós" (*you*, plural) and the verb forms showing subject agreement with "vós" are currently used only dialectally or in contexts where archaic speech is used, like religious ones. Because these verb forms do not take other second person plural subjects (like "vocês" above, or noun headed NPs), an incompatible type must be assigned to them. The type *2nd-non-standard* is used to this end. *2nd-non-standard* and *2nd* are incompatible, as depicted in Figure 8.4, because *2nd* is supposed to mean *real* second person.

- A supertype for *2nd-non-standard* and *2nd* is actually needed, because the forms "vos" (*you*, plural, accusative or dative) and "convosco" (*with you*, plural) correspond to nominative "vocês" (assigned type *2nd-informal*), to plural NPs headed by common nouns with second person interpretations (assigned type *2nd-formal*), and also to nominative "vós" (assigned *2nd-non-standard*). Forms "vos" and "convosco" are hence assigned the value *2nd_or_non-standard*, and considered plural.

- Several noun headed NPs, like "o senhor", are ambiguous between *2nd-formal* readings and *3rd* person readings. A common supertype for these two, *2nd-formal_or_3rd*, is thus required in order to have these NPs underspecified for person.

- Singular verbs forms that take as subjects singular NPs like the ones just mentioned also take "você" as subject. Therefore, they must be even more underspecified. *2nd-distant_or_3rd* is used in these forms and has as direct descendants types *2nd-formal_or_3rd* and *2nd-distant*. These are the verb forms traditionally classified as third person singular.

Personal pronoun	English translation	PERSON value
"eu"	*I*	*1st*
"tu"	*you*	*2nd-proximal*
"você"	*you*	*2nd-informal-distant*
"ele"	*he*	*3rd*
"ela"	*she*	*3rd*
"nós"	*we*	*1st*
"vós"	*you*	*2nd-non-standard*
"vocês"	*you*	*2nd-informal*
"eles"	*they* (masculine)	*3rd*
"elas"	*they* (feminine)	*3rd*
"me"	*me*, *myself*	*1st*
"te"	*you*, *yourself*	*2nd-proximal*
"o"	*you*, *him*	*2nd-distant_or_3rd*
"a"	*you*, *her*	*2nd-distant_or_3rd*
"se"	*yourself*, *yourselves*, *himself*, *herself*, *themselves*	*2nd-distant_or_3rd*
"nos"	*us*, *ourselves*	*1st*
"vos"	*you*, *yourselves*	*2nd_or_non-standard*
"os"	*you*, *them* (masculine)	*2nd_or_3rd*
"as"	*you*, *them* (feminine)	*2nd_or_3rd*
"lhe"	*you*, *him*, *her*	*2nd-distant_or_3rd*
"lhes"	*you*, *them*	*2nd_or_3rd*
"mim"	*me*	*1st*
"ti"	*you*	*2nd-proximal*
"si"	*you*, *yourself*, *yourselves*, *himself*, *herself*, *themselves*	*2nd-distant_or_3rd*
"migo"	*me*	*1st*
"tigo"	*you*	*2nd-proximal*
"sigo"	*you*, *yourself*, *yourselves*, *himself*, *herself*, *themselves*	*2nd-distant_or_3rd*
"nosco"	*us*, *ourselves*	*1st*
"vosco"	*you*, *yourselves*	*2nd_or_non-standard*

Table 8.2: PERSON values of personal pronouns, with T-V information

- Plural verbs forms that take as subjects plural NPs like the ones just mentioned also take "vocês" as subject. Note that whereas "você" is *2nd-informal-distant*, "vocês" is *2nd-informal.* These verb forms are thus even more abstract than their singular counterparts and must be given the type *2nd_or_3rd*, a supertype of *2nd-distant_or_3rd* and *2nd.* These are the verb forms traditionally classified as third person plural.

- In some tenses the traditionally called third person singular forms collide with first person singular forms. Forms of the "Pretérito Imperfeito" are one example ("era", *I, he, she, it was*). The type *1st_or_2nd-distant_or_3rd* is used to constrain their subject.

- Type *1st_or_2nd_or_non-standard* is used grammar internally. Imperative forms are constrained to take a subject with this value for PERSON. This way third person imperatives are blocked.

The lexical entries for personal pronouns have their PERSON feature constrained with the values presented in Table 8.2.

$$\begin{bmatrix} \textit{mrs} & \\ \text{RELS} & \left\langle \begin{bmatrix} \textit{named\_rel} & \\ \text{LBL} & \boxed{h17}\ h \\ \text{ARG0} & \boxed{x13} \\ \text{ARG1} & \boxed{s18}\ s \end{bmatrix}, \begin{bmatrix} \textit{string-equals\_rel} & \\ \text{LBL} & \boxed{h17} \\ \text{ARG0} & \boxed{s18} \\ \text{CARG} & \textit{\'Africa} \end{bmatrix} \right\rangle \end{bmatrix}$$

Figure 8.5: MRS fragment associated with the proper name “África” (*Africa*).

8.4 Personal Pronouns

Personal pronouns are associated with two relations *pronoun_q_rel* and *pronoun_n_rel*. The representation used is employed in several other grammars. We chose to also use it in LXGram, in order to have representations similar to the other grammars.

The relation *pronoun_n_rel* fills the restrictor of the quantifier relation *pronoun_q_rel*, so personal pronouns receive semantics similar to $\lambda P.pronoun\_q(x, pronoun\_n(x), P(x))$.

8.5 Proper Names

8.5.1 Semantics

In LXGram proper names are given a semantics which is slightly different from the one in the LinGO Grammar Matrix and the other DELPH-IN grammars. Instead of a single binary *named_rel* relation between a referential index and a character string, two relations are used: a binary one between a referential index and a string variable, and a second binary relation holding between the same string variable and a string literal. The second relation is intended to mean string equality.

An example of the MRS fragment corresponding to the name “África” (*Africa*) is presented in Figure 8.5. The equivalent expression

$$\lambda x.named(x, s_1) \wedge string\text{-}equals(s_1, \text{"}\acute{A}frica\text{"})$$

is intended to be synonymous to (albeit more verbose than) the standard

$$\lambda x.named(x, \text{"}\acute{A}frica\text{"})$$

with s_1 to be interpreted as existentially quantified and *string-equals* as the equality relation between strings.

The motivation for this is explained in Section 8.5.3.2.

8.5.2 Information in Lexical Entries for Proper Names

Proper names are cross-classified in the following dimensions:

- Gender
- Number morphology;
- Grammatical person;
- Co-occurrence with determiners.

As with common nouns, gender must be specified lexically. There are names with gender variants, like “Francisco”, a masculine proper name, and “Francisca”, a feminine one, and pairs with masculine “-o” and feminine “-a” spellings are recurrent. For these, it would be desirable to produce both forms from a single lexical entry, for reasons of lexical economy, as is done for common nouns. This is currently not the case. The reason is that the CARG feature in the feature structure representing the semantic relation of these nouns should intuitively be identical to their surface form (different for the two names in the pair). To the best of our knowledge, the LKB does not make it possible to derive the value of CARG from the surface form, so elements of pairs like the one presented must be listed in the lexicon. There are thus only three possibilities for proper names regarding gender: masculine (e.g. “Tejo”, *Tagus*, a river), feminine (e.g. “Lisboa”, *Lisbon*, a city) or underspecified (“Castro”, a surname).

The application of morphological rules for number is also controlled in the lexical entries. Some can only be plural (“Estados Unidos”, *United States*; “Canárias”, *Canary Islands*), some have the same form for both numbers (“Egas”, a men’s name; “Íris”, a women’s name; “Borges”, a surname). These do not undergo the lexical rules for inflection in number.

Proper names are classified in the lexicon as giving rise to third person NPs or NPs ambiguous between third person and second person formal (see Section 8.3). The former are constrained with

$$\left[\textsc{synsem}|\textsc{local}|\textsc{cat}|\textsc{head}|\textsc{agr}|\textsc{png}|\textsc{person}\quad \textit{3rd}\right]$$

and the constraint

$$\left[\textsc{synsem}|\textsc{local}|\textsc{cat}|\textsc{head}|\textsc{agr}|\textsc{png}|\textsc{person}\quad \textit{2nd-distant\_or\_3rd}\right]$$

is inherited by the latter.

As far as the co-occurrence of proper names and determiners is concerned, there are three possibilities: (1) a determiner is obligatory (the definite article by default, but may be another), (2) a determiner is optional, and (3) the definite article is not possible. The sentences in (40) illustrate these three patterns.

(40) a. Foi *em/na Grécia.
it was in/in the Greece
It was in Greece.

b. Foi em/na Itália.
it was in/in the Italy
It was in Spain.

c. Foi em/*na Malta.
it was in/in the Malta
It was in Malta.

The way this different behavior is controlled in LXGram is by resorting to the types for quantifier relations. Quantifier relations are visible in the feature structures for all nouns, under the path SYNSEM|LOCAL|CONT|KEYS|QUANT-REL. A noun dependent that introduces a quantifier relation (e.g. a determiner) in the MRS unifies that relation with this feature in their sister node, e.g.:

$$\left[\textsc{synsem}|\textsc{local}\ \left[\begin{array}{l}\textsc{cat}|\textsc{head}|\textsc{marker}|\textsc{select}|\textsc{local}|\textsc{cont}|\textsc{keys}|\textsc{quant-rel}\ \boxed{1}\\ \textsc{cont}|\textsc{rels}\ \left\{\boxed{1}\right\}\end{array}\right]\right]$$

The feature KEYS is unified between the mother and the head daughter in all headed constructions, via a constraint added to the type *headed-phrase*, from which all headed constructions inherit:

$$\begin{bmatrix}\textsc{synsem}|\textsc{local}|\textsc{cont}|\textsc{keys}\ \boxed{1} \\ \textsc{head-dtr}|\textsc{synsem}|\textsc{local}|\textsc{cont}|\textsc{keys}\ \boxed{1}\end{bmatrix}$$

This way, the quantifier relation of an NP is always visible in the feature structure for the noun that is the head of that NP.

We use a type hierarchy of quantifier relations to control the co-occurrence pattern of proper names with determiners. An example follows. The construction for bare NPs introduces a quantifier relation of the type *bare-qrel.* A proper name that must be preceded by a determiner, like "Grécia" (*Greece*) in (40a), comes in the lexicon with the its QUANT-REL constrained to be of type *expressed-or-plural-qrel.*[1]

We organize these types for quantifier relations in a hierarchy in such a way that the unifier of *bare-qrel* and *expressed-or-plural-qrel* is constrained to correspond to a plural noun (via a constraint under ARG0|PNG|NUMBER inside the quantifier relation).

The other co-occurrence restrictions are controlled in a similar way: proper names that cannot occur with definite articles constraint their QUANT-REL feature with a type that is incompatible with the type of the quantifier relation introduced by definite articles.

Proper names that cannot co-occur with definite articles (when there is no other material in the NP) must nevertheless be preceded by a definite article (or other determiner) if they are modified. Compare the following two sentences:

(41) a. Isso aconteceu em/*na Lisboa.
that happened in/in the Lisbon
That happened in Lisbon.

b. Isso aconteceu *em/na Lisboa de 1700.
that happened in/in the Lisbon of 1700
That happened in 18th century Lisbon.

LXGram employs an attribute MODIFICATION|MODIFIERS under SYNSEM|LOCAL|CAT to control this behavior (see Section 5.12). The two main values that this feature takes are *has-modifiers* and *no-modifiers.* Nouns come in the lexicon with the latter value, and Head-Functor constructions produce nodes with the value *has-modifiers* for this feature.

These types also have a QUANT-REL feature, where a quantifier relation is represented. This relation is unified with the quantifier relation of the NP by determiners and bare-NP constructions. The type of the quantifier relation under *no-modifiers* is such that when unified with the quantifier relation for these names, it results in a type that is incompatible with the type of

[1]A name like this one can in fact appear without a determiner if it is plural. Consider the examples with names of letters:

(1) a. * (O) "A" é a primeira letra do alfabeto.
the A is the first letter of the alphabet
A is the first letter of the alphabet.

b. Essa palavra não tem "A"s.
that word not has As
That word has no A in it.

quantifier relation of definite articles. The type of the quantifier relation under *has-modifiers* is such that the result of unifying it with the type of quantifier relation associated to these names results in a type that is incompatible with the type of quantifier relation introduced by the bare NP constructions.

8.5.3 Name-Name Phrases

By name-name phrases we mean expressions that are the simple concatenation of proper names and have the distribution of a proper name. Some examples are “Fernando Pessoa”, “João Sebastião Ribeiro”.

8.5.3.1 Syntax

These constructions can be analysed as head-initial. This is compatible with the observation that features like gender percolate from the leftmost daughter, as examples (42) illustrate (in LXGram, the feature AGR, where agreement is constrained, is a feature of HEAD – see Section 5.10).

(42) a. o João
the.MASCULINE João
João (a men’s name)

b. a Maria
the.FEMININE Maria
Maria (a woman’s name)

c. o João Maria
the.MASCULINE João Maria
João Maria (a men’s name)

d. a Maria João
the.FEMININE Maria João
Maria João (a women’s name)

The implementation makes use of a dedicated syntax rule for this construction, the *name-name* phrase. It is a binary rule that constrains both daughters to have HEAD features of type *proper-noun* and is head-initial, i.e. the HEAD feature of the mother node is unified with the HEAD of the left daughter.

The number of names in this construction has no upper bound, so the rule must be able to iterate. As stated, it allows both left recursion and right recursion. This produces multiple analyses when more than two names are present, e.g. “António Oliveira Salazar” yields [[António Oliveira] Salazar] and [António [Oliveira Salazar]]. For reasons related to semantic composition (see Section 8.5.3.2), LXGram chooses to allow only right recursion by constraining the left daughter to have a SYNSEM of type *lex-synsem* (most phrases in LXGram, including the *name-name* phrase, produce nodes with a SYNSEM of type *phrase-synsem*, which is incompatible with *lex-synsem*).

The interaction with postnominal modification of names also multiplies parses. Consider example (43).

(43) Aquela não era a Maria João que conhecíamos.
that not was the Maria João that we knew
That wasn’t the Maria João that we knew.

No constraints have yet been mentioned to prevent both analyses [[Maria João] [que conhecíamos]] and [Maria [João [que conhecíamos]]]. In LXGram the second analysis is rejected

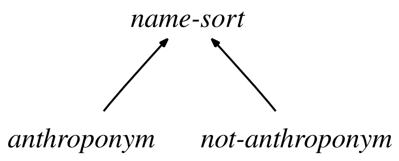


Figure 8.6: Hierarchy for values of feature NAME-SORT.

by a constraint in *name-name* phrases whereby the daughters cannot contain modifiers. More specifically, the left daugher is already constrained to have a *lex-synsem*, so prenominal modifiers are not problematic; the constraint

$$\left[\text{NON-HEAD-DTR}|\text{SYNSEM}|\text{LOCAL}|\text{CAT}|\text{MODIFICATION}|\text{MODIFIERS} \quad \textit{no-modifiers}\right]$$

is included to block modifiers in the right (non-head) daughter (the feature MODIFICATION is explained in Section 5.12).

The nouns that can appear in this construction are person names. Other proper names (e.g. names of countries) do not participate in it. In order to control for this, it is necessary to state in the lexical entries for proper nouns whether they denote people names or not. It turns out that this information is already there, under grammatical person (see Section 8.5.2): if a proper name is marked as allowing for second person readings, it is a person's name; otherwise it is not.

The advantage of blocking non-person names from this construction is to prevent multiple analyses for non-ambiguous sentences. Example (8.5.3.1) shows a sentence that would give rise to an illegitimate parse if non-person names were allowed in this construction (with "Lisboa Kiev" as a single name and a null subject).

(44)	Como	Lisboa	Kiev	também	tem	sete	colinas.
	like	Lisbon	Kiev	also	has	seven	hills

Like Lisbon, Kiev also has seven hills.

It is worth noting that the fact that many place names can also be surnames does not undermine this restriction, because multiple lexical entries are needed in these cases. Multiple lexical entries are necessary, because surnames tend to optionally go with definite articles (see Section 8.5.2) and be underspecified for gender, whereas place names generally have a specific gender value and are more idiosyncratic when it comes to co-occurrence with determiners.

This restriction cannot however be enforced by constraining person values. Since the names that are legitimate here are the ones that are constrained with a PERSON feature with value *2nd-distant_or_3rd* and the ones that are not allowed come in the lexicon with a PERSON that is *3rd*, the only way to control rule application based on this feature would be to say that the daughters of this rule have PERSON constrained to be *2nd*, which is a subtype of *2nd-distant_or_3rd* and has no unifier with *3rd* (see Figure 8.4). But this would constrain the PERSON value of the resulting phrase inappropriately to be *2nd* person (since it is percolated from the head daughter), as the resulting phrase is also underspecified as *2nd-distant_or_3rd.*

Therefore, another mechanism is used to restrict application of this syntactic rule to people's names. A feature NAME-SORT is involved, which is a feature of HEAD appropriate for the type *proper-noun*, a subtype of *head* (see Figure 8.1). NAME-SORT is of type *name-sort*, with the subtypes displayed in Figure 8.6.

Since the PERSON value of proper names must be stated in the lexicon, two supertypes of lexical leaf types can be used to constrain it, *noun-proper-second-or-third-person-item* and *noun-*

$$\left[\begin{array}{ll} \textit{name-name-phrase} & \\ \text{SYNSEM} & \left[\begin{array}{l} \textit{phrase-synsem} \\ \text{LOCAL}|\text{CAT}|\text{HEAD}\ \boxed{1}\left[\begin{array}{ll}\textit{proper-noun} & \\ \text{NAME-SORT} & \textit{anthroponym}\end{array}\right]\end{array}\right] \\ \text{HEAD-DTR} & \boxed{2}\left[\text{SYNSEM}\left[\begin{array}{l}\textit{lex-synsem}\\ \text{LOCAL}|\text{CAT}|\text{HEAD}\ \boxed{1}\end{array}\right]\right] \\ \text{NON-HEAD-DTR} & \boxed{3}\left[\text{SYNSEM}|\text{LOCAL}|\text{CAT}\left[\begin{array}{ll}\text{HEAD} & \left[\begin{array}{ll}\textit{proper-noun} & \\ \text{NAME-SORT} & \textit{anthroponym}\end{array}\right] \\ \text{MODIFICATION}|\text{MODIFIERS} & \textit{no-modifiers}\end{array}\right]\right] \\ \text{ARGS} & \left\langle \boxed{2}, \boxed{3}\right\rangle \end{array}\right]$$

Figure 8.7: Main syntactic constraints on *name-name* phrases.

proper-third-person-item, and simlutaneously the attribute NAME-SORT. In *noun-proper-second-or-third-person-item*, the relevant constraints are

$$\left[\begin{array}{ll} \textit{noun-proper-second-or-third-person-item} & \\ \text{SYNSEM}|\text{LOCAL}|\text{CAT}|\text{HEAD} & \left[\begin{array}{ll}\text{AGR}|\text{PNG}|\text{PERSON} & \textit{2nd-distant\_or\_3rd} \\ \text{NAME-SORT} & \textit{anthroponym}\end{array}\right]\end{array}\right]$$

and they are

$$\left[\begin{array}{ll} \textit{noun-proper-third-person-item} & \\ \text{SYNSEM}|\text{LOCAL}|\text{CAT}|\text{HEAD} & \left[\begin{array}{ll}\text{AGR}|\text{PNG}|\text{PERSON} & \textit{3rd} \\ \text{NAME-SORT} & \textit{not-anthroponym}\end{array}\right]\end{array}\right]$$

in *noun-proper-third-person-item.* The lexical types for proper names that allow second person readings inherit from *noun-proper-second-or-third-person-item*, and the ones that do not allow them inherit from *noun-proper-third-person-item.* The *name-name* phrase constrains both daughters to have a NAME-SORT of type *anthroponym.*

The key syntactic properties of the *name-name* phrase can thus be presented in Figure 8.7.

8.5.3.2 Semantics

If the piece of semantics a grammar associates to the proper name "João" is

$$\lambda x.named(x, \text{“}João\text{”})$$

and the one it associates to "Maria" is

$$\lambda x.named(x, \text{“}Maria\text{”})$$

then

$$\lambda x.named(x, \text{“}João\text{”}) \land named(x, \text{“}Maria\text{”})$$
$$\land\ name\text{-}precedes(x, \text{“}João\text{”}, \text{“}Maria\text{”})$$

is an intuitive representation for the combination “João Maria”. The *name-precedes* relation is necessary because it does not follow from the fact that someone has the name “João” and the name “Maria” that their name is “João Maria” (it can be “Maria João”). The first argument of that *name-precedes* relation is necessary because the order can be different in different people’s names.

However, the LKB does not seem to allow a relation to have two arguments that are Lisp strings. The problem arises at generation (when the relations are indexed for generation, string literal arguments are also indexed, but apparently only the first one is considered). An ‘’invalid predicates” error message for the *name-precedes* relation appears and generation is aborted.

A fix to this problem is to use string variables. Instead of giving the expression “João Maria” the semantics presented above and repeated here

$$\lambda x.named(x, \text{“}João\text{”}) \wedge named(x, \text{“}Maria\text{”}) \\ \wedge\ name\text{-}precedes(x, \text{“}João\text{”}, \text{“}Maria\text{”})$$

we can produce

$$\lambda x.\exists s_1.\exists s_2.named(x, s_1) \wedge string\text{-}equals(s_1, \text{“}João\text{”}) \\ \wedge\ named(x, s_2) \wedge string\text{-}equals(s_2, \text{“}Maria\text{”}) \\ \wedge\ name\text{-}precedes(x, s_1, s_2)$$

instead. No relation now has two arguments that are Lisp strings, and the error is avoided.

If this semantics is to be produced in *name-name* phrases compositionally from the semantics of proper names, the latter must be changed. A proper name like “Maria” cannot come in the lexicon with the semantics

$$\lambda x.named(x, \text{“}Maria\text{”})$$

but rather with the much more verbose

$$\lambda x.\exists s_1.named(x, s_1) \wedge string\text{-}equals(s_1, \text{“}Maria\text{”})$$

as addressed in Section 8.5.1. This is because the second argument of the *named* relation cannot be changed from a string literal to a variable, since composition of semantics must be monotonic.

The actual implementation is as follows. First, the existential quantifiers in the formulas above are not explicitly included, but it is rather assumed that all string variables are existentially quantified.

Two subtypes of *string* are created: *string-literal* and *string-variable*. The latter also inherits from *individual*, because in the LinGO Grammar Matrix the ARG0 feature of relations is declared to be of this type, and some of the relations involved are implemented with an ARG0 of type *string-variable*.

In the LKB configuration file `globals.lsp` the parameter ***string-type*** is set to the new *string-literal*, effectively telling the LKB that TDL string literals (corresponding to Lisp strings) are accepted in all syntactic contexts where objects of type *string-literal* are accepted. In `mrsglobals.lisp` a condition is added to ***determine-variable-type*** associating *string-variable* instances to “s” (this makes *string-variable* instances be represented as s_n in MRSs, where n is an integer).

The MRS for the *name-name* phrase “Maria João” is presented in Figure 8.8. The *name-precedes* relation is contributed by the syntactic rule.

In Section 8.5.3.1 it was mentioned that only right recursive structures are accepted for semantic reasons. The explanation is that the ARG1 of the *name-precedes* relation comes from the left daughter and the ARG2 of the same relation comes from the right daughter (which in turn

$$\left[\begin{array}{ll} \textit{mrs} & \\ \text{RELS} & \left\langle \left[\begin{array}{lll} \multicolumn{3}{l}{\textit{name-precedes\_rel}} \\ \text{LBL} & \boxed{h11} & h \\ \text{ARG0} & \boxed{x5} & \\ \text{ARG1} & \boxed{s13} & s \\ \text{ARG2} & \boxed{s12} & s \end{array}\right], \left[\begin{array}{ll} \multicolumn{2}{l}{\textit{named\_rel}} \\ \text{LBL} & \boxed{h11} \\ \text{ARG0} & \boxed{x5} \\ \text{ARG1} & \boxed{s13} \end{array}\right], \left[\begin{array}{ll} \multicolumn{2}{l}{\textit{string-equals\_rel}} \\ \text{LBL} & \boxed{h11} \\ \text{ARG0} & \boxed{s13} \\ \text{CARG} & \textit{Maria} \end{array}\right], \left[\begin{array}{ll} \multicolumn{2}{l}{\textit{named\_rel}} \\ \text{LBL} & \boxed{h11} \\ \text{ARG0} & \boxed{x5} \\ \text{ARG1} & \boxed{s12} \end{array}\right], \left[\begin{array}{ll} \multicolumn{2}{l}{\textit{string-equals\_rel}} \\ \text{LBL} & \boxed{h11} \\ \text{ARG0} & \boxed{s12} \\ \text{CARG} & \textit{João} \end{array}\right] \right\rangle \end{array}\right]$$

Figure 8.8: MRS fragment associated with the name-name phrase "Maria João"

comes from its own left daughter). For the phrase "João Sebastião Ribeiro", the right recursive binary structure [João [Sebastião Ribeiro]] produces a semantic representation that fully describes the ordering of the named involved. On the other hand, the left recursive structure [[João Sebastião] Ribeiro] would produce a semantic representation that only says that "João" precedes "Sebastião" and "João" precedes "Ribeiro", but does not say whether "Sebastião" precedes "Ribeiro" or "Ribeiro" precedes "Sebastião" (i.e. [[João Ribeiro] Sebastião] would be associated to equivalent semantics).

8.6 NP Structure

The table in Table 8.3 presents some of the data covered here.

The category Predeterminers (Position I) in this table contains elements like "todo" (*all*).

The category Determiners (Position II) includes the definite and indefinite articles, the demonstratives and other items, like "bastante(s)" (*much*, *several*).

The category Possessives (Position III) contains prenominal possessives, which in European Portuguese are preceded by determiners.

The category Cardinals (in Position IV) includes the cardinal numerals, either atomic ("dois", *two*) or complex ("vinte e dois", *twenty two*).

The category Ordinals (in Position IV) includes the ordinal numerals, atomic ("primeiro", *first*) and complex ones ("vigésimo primeiro", *twenty-first*).

The category Vague Quantifiers (in Position IV) contains elements like "muitos" (*many*), "poucos" (*few*). The distinction between determiners like "bastantes" and vague quantifiers is not semantic but syntactic. Consider their different behavior with respect to a preceding definite article, exemplified in the following sentences:

(45) a. [$_{\text{NP}}$ Muitas espécies de sapos da Amazónia] já estão extintas.
many species of frogs of the Amazon Rainforest already are extinct
Many species of frogs of the Amazon Rainforest are already extinct.

b. [$_{\text{NP}}$ Bastantes espécies de sapos da Amazónia] já estão
several species of frogs of the Amazon Rainforest already are
extintas.
extinct
Several species of frogs of the Amazon Rainforest are already extinct.

c. [$_{\text{NP}}$ As muitas espécies de sapos da Amazónia] já estão
the many species of frogs of the Amazon Rainforest already are
extintas.
extinct
The many species of frogs of the Amazon Rainforest are already extinct.

	I	II	III	IV	V	VI	VII	VIII	IX
(a)		a	minha	primeira		bicicleta		com pedais amarelos	
		the	*my*	*first*		*bicycle*		*with yellow pedals*	
	`my first bicycle with yellow pedals`								
(b)	todas	aquelas		três mil		pessoas		ali	
	all	*those*		*three thousand*		*people*		*over there*	
	`all those three thousand people over there`								
(c)		um		certo	grande	espírito			que criou o mundo
		a		*certain*	*great*	*spirit*			*that created the world*
	`a certain great spirit that created the world`								
(d)		a				invasão	americana	do Iraque	
		the				*invasion*	*American*	*of the Iraq*	
	`the American invasion of Iraq`								
(e)				quatro		colegas		teus	
				four		*collegues*		*of yours*	
	`four collegues of yours`								
(f)		a				pesca	baleeira	intensa	
		the				*fishing*	*whale-like*	*intense*	
	`the intense whale fishing`								
(g)		aquelas	suas	muitas		queixas			
		those	*their*	*many*		*complaints*			
	`those many complaints of theirs`								
(h)		o				papa		esse	que é tão snob
		the				*pope*		*that*	*who is such a snob*
	`that pope who is such a snob`								

Table 8.3: NP constituents. Positions within the Noun Phrase: I — Predeterminers; II — Determiners; III — Prenominal Possessives; IV — Cardinals (b) (e), Ordinals (a), Vague Quantifiers (g), Markers of Indefinite Specifics (c); V — Prenominal Adjective Phrases; VI — Head Noun; VII — Adjectival Arguments; VIII — Adjective Phrase Adjuncts (f), Prepositional Phrase Arguments (d), Prepositional Phrase Adjuncts (a), Adverbial Phrase Adjuncts (b), Postnominal Possessives (e), Postnominal Demonstratives (h); IX — Restrictive Relative Clauses.

d. * [$_{\text{NP}}$

As	bastantes	espécies	de	sapos	da	Amazónia	] já	estão
the	several	species	of	frogs	of the	Amazon Rainforest	already	are

extintas.
extinct

The category Indefinite Specifics (in Position IV) contains elements like "certo" and "determinado" (*certain*), that mark NPs with exclusively indefinite specific readings, as in the first example below (46a):

(46) a.

Todas	as	pessoas	leram	um	certo	livro.
all	the	people	have read	a	certain	book

All people have read a certain book.

$\exists y[book(y) \wedge \forall x[person(x) \rightarrow read(x, y)]]$

b.

Todas	as	pessoas	leram	um	livro.
all	the	people	have read	a	book

All people have read a book.

$\forall x[person(x) \rightarrow \exists y[book(y) \wedge read(x, y)]]$

$\exists y[book(y) \wedge \forall x[person(x) \rightarrow read(x, y)]]$

The most interesting property of these elements is that their contribution to the meaning of the sentences where they occur consists in merely restricting the relative scope possibilities between the quantifiers in these sentences. The example Portuguese sentence in (46b) is ambiguous between the two readings shown below it. In contrast, the example sentence in (46a) is not ambiguous and only has the reading where the existential quantifier has wide scope — its specific reading. This issue is explored in Section 8.6.4.1.

The category Prenominal Adjective Phrases (Position V) includes adjective phrases (APs) that precede the noun, and the slot named Head Noun (Position VI) represents the position where the noun surfaces.

The slot for Adjectival Arguments (Position VII) represents the position where adjectives that realize arguments of nouns surface. In the example in the table repeated below, the adjective form "americana" (*American*) realizes one of the arguments of the noun "invasão" (*invasion*). The semantics of this NP is quite similar to the semantics of a sentence like *The U.S. invaded Iraq.* More specifically, the arguments of the semantic relations for the noun "invasão"/*invasion* and the verb *invade* are the same in these examples.

(47)

a	invasão	americana	do	Iraque
the	invasion	American	of the	Iraq

the American invasion of Iraq

In Position VIII one finds APs that do not saturate noun arguments, prepositional phrase (PP) adjuncts (not realizing noun arguments) and complements (realizing noun arguments), adverbial phrase (AdvP) adjuncts of nouns, postnominal demonstratives and postnominal possessives (adjuncts or complements). Not all adverbs can occur in this context (as noun modifiers). Among the adverbial phrases that can modify nouns one finds "aqui", "aí", "ali", "dentro (de NP)", "fora (de NP)", "junto (a/de NP)" respectively *here*, *there*, *there*, *inside (NP)*, *outside/out of NP*, *nearby/near NP*.

The last slot is for relative clauses (Position IX).

Elements occupying the same position in Table 8.3 generally show free word order among themselves (but, depending on the category of these elements, there are some restrictions that will be presented in the following Sections). For instance the relative word order between cardinals and ordinals (both in Position IV) is unconstrained:

(48) a. Os primeiros dois filmes foram cancelados.
the first two films were canceled
The first two films were canceled.

b. Os dois primeiros filmes foram cancelados.
the two first films were canceled
The two first films were canceled.

The numbering of these positions reflects precedence constraints among these elements: to give an example, prenominal adjectives (Position V) cannot precede cardinals (Position IV):

(49) a. Os adeptos entusiasmaram-se depois de [$_{\mathrm{NP}}$ duas grandes vitórias do
the fans got excited after two great victories of the
clube.]
club
The fans got excited after two great victories of their club.

b. * Os adeptos entusiasmaram-se depois de [$_{\mathrm{NP}}$ grandes duas vitórias do
the fans got excited after great two victories of the
clube.]
club

8.6.1 General Constraints

The type hierarchy in Figure 8.9 presents the values of the features MARKING and MARK that are used to account for the NP structure of Portuguese. The Head-Functor configurations presented in Section 5.3 are used to implements many of the NP constituents. This hierarchy controls their syntactic distribution, as explained in the following sections.

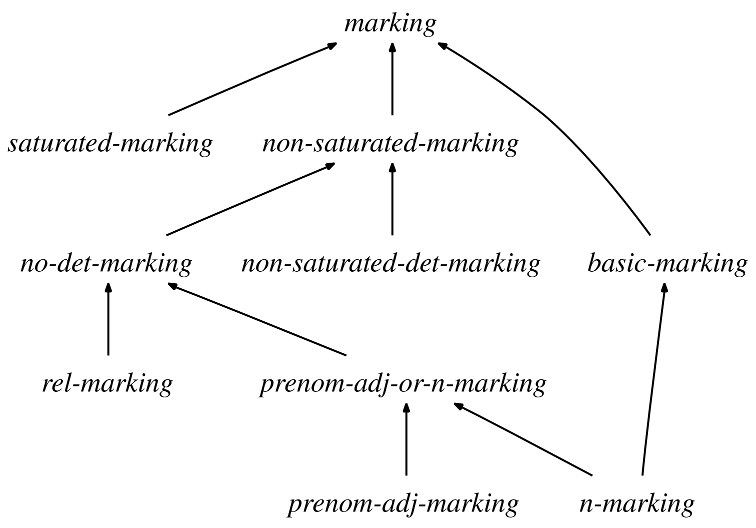


Figure 8.9: Simplified hierarchy for the values of the features MARKING and MARK in noun headed constituents.

Elements that select NPs select for a constituent with a feature MARKING of the type *saturated*. Nouns come in the lexicon with the type *n-marking* for this feature. Since these two types are incompatible according to this hierarchy, a noun needs to combine with other elements in order to form an NP.

The type *marking* is defined to have a subfeature MK-VAL, and that feature has the subfeatures DEMONSTRATIVE, QUALQUER, TAL, POSSESSIVE, OUTRO, CARDINAL, INDEF-SPEC and ORDINAL. These features are used to encode the presence of elements like possessives, cardinals, etc., and to control their iterability and co-occurrence. The feature DEMONSTRATIVE is for demonstratives, QUALQUER is for the element "qualquer" (*any*), TAL is for the prenominal "tal" (*such*), POSSESSIVE is for possessives, OUTRO is for the element "outro" (*other*), CARDINAL is for cardinals, ORDINAL is for ordinals, INDEF-SPEC is for prenominal "certo" and "determinado" (*certain*), and ORDINAL is for ordinals. They will also be explained in the next sections. The type hierarchy with the values that all these features can take is in Figure 8.10.

$$\begin{tikzcd}
 & \textit{present-or-absent} & \\
\textit{absent-or-prehead-present} \arrow[ur] & \textit{absent-or-posthead-present} \arrow[u] & \textit{present} \arrow[ul] \\
\textit{absent} \arrow[u] \arrow[ur] & \textit{prehead-present} \arrow[ul] \arrow[ur] & \textit{posthead-present} \arrow[ul] \arrow[u]
\end{tikzcd}$$

Figure 8.10: Type hierarchy under *present-or-absent*.

In this hierarchy, the type *absent* denotes the fact that the relevant element is not present in a given phrase. For instance, a constituent with this value for the feature POSSESSIVE is a constituent where no possessive occurs. The value *present* denotes the presence of the relevant element, and its two subtypes *prehead-present* and *posthead-present* state whether the relevant element precedes or follows the head noun respectively. Types like *absent-or-prehead-present* and *absent-or-posthead-present* are used to control co-occurrence restrictions between these elements.

The marking type *n-marking* is constrained in the following manner:

$$\left[\begin{array}{ll}
\multicolumn{2}{l}{\textit{n-marking}} \\
\text{MK-VAL} & \left[\begin{array}{ll}
\text{TAL} & \textit{absent} \\
\text{POSSESSIVE} & \textit{absent-or-posthead-present} \\
\text{CARDINAL} & \textit{absent} \\
\text{ORDINAL} & \textit{absent} \\
\text{INDEF-SPEC} & \textit{absent}
\end{array}\right]
\end{array}\right]$$

This type, *n-marking*, is the value of the feature MARKING of nouns. It is also the value of all noun headed constituents made up by a noun and postnominal material. Possessives, for instance, can follow the noun, so the value of the feature POSSESSIVE is *absent-or-posthead-present*. It will be *absent* for nouns (all the features under MK-VAL are constrained to be *absent* in the lexical entries of nouns), but it will take the value *posthead-present* if a postnominal possessive is present in a constituent with MARKING of the type *n-marking*.

Other elements that attach to noun-headed constituents (and whose presence is not represented with these attributes) simply unify the MK-VAL feature of the constituent that they select with the MK-VAL under their MARK feature. The example of adjectives illustrate this point They have the following constraints:

$$\left[\text{SYNSEM}|\text{LOCAL}|\text{CAT}|\text{HEAD}|\text{MARKER} \left[\begin{array}{l}
\text{MARK}|\text{MK-VAL } \boxed{1} \\
\text{SELECT}|\text{LOCAL}|\text{CAT}|\text{MARKING}|\text{MK-VAL } \boxed{1}
\end{array}\right]\right]$$

Elements that select NPs also constrain them to have discharged COMPS, besides requiring them to have a MARKING feature compatible with the type *saturated-marking*. However, in LXGram this is not achieved by constraining the COMPS feature to be an empty list. Instead, this feature is constrained to be of a type similar to the type *olist* that comes in the LinGO Grammar Matrix. The name of this type is *list-of-optional-synsems* in LXGram. See Section 7.2.

8.6.2 Determiners and Predeterminers

Saturated NPs can be introduced by a determiner or a predeterminer:

(50) a. [$_{\text{NP}}$ Os$_{\text{D}}$ seres humanos] são livres.
the human beings are free

Human beings are free.

b. [$_{\text{NP}}$ Todos$_{\text{PreD}}$ os$_{\text{D}}$ seres humanos] são livres.
all the human beings are free

All human beings are free.

In the first case the quantifier relation of the NP comes from the determiner, but in the second case it comes from the predeterminer. When a predeterminer introduces an NP, a determiner must be present (51).

(51) a. todas as pessoas
all the people

all (the) people

b. todas aquelas pessoas
all those people

all those people

c. *EP/BP todas pessoas
all people

The last example is actually a possible NP in Brazilian Portuguese. More on this is said below.

Determiners that co-occur with predeterminers must thus be different from determiners introducing an NP, since the former contribute no quantifier semantics but the latter do. Multiple lexical items are required in view of the fact that it is not possible to underspecify the number of elementary predications that a given lexical item contributes to the MRS representation.

The lexical entries for determiners that contribute quantifier semantics and appear at the left edge of NPs (the form "os" in (50a)) are constrained in the following manner:

$$\left[\begin{array}{ll}
\text{SYNSEM}|\text{LOCAL} & \left[\begin{array}{ll}
\text{CAT}|\text{HEAD}|\text{MARKER} & \left[\begin{array}{ll}
\multicolumn{2}{l}{\textit{pre-only-marker}} \\
\text{SELECT}|\text{LOCAL} & \left[\begin{array}{ll}
\text{CAT} & \left[\begin{array}{ll}\text{HEAD} & \textit{noun}\\ \text{MARKING} & \textit{no-det-marking}\end{array}\right]\\
\text{CONT}|\text{HOOK} & \left[\begin{array}{ll}\text{LTOP} & \boxed{1}\\ \text{INDEX} & \boxed{2}\end{array}\right]
\end{array}\right]\\
\text{MARK} & \textit{saturated-marking}
\end{array}\right]\\
\text{CONT} & \left[\begin{array}{ll}
\text{HOOK}|\text{LTOP} & \boxed{3}\\
\text{RELS} & \left\{\left[\begin{array}{ll}\text{LBL} & \boxed{3}\\ \text{ARG0} & \boxed{2}\\ \text{RSTR} & \boxed{4}\end{array}\right]\right\}\\
\text{HCONS} & \left\{\left[\begin{array}{ll}\multicolumn{2}{l}{\textit{qeq}}\\ \text{HARG} & \boxed{4}\\ \text{LARG} & \boxed{1}\end{array}\right]\right\}
\end{array}\right]
\end{array}\right]
\end{array}\right]$$

The relevant constraints on the determiners that follow predeterminers and contribute no semantics (the form “os” in (50b)) are:

$$\left[\begin{array}{ll}
\text{SYNSEM}|\text{LOCAL} & \left[\begin{array}{ll}
\text{CAT}|\text{HEAD}|\text{MARKER} & \left[\begin{array}{ll}
\multicolumn{2}{l}{\textit{pre-only-marker}} \\
\text{SELECT}|\text{LOCAL} & \left[\begin{array}{ll}
\text{CAT} & \left[\begin{array}{ll}\text{HEAD} & \textit{noun}\\ \text{MARKING} & \textit{no-det-marking}\end{array}\right]\\
\text{CONT}|\text{HOOK} & \boxed{1}
\end{array}\right]\\
\text{MARK} & \textit{non-saturated-det-marking}
\end{array}\right]\\
\text{CONT} & \left[\begin{array}{ll}
\text{HOOK} & \boxed{1}\\
\text{RELS} & \{\}\\
\text{HCONS} & \{\}
\end{array}\right]
\end{array}\right]
\end{array}\right]$$

They have the HOOK of their sister node (so that the LTOP of the mother node in Head-Functor rules, which comes from the functor daughter, is the same as the head daughter’s LTOP).

Predeterminers have constraints like:

$$\left[\begin{array}{ll}
\textsc{synsem}|\textsc{local} & \left[\begin{array}{ll}
\textsc{cat}|\textsc{head}|\textsc{marker} & \left[\begin{array}{ll}
\multicolumn{2}{l}{\textit{pre-only-marker}} \\
\textsc{select}|\textsc{local} & \left[\begin{array}{ll}
\textsc{cat}|\textsc{head} & \textit{noun} \\
\textsc{cont}|\textsc{hook} & \left[\begin{array}{ll} \textsc{ltop} & \boxed{1} \\ \textsc{index} & \boxed{2} \end{array}\right]
\end{array}\right] \\
\textsc{prehead} & \left[\begin{array}{l}
\textsc{select}|\textsc{local}|\textsc{cat}|\textsc{marking}\ \textit{non-saturated-det-marking} \\
\textsc{mark}\ \textit{saturated-marking}
\end{array}\right]
\end{array}\right] \\
\textsc{cont} & \left[\begin{array}{ll}
\textsc{hook}|\textsc{ltop} & \boxed{3} \\
\textsc{rels} & \left\{\left[\begin{array}{ll} \textsc{lbl} & \boxed{3} \\ \textsc{arg0} & \boxed{2} \\ \textsc{rstr} & \boxed{4} \end{array}\right]\right\} \\
\textsc{hcons} & \left\{\left[\begin{array}{ll} \multicolumn{2}{l}{\textit{qeq}} \\ \textsc{harg} & \boxed{4} \\ \textsc{larg} & \boxed{1} \end{array}\right]\right\}
\end{array}\right]
\end{array}\right]
\end{array}\right]$$

They require the presence of a semantically vacuous determiner (therefore they select for a sister node with MARKING of type *non-saturated-det-marking*). They produce a saturated phrase, since their feature MARK is of type *saturated-marking.* They also introduce quantifier semantics.

Predeterminers can also appear postnominally, as in (52). As the second example shows, they occupy an NP internal position. This means that syntactic and semantic scope do not match in such structures, and more features are therefore needed to pass the relevant information along the syntax tree. We will not elaborate on this issue, as this is left to future work (LXGram includes a preliminary treatment for postnominal universal quantifiers that, however, makes them outscope the rest of the NP and cannot account for the data in (52b)).

(52) a. as pessoas todas
the people all
all (the) people

b. as pessoas todas dessa aldeia
the people all from that village
all (the) people from that village

To account for Brazilian Portuguese "todo" (51c), we resort to positing more than one lexical entries for "todo". The constraints associated with the HEAD attribute of this item only differ from the ones of the head type of predeterminers above in that, instead of selecting for a constituent with MARKING *non-saturated-det-marking*, this item selects for an element with MARKING *no-det-marking* (i.e. this item is encoded as a determiner).

8.6.3 Prenominal Possessives

In definite NPs, possessives can appear prenominally (53a), while postnominal possessives can occur in indefinite NPs (53b). However, demonstratives license both prenominal and postnominal possessives (54).

(53) a. A minha bicicleta tem um pneu furado.
the my bicycle has a tire flat
My bicycle has a flat tire.

b. Uma bicicleta minha tem um pneu furado.
a bicycle my/mine has a tire flat

A bicycle of mine has a flat tire.

(54) a. Aquela tua bicicleta tem um pneu furado.
that your bicycle has a tire flat

That bicycle of yours has a flat tire.

b. Aquela bicicleta tua tem um pneu furado.
that bicycle your/yours has a tire flat

That bicycle of yours has a flat tire.

Other contexts allow prenominal possessives. Examples are vocatives (55a) and predicative nominals lacking a determiner (55b).

(55) a. Minha senhora, eu quero a mala.
my lady, I want the bag

I'd like the bag, Miss.

b. É teu irmão?
is your brother

Is he your brother? (a brother of yours)

Postnominal possessives are covered in Section 8.6.11.2, as well as the mechanism to control relative word order between noun and possessive.

Prenominal possessives always occur after the determiner (article, demonstrative, ...), if it is present, and they always precede cardinals, if both occur (56).

(56) a. as minhas duas bicicletas
the my two bicycles

my two bicycles

b. * minhas as duas bicicletas
my the two bicycles

c. * as duas minhas bicicletas
the two my bicycles

The HEAD of possessives is thus constrained in the following way:

$$\left[\begin{array}{ll}
\multicolumn{2}{l}{\textit{possessive}} \\
\textsc{marker} & \left[\begin{array}{ll}
\textsc{select}|\textsc{local}|\textsc{cat} & \left[\begin{array}{ll}
\textsc{head} & \textit{noun} \\
\textsc{marking} & \left[\begin{array}{ll}
\textsc{mk-val} & \left[\begin{array}{ll}
\textsc{possessive} & \textit{absent} \\
\textsc{tal} & \boxed{1} \\
\textsc{cardinal} & \boxed{2} \\
\textsc{ordinal} & \boxed{3} \\
\textsc{indef-spec} & \boxed{4} \\
\textsc{outro} & \boxed{5} \\
\textsc{qualquer} & \boxed{6} \\
\textsc{demonstrative} & \boxed{7}
\end{array}\right]
\end{array}\right]
\end{array}\right] \\
\textsc{mark} & \left[\begin{array}{ll}
\textsc{mk-val} & \left[\begin{array}{ll}
\textsc{possessive} & \textit{present} \\
\textsc{tal} & \boxed{1}\ \textit{present-or-absent} \\
\textsc{cardinal} & \boxed{2}\ \textit{present-or-absent} \\
\textsc{ordinal} & \boxed{3}\ \textit{present-or-absent} \\
\textsc{indef-spec} & \boxed{4}\ \textit{present-or-absent} \\
\textsc{outro} & \boxed{5}\ \textit{present-or-absent} \\
\textsc{qualquer} & \boxed{6}\ \textit{present-or-absent} \\
\textsc{demonstrative} & \boxed{7}\ \textit{absent-or-posthead-present}
\end{array}\right]
\end{array}\right] \\
\textsc{prehead} & \left[\begin{array}{ll}
\textsc{select}|\textsc{local}|\textsc{cat}|\textsc{marking} & \textit{no-det-marking} \\
\textsc{mark} & \left[\begin{array}{ll}
\multicolumn{2}{l}{\textit{no-det-marking}} \\
\textsc{mk-val}|\textsc{possessive} & \textit{prehead-present}
\end{array}\right]
\end{array}\right]
\end{array}\right]
\end{array}\right]$$

This accounts for prenominal possessives after a definite article or demonstrative determiner.

In Brazilian Portuguese, possessives can introduce an NP. In the corresponding phrases in European Portuguese the definite article must precede the possessive. A Brazilian example is in (57).

(57) Minha bicicleta tem um pneu furado.
my bicycle has a tire flat
My bicycle has a flat tire.

Since quantifier semantics is generally introduced in the predeterminer or determiner slots (Position I and Position II in Table 8.3), the Brazilian possessives must have different lexical items from the possessives occurring with determiners, because these do not carry quantifier semantics, but the former must do so (see the next section for semantic representations of possessives). Also, they will present different constraints related to MARKING. More specifically, the feature MARK bears the value *saturated*.

Also note that NPs introduced by a possessive, like the one in (57), do not have readings characteristic of bare NPs — but bare NPs headed by a singular count noun are actually possible in Brazilian Portuguese ([Munn and Schmitt, 1998] and [Müller, 2002]) —, so these NPs should not be considered to be bare NPs.

Prenominal possessives following a definite article or other determiner are also attested in Brazilian Portuguese.

This analysis also covers sequences made up by a predeterminer "todo" (Section 8.6.2) followed immediately by a possessive, which is a possibility in Brazilian Portuguese. These sequences are derived by the lexical entry for "todo" that is specific to Brazilian Portuguese and a lexical entry for a possessive that is available to both varieties.

8.6.3.1 Possessives as Arguments of Nouns

Possessives can realize arguments of noun relations, which in Portuguese are in the unmarked case realized by postnominal material (58a). Consider (58b).

$$\left[\begin{array}{ll}
\textit{mrs} & \\
\text{LTOP} & \boxed{h1}\ h \\
\text{INDEX} & \boxed{x2}\ x \\
\text{RELS} & \left\langle \begin{array}{l}
\left[\begin{array}{ll} \textit{\_o\_q\_rel} & \\ \text{LBL} & \boxed{h1} \\ \text{ARG0} & \boxed{x2} \\ \text{RSTR} & \boxed{h4}\ h \\ \text{BODY} & \boxed{h3}\ h \end{array}\right],
\left[\begin{array}{ll} \textit{possessive\_a\_rel} & \\ \text{LBL} & \boxed{h5}\ h \\ \text{ARG0} & \boxed{e6}\ e \\ \text{ARG1} & \boxed{x2} \\ \text{ARG2} & \boxed{x7}\left[\begin{array}{ll} x & \\ \text{PNG.PERSON} & \textit{3rd} \end{array}\right] \end{array}\right], \\
\left[\begin{array}{ll} \textit{pronoun\_q\_rel} & \\ \text{LBL} & \boxed{h8}\ h \\ \text{ARG0} & \boxed{x7} \\ \text{RSTR} & \boxed{h9}\ h \\ \text{BODY} & \boxed{h10}\ h \end{array}\right],
\left[\begin{array}{ll} \textit{pronoun\_n\_rel} & \\ \text{LBL} & \boxed{h11}\ h \\ \text{ARG0} & \boxed{x7} \end{array}\right],
\left[\begin{array}{ll} \textit{\_cavalo\_n\_rel} & \\ \text{LBL} & \boxed{h5} \\ \text{ARG0} & \boxed{x2} \end{array}\right]
\end{array}\right\rangle \\
\text{HCONS} & \left\langle \left[\begin{array}{ll} \textit{qeq} & \\ \text{HARG} & \boxed{h4} \\ \text{LARG} & \boxed{h5} \end{array}\right], \left[\begin{array}{ll} \textit{qeq} & \\ \text{HARG} & \boxed{h9} \\ \text{LARG} & \boxed{h11} \end{array}\right] \right\rangle
\end{array}\right]$$

Figure 8.11: MRS fragment corresponding to the NP "o seu cavalo" (*his/her/their horse*).

(58) a. o irmão da Ana
the brother of the Ana
Ana's brother

b. o seu irmão
the her brother
her brother

In both examples, "irmão" denotes a two-place predicate. In (58a) the second argument is realized by the PP "de Ana", and in (58b) it surfaces as "seu".

Possessives are implemented in LXGram as carrying personal pronoun semantics (see Section 8.4).

When possessives do not fill a noun argument, an extra relation is included between the index of the personal pronoun and that of the head noun, called *possessive_a_rel*. An example is in Figure 8.11.

When possessives realize noun arguments, this relation is not present in the MRS. Instead, the index of the personal pronoun occurs as the second argument of the relation corresponding to the head noun. Figure 8.12 contains an MRS example of argumental possessives, for the NP "o seu irmão" (*his/her/their brother*).

Because the number of elementary predications contributed to an MRS by these two sorts of elements (argumental vs. modifying possessives) is different, multiple lexical entries are required for possessives.

Argumental possessives and modifying possessives have the same syntactic distribution, though. This creates problems for the treatment of argumental possessives, since we assume that noun complements are saturated at a much lower level (see Section 8.6.8).

$$\left[\begin{array}{ll}
\multicolumn{2}{l}{\textit{mrs}} \\
\text{LTOP} & \boxed{h1}\ h \\
\text{INDEX} & \boxed{x2}\ x \\
\text{RELS} & \left\langle \left[\begin{array}{ll} \multicolumn{2}{l}{\textit{\_o\_q\_rel}} \\ \text{LBL} & \boxed{h1} \\ \text{ARG0} & \boxed{x2} \\ \text{RSTR} & \boxed{h4}\ h \\ \text{BODY} & \boxed{h3}\ h \end{array}\right], \left[\begin{array}{ll} \multicolumn{2}{l}{\textit{pronoun\_q\_rel}} \\ \text{LBL} & \boxed{h5}\ h \\ \text{ARG0} & \boxed{x6}\left[\begin{array}{ll} \multicolumn{2}{l}{x} \\ \text{PNG.PERSON} & \textit{3rd} \end{array}\right] \\ \text{RSTR} & \boxed{h7}\ h \\ \text{BODY} & \boxed{h8}\ h \end{array}\right], \left[\begin{array}{ll} \multicolumn{2}{l}{\textit{pronoun\_n\_rel}} \\ \text{LBL} & \boxed{h9}\ h \\ \text{ARG0} & \boxed{x6} \end{array}\right], \left[\begin{array}{ll} \multicolumn{2}{l}{\textit{\_irmão\_n\_-de-\_rel}} \\ \text{LBL} & \boxed{h10}\ h \\ \text{ARG0} & \boxed{x2} \\ \text{ARG1} & \boxed{x6} \end{array}\right] \right\rangle \\
\text{HCONS} & \left\langle \left[\begin{array}{ll} \multicolumn{2}{l}{\textit{qeq}} \\ \text{HARG} & \boxed{h4} \\ \text{LARG} & \boxed{h10} \end{array}\right], \left[\begin{array}{ll} \multicolumn{2}{l}{\textit{qeq}} \\ \text{HARG} & \boxed{h7} \\ \text{LARG} & \boxed{h9} \end{array}\right] \right\rangle
\end{array}\right]$$

Figure 8.12: MRS fragment corresponding to the NP “o seu irmão” (*his/her/their brother*).

The first question to ask is whether projections of prenominal argumental possessives should be produced by some Head-Complement construction or by the *functor-head-phrase* discussed above.

The motivation for considering argumental possessives to be complements is that they are in complementary distribution with PP complements (59a). The motivation for considering them functors is that they are also in complementary distribution with modifying possessives (59b).

(59) a. * o seu irmão da Ana
the her brother of the Ana

b. * o seu seu irmão
the her her brother

If they are treated as functors, then they are unusual in saturating an argument of the head they select.

If they are complements, then prenominal argumental possessives are unusual in preceding the head (in Portuguese this only occurs with clitics and fronted constituents).

In LXGram they are implemented as functors. This choice has the advantage of not requiring more syntactic machinery, but it results in untypical feature structures because, since argumental possessives are considered functors, they cannot discharge an element from the COMPS list of their head, in spite of realizing it themselves.

Since they can see the entire SYNSEM of their sister node via the SELECT attributes, they can unify the index of the personal pronoun relations they introduce with the index of an element in the COMPS attribute of the nominal projection they select for. This produces the right semantics, namely semantic representations exactly like the ones produced by Head-Complement constructions.

They place the same constraints on the values of *marking* as their modifier counterparts. Their non-iterability is in this way immediately predicted.

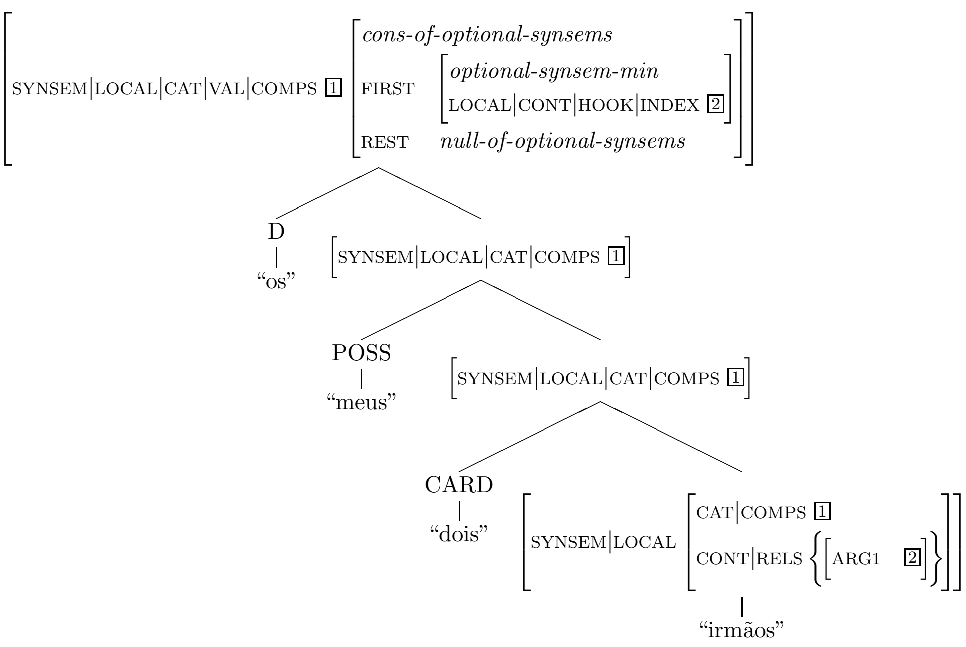
It is important to mention that what enables a non-empty COMPS to appear high enough in a tree in order to be visible by possessives in general is the choice of using the type *list-of-optional-synsems* instead of *null* to constrain the COMPS of NPs, as explained in Section 8.6.1. Consider the following example:

(60)	[NP os	[meus	[dois irmãos	]]]
	the	my	two brothers	

my two brothers

All bracketed phrases in this example have the same value for the feature COMPS according to the implementation. If NPs were constrained to have an empty COMPS, a unary rule would be needed to discharge the unexpressed complement of the noun (this rule could simply pass up the tail of the COMPS of its daughter). It would make sense to have this rule apply in the most embedded position (before the cardinal attaches), for several reasons: it is where Head-Complement constructions occur; some of the constraints common to unary and binary Head-Complement constructions could be factored out in a single supertype; discharging all complements in the same position is also less error-prone and makes the grammar easier to understand, to extend and to debug if needed. In this scenario, the sister node of the possessive would also have an empty COMPS.

By using the type *list-of-optional-synsems* to constrain the COMPS of NPs instead (Section 8.6.1), unrealized complements are visible at the point where possessives attach. For instance, the NP in (60) receives the following simplified analysis:



Relational nouns, like "irmão" (*brother*) above, unify the SARG (see Section 5.11) of their complement with the ARG1 of the relation they introduce, so that the entry for "irmão" contains these constraints, among others:

$$\left[\text{SYNSEM|LOCAL}\ \left[\begin{array}{l}\text{CAT|VAL|COMPS}\ \left\langle\left[\text{LOCAL|CONT|HOOK|SARG}\ \boxed{1}\right]\right\rangle\\ \text{CONT|RELS}\ \left\{\left[\text{ARG1}\ \boxed{1}\right]\right\}\end{array}\right]\right]$$

Constraints in the lexical types for argumental possessives are used to unify the index associated with the personal pronoun relations that they introduce with the index of the first element in the head's COMPS:

$$\left[\textsc{synsem}|\textsc{local}\left[\begin{array}{l}\textsc{cat}|\textsc{head}|\textsc{marker}|\textsc{select}|\textsc{local}|\textsc{cat}|\textsc{val}|\textsc{comps}|\textsc{first}|\textsc{local}|\textsc{cont}|\textsc{hook}|\textsc{sarg}\ \boxed{1}\\ \textsc{cont}|\textsc{rels}\left\{\left[\textsc{arg0}\ \ \boxed{1}\right],\left[\textsc{arg0}\ \ \boxed{1}\right]\right\}\end{array}\right]\right]$$

Crucially, the possessive cannot simply unify the entire SYNSEM of the head's complement with its own SYNSEM, for a number of reasons: (1) a cyclic structure would result, a situation that is not allowed by the systems used; (2) the noun selects for a PP, but a possessive is not a PP — the HEAD feature is different, for instance, and would not unify —; and (3) the first element of COMPS, which in examples like (60) is reduced to type *list-of-optional-synsems*, is an *optional-synsem-min*, but the SYNSEM of the possessive ends up as a *canonical-synsem-min*, since it is realized, and these synsem types are incompatible (see Section 8.6.1).

The fact that the complement of a noun with a synsem of type *optional-synsem-min* (the type of the SYNSEM of unexpressed elements) is actually realized makes this analysis rather uninteresting.

Since possessives can only realize PP complements of nouns (and not CPs for instance), argumental possessives must constrain the nominal projection they attach to to have a COMPS whose first element is a PP headed by a non-predicational preposition (see Section 7.8):

$$\left[\textsc{synsem}|\textsc{local}|\textsc{cat}|\textsc{head}|\textsc{marker}|\textsc{select}|\textsc{local}|\textsc{cat}|\textsc{val}|\textsc{comps}|\textsc{first}|\textsc{local}|\textsc{cat}|\textsc{head}\ \textit{particle-np}\right]$$

This constraint, like the constraint above to fix the semantics, is extremely non-local and against the spirit of HPSG. Furthermore, it means that nouns do not necessarily have visibility over the entire SYNSEM of their complement: if nouns constrain it to be a PP, a possessive can detect this and realize it instead, but a possessive can have constrains on its SYNSEM drastically different from the constraints on the noun's complement. A consequence is that if constraints on noun complements must be added in the future to cover additional phenomena, it may be the case that the definitions for argumental possessives require modifications as well — the analysis is not extensible.

This is the analysis implemented in LXGram currently. An interesting alternative to the analysis of prenominal possessives would be to treat them as elements extracted from a postnominal position. An analysis could be envisaged in a way similar to the treatment of long-distance dependencies, but possibly resorting to other features, so as to not interact with the analysis of unbounded dependencies. This would explain the paradox of arguments realized by possessives surfacing on the left of their head, and, under the assumption of a parallelism between sentence structure and NP structure, it would provide the NP counterpart for the left periphery of sentences.

8.6.4 Cardinals, Ordinals, Vague Quantifiers, Markers of Indefinite Specific NPs

Position IV can be filled in by cardinals, ordinals or markers of indefinite specific NPs, like "certo" or "determinado" (*certain*).

They can co-occur with each other in almost any order (61), the exception being that ordinals cannot precede markers of indefinite specifics, as in (61d).

(61) a.

os	primeiros	dois	capítulos
the	first	two	chapters

the first two chapters

b.

os	dois	primeiros	capítulos
the	two	first	chapters

the first two chapters

c.		um	certo	primeiro	capítulo
		a	certain	first	chapter
		a certain first chapter			
d.	*	um	primeiro	certo	capítulo
		a	first	certain	chapter
		a certain first chapter			

"Certo" is limited to indefinite NPs. Cardinals cannot co-occur with indefinite determiners,[2] so to test the word order possibilities between cardinals and "certo", we have to look at NPs that begin with a cardinal or "certo", as in (62). Such NPs are covered in Section 8.6.5, but (62) already shows that word order between cardinals and markers of indefinite specifics is in general also unconstrained.

(62)	a.	dois	certos	capítulos
		two	certain	chapters
		two certain chapters		
	b.	certos	dois	capítulos
		certain	two	chapters
		two certain chapters		

At most one item of each class can be present (63). They are not repeatable even when an item of a different sort intervenes (64).

(63)	a.	*	Os	dois	três	carros	avariaram.	
			the	two	three	cars	broke down	
	b.	*	O	primeiro	segundo	lugar	está	ocupado.
			the	first	second	seat	is	taken
	c.	*	Um	determinado	certo	carro	avariou.	
			a	certain	certain	car	broke down	

(64)	a.	*	Os	dois	primeiros	três	lugares	estão	ocupados.
			the	two	first	three	places	are	taken
	b.	*	Os	primeiros	dois	segundos	pratos	estão	atrasados.
			the	first	two	second	dishes	are	late
	c.	*	Certos	dois	certos	carros	avariaram.		
			certain	two	certain	cars	broke down		

A class of prenominals, "vague quantifiers" or "quantificational adjectives" (65), has the exact distribution of cardinals .

(65)	Os	vários	participantes	passeiam	as	folhas	pela	sala.
	the	various	participants	walk	the	paper sheets	through	the room
	The various participants walk the paper sheets through the room.							

They cannot co-occur with cardinals (66).

[2]NPs like *some three cars* can be analyzed as involving an item *some* that is not a determiner but rather a modifier of the cardinal, since *some three* roughly means *around three.* This also applies to Portuguese expressions like "alguns três", "uns três", with the same meaning.

(66) a. * os vários vinte participantes
the various twenty participants

b. * os vinte vários participantes
the twenty various participants

Vague quantifiers occur with ordinals (67).

(67) a. os vários primeiros lugares
the various first seats
the various first seats

b. os primeiros vários lugares
the first various seats
the various first seats

They cannot iterate (68).

(68) a. * os vários vários participantes
the various various participants

b. * os vários vinte vários participantes
the various vinte various participants

Vague quantifiers can thus be constrained exactly like cardinals. In the following discussion we will thus ignore them and only talk about cardinals. In LXGram they are implemented essentially as cardinals, with little differences.

Similarly, the class of ordinals can also be considered to include other elements with the same syntactic distribution. This is the case of items like "último" (*last*) and "próximo" (*next*). Consider:

(69) a. * os próximos primeiros capítulos
the next first chapters

b. * os primeiros próximos capítulos
the first next chapters

c. os três próximos capítulos
the three next chapters
the next three chapters

d. os próximos três capítulos
the next three chapters
the next three chapters

We will also have these elements in mind when we discuss ordinals, from now on. In LXGram these items receive the same lexical type as ordinals.

We now turn to the discussion of implementing these three classes: ordinals, cardinals/vague quantifiers and markers of indefinite specific NPs.

Cardinals have the following constraints under their HEAD:

$$
\left[\begin{array}{ll}
\textit{cardinal} & \\
\textsc{marker} & \left[\begin{array}{ll}
\textit{pre-only-marker} & \\
\textsc{mark} & \left[\begin{array}{ll}
\textit{no-det-marking} & \\
\textsc{mk-val} & \left[\begin{array}{ll}
\textsc{cardinal} & \textit{prehead-present}\\
\textsc{demonstrative} & \boxed{1}\\
\textsc{possessive} & \boxed{2}\\
\textsc{qualquer} & \boxed{3}\\
\textsc{tal} & \boxed{4}\\
\textsc{outro} & \boxed{5}\\
\textsc{indef-spec} & \boxed{6}\\
\textsc{ordinal} & \boxed{7}
\end{array}\right]
\end{array}\right]\\
\textsc{select}|\textsc{local}|\textsc{cat} & \left[\begin{array}{ll}
\textsc{head}\ \textit{noun} & \\
\textsc{marking} & \left[\begin{array}{ll}
\textit{no-det-marking} & \\
\textsc{mk-val} & \left[\begin{array}{ll}
\textsc{cardinal} & \textit{absent}\\
\textsc{demonstrative} & \boxed{1}\ \textit{absent-or-posthead-present}\\
\textsc{possessive} & \boxed{2}\ \textit{absent-or-posthead-present}\\
\textsc{qualquer} & \boxed{3}\ \textit{present-or-absent}\\
\textsc{tal} & \boxed{4}\ \textit{present-or-absent}\\
\textsc{outro} & \boxed{5}\ \textit{present-or-absent}\\
\textsc{indef-spec} & \boxed{6}\ \textit{present-or-absent}\\
\textsc{ordinal} & \boxed{7}\ \textit{present-or-absent}
\end{array}\right]
\end{array}\right]
\end{array}\right]
\end{array}\right]
\end{array}\right]
$$

The HEAD of ordinals is:

$$
\left[\begin{array}{ll}
\textit{ordinal} & \\
\textsc{marker} & \left[\begin{array}{ll}
\textit{pre-only-marker} & \\
\textsc{mark} & \left[\begin{array}{ll}
\textit{no-det-marking} & \\
\textsc{mk-val} & \left[\begin{array}{ll}
\textsc{cardinal} & \boxed{1}\\
\textsc{demonstrative} & \boxed{2}\\
\textsc{possessive} & \boxed{3}\\
\textsc{qualquer} & \boxed{4}\\
\textsc{tal} & \boxed{5}\\
\textsc{outro} & \boxed{6}\\
\textsc{indef-spec} & \boxed{7}\\
\textsc{ordinal} & \textit{prehead-present}
\end{array}\right]
\end{array}\right]\\
\textsc{select}|\textsc{local}|\textsc{cat} & \left[\begin{array}{ll}
\textsc{head}\ \textit{noun} & \\
\textsc{marking} & \left[\begin{array}{ll}
\textit{no-det-marking} & \\
\textsc{mk-val} & \left[\begin{array}{ll}
\textsc{cardinal} & \boxed{1}\ \textit{present-or-absent}\\
\textsc{demonstrative} & \boxed{2}\ \textit{absent-or-posthead-present}\\
\textsc{possessive} & \boxed{3}\ \textit{absent-or-posthead-present}\\
\textsc{qualquer} & \boxed{4}\ \textit{present-or-absent}\\
\textsc{tal} & \boxed{5}\ \textit{present-or-absent}\\
\textsc{outro} & \boxed{6}\ \textit{present-or-absent}\\
\textsc{indef-spec} & \boxed{7}\ \textit{absent}\\
\textsc{ordinal} & \textit{absent}
\end{array}\right]
\end{array}\right]
\end{array}\right]
\end{array}\right]
\end{array}\right]
$$

The constraint on the feature INDEF-SPEC to be *absent* is to prevent ordinals from preceding "certo" and "determinado", thus blocking examples like (61d).

The HEAD of markers of indefinite specifics is very similar:

$$\left[\begin{array}{ll}
\multicolumn{2}{l}{\textit{indef-specific}} \\
\text{MARKER} & \left[\begin{array}{ll}
\multicolumn{2}{l}{\textit{pre-only-marker}} \\
\text{MARK} & \left[\begin{array}{ll}
\multicolumn{2}{l}{\textit{no-det-marking}} \\
\text{MK-VAL} & \left[\begin{array}{ll}
\text{CARDINAL} & \boxed{1} \\
\text{DEMONSTRATIVE} & \boxed{2} \\
\text{POSSESSIVE} & \boxed{3} \\
\text{QUALQUER} & \boxed{4} \\
\text{TAL} & \boxed{5} \\
\text{OUTRO} & \boxed{6} \\
\text{INDEF-SPEC} & \textit{present} \\
\text{ORDINAL} & \boxed{7}
\end{array}\right]
\end{array}\right] \\
\text{SELECT}|\text{LOCAL}|\text{CAT} & \left[\begin{array}{ll}
\multicolumn{2}{l}{\text{HEAD } \textit{noun}} \\
\text{MARKING} & \left[\begin{array}{ll}
\multicolumn{2}{l}{\textit{no-det-marking}} \\
\text{MK-VAL} & \left[\begin{array}{ll}
\text{CARDINAL} & \boxed{1}\ \textit{present-or-absent} \\
\text{DEMONSTRATIVE} & \boxed{2}\ \textit{absent} \\
\text{POSSESSIVE} & \boxed{3}\ \textit{absent-or-posthead-present} \\
\text{QUALQUER} & \boxed{4}\ \textit{present-or-absent} \\
\text{TAL} & \boxed{5}\ \textit{absent} \\
\text{OUTRO} & \boxed{6}\ \textit{present-or-absent} \\
\text{INDEF-SPEC} & \textit{absent} \\
\text{ORDINAL} & \boxed{7}\ \textit{present-or-absent}
\end{array}\right]
\end{array}\right]
\end{array}\right]
\end{array}\right]
\end{array}\right]$$

The constraint on the feature DEMONSTRATIVE is because markers of indefinite specific NPs cannot co-occur with demonstratives in the same NP. Since prenominal demonstratives outscope markers of indefinite specifics, prenominal demonstratives also select for a sister node with the value *absent* for the feature INDEF-SPEC.

8.6.4.1 Semantics of Markers of Indefinite Specifics

Prenominal items like "certo" and "determinado" (*certain*) as in the examples (46), repeated below, carry no semantic relations but instead simply restrict the set of available readings:

(70) a.

Todas	as	pessoas	leram	um	certo	livro.
all	the	people	have read	a	certain	book

All people have read a certain book.
$\exists y[book(y) \wedge \forall x[person(x) \rightarrow read(x,y)]]$

b.

Todas	as	pessoas	leram	um	livro.
all	the	people	have read	a	book

All people have read a book.
$\forall x[person(x) \rightarrow \exists y[book(y) \wedge read(x,y)]]$
$\exists y[book(y) \wedge \forall x[person(x) \rightarrow read(x,y)]]$

The MRS assigned by LXGram to the sentence in (70a) is in Figure 8.13, and the MRS for the sentence in (70b) is in Figure 8.14.

The two MRSs have exactly the same relations and handle constraints. The only differences lie in the values of the features SCOPE in some of the handles (of type *h*) in these MRSs. The handles for which this feature is not displayed have it completely unconstrained (minimal types for handles are used to hide unconstrained SCOPE features).

Without these constraints for the SCOPE feature, these two MRSs can be scope resolved in the two following formulas:

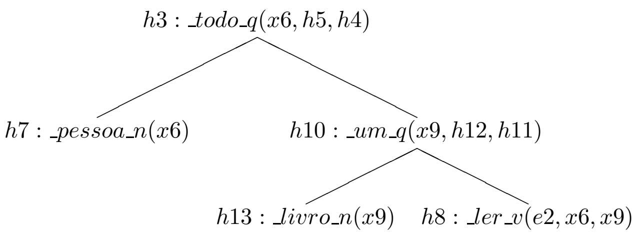
$$\begin{bmatrix} \textit{mrs} & \\ \text{LTOP} & \boxed{h1}\ h \\ \text{INDEX} & \boxed{e2}\ e \\ \text{RELS} & \left\langle \begin{bmatrix} \textit{\_todo\_q\_rel} & \\ \text{LBL} & \boxed{h3}\ h \\ \text{ARG0} & \boxed{x6}\ x \\ \text{RSTR} & \boxed{h5}\begin{bmatrix} h & \\ \text{SCOPE} & \textit{narrow} \end{bmatrix} \\ \text{BODY} & \boxed{h4}\begin{bmatrix} h & \\ \text{SCOPE} & \textit{narrow} \end{bmatrix} \end{bmatrix}, \begin{bmatrix} \textit{\_pessoa\_n\_rel} & \\ \text{LBL} & \boxed{h7}\ h \\ \text{ARG0} & \boxed{x6} \end{bmatrix}, \begin{bmatrix} \textit{tam\_rel} & \\ \text{LBL} & \boxed{h8}\ h \\ \text{ARG0} & \boxed{e2} \end{bmatrix}, \begin{bmatrix} \textit{\_ler\_v\_rel} & \\ \text{LBL} & \boxed{h8} \\ \text{ARG0} & \boxed{e2} \\ \text{ARG1} & \boxed{x6} \\ \text{ARG2} & \boxed{x9}\ x \end{bmatrix}, \begin{bmatrix} \textit{\_um\_q\_rel} & \\ \text{LBL} & \boxed{h10}\begin{bmatrix} h & \\ \text{SCOPE} & \textit{wide} \end{bmatrix} \\ \text{ARG0} & \boxed{x9} \\ \text{RSTR} & \boxed{h12}\begin{bmatrix} h & \\ \text{SCOPE} & \textit{non-widest} \end{bmatrix} \\ \text{BODY} & \boxed{h11}\begin{bmatrix} h & \\ \text{SCOPE} & \textit{non-widest} \end{bmatrix} \end{bmatrix}, \begin{bmatrix} \textit{\_livro\_n\_rel} & \\ \text{LBL} & \boxed{h13}\ h \\ \text{ARG0} & \boxed{x9} \end{bmatrix} \right\rangle \\ \text{HCONS} & \left\langle \begin{bmatrix} \textit{qeq} & \\ \text{HARG} & \boxed{h1} \\ \text{LARG} & \boxed{h8} \end{bmatrix}, \begin{bmatrix} \textit{qeq} & \\ \text{HARG} & \boxed{h5} \\ \text{LARG} & \boxed{h7} \end{bmatrix}, \begin{bmatrix} \textit{qeq} & \\ \text{HARG} & \boxed{h12} \\ \text{LARG} & \boxed{h13} \end{bmatrix} \right\rangle \end{bmatrix}$$

Figure 8.13: MRS with constrained quantifier scope. The sentence is "Todas as pessoas leram um certo livro" (*all people have read a certain book*).

- $\_um\_q(x9, \_livro\_n(x9), \_todo\_q(x6, \_pessoa\_n(x6), \_ler\_v(e2, x6, x9)))$
- $\_todo\_q(x6, \_pessoa\_n(x6), \_um\_q(x9, \_livro\_n(x9), \_ler\_v(e2, x6, x9)))$

These are in fact the two readings for the sentence in (70b). The constraints on SCOPE are intended to block the second reading for the example (70a), in Figure 8.13.

The idea is that, in the second reading, the handle tagged with $\boxed{h4}$ and the handle tagged with $\boxed{h10}$ in these MRSs correspond to the same node in the syntax tree for the scoped formula:



Since they represent the same node, we can assume that they must be compatible. The approach is then to make the constraints on these two handles incompatible in the MRS for the example (70a), but compatible in the MRS for the sentence in (70b).

The type hierarchy for the values that the feature SCOPE can take is in Figure 8.15.

The MRS in Figure 8.13 (for the example in (70a)) has $\boxed{h4}$ with its feature SCOPE with the value *narrow*, and the SCOPE feature of the handle $\boxed{h10}$ has the value *wide*. These types are incompatible according to the hierarchy in Figure 8.15.

$$\begin{bmatrix} \textit{mrs} & \\ \text{LTOP} & \boxed{h1}\ h \\ \text{INDEX} & \boxed{e2}\ e \\ \text{RELS} & \left\langle \begin{bmatrix}\textit{\_todo\_q\_rel} & \\ \text{LBL} & \boxed{h3}\ h \\ \text{ARG0} & \boxed{x6}\ x \\ \text{RSTR} & \boxed{h5}\begin{bmatrix} h & \\ \text{SCOPE} & \textit{narrow}\end{bmatrix} \\ \text{BODY} & \boxed{h4}\begin{bmatrix} h & \\ \text{SCOPE} & \textit{narrow}\end{bmatrix}\end{bmatrix}, \begin{bmatrix}\textit{\_pessoa\_n\_rel} & \\ \text{LBL} & \boxed{h7}\ h\\ \text{ARG0} & \boxed{x6}\end{bmatrix}, \begin{bmatrix}\textit{tam\_rel} & \\ \text{LBL} & \boxed{h8}\ h\\ \text{ARG0} & \boxed{e2}\end{bmatrix}, \begin{bmatrix}\textit{\_ler\_v\_rel} & \\ \text{LBL} & \boxed{h8}\\ \text{ARG0} & \boxed{e2}\\ \text{ARG1} & \boxed{x6}\\ \text{ARG2} & \boxed{x9}\ x\end{bmatrix}, \begin{bmatrix}\textit{\_um\_q\_rel} & \\ \text{LBL} & \boxed{h10}\ h\\ \text{ARG0} & \boxed{x9}\\ \text{RSTR} & \boxed{h12}\begin{bmatrix} h & \\ \text{SCOPE} & \textit{non-widest}\end{bmatrix}\\ \text{BODY} & \boxed{h11}\begin{bmatrix} h & \\ \text{SCOPE} & \textit{non-widest}\end{bmatrix}\end{bmatrix}, \begin{bmatrix}\textit{\_livro\_n\_rel} & \\ \text{LBL} & \boxed{h13}\ h\\ \text{ARG0} & \boxed{x9}\end{bmatrix}\right\rangle \\ \text{HCONS} & \left\langle \begin{bmatrix}\textit{qeq} & \\ \text{HARG} & \boxed{h1}\\ \text{LARG} & \boxed{h8}\end{bmatrix}, \begin{bmatrix}\textit{qeq} & \\ \text{HARG} & \boxed{h5}\\ \text{LARG} & \boxed{h7}\end{bmatrix}, \begin{bmatrix}\textit{qeq} & \\ \text{HARG} & \boxed{h12}\\ \text{LARG} & \boxed{h13}\end{bmatrix}\right\rangle \end{bmatrix}$$

Figure 8.14: MRS allowing for quantifier scope ambiguity. The sentence is "Todas as pessoas leram um livro" (*all people have read a book*).

This mechanism does not work in practice, because the LKB scope resolution algorithm does not perform unification operations on the handles that end up denoting the same node in the fully scoped formulas. For this reason, these constraints on handles are still experimental in LXGram. It would of course be possible to resolve MRSs with an external component, that could take this information into account.

Markers of indefinite specifics contribute no semantics. Instead they simply constrain the features SCOPE of the associated quantifier relation. In LXGram, the quantifier relation of an NP is accessible in all noun headed phrases in the feature QUANT-REL under a feature KEYS (for key relations).[3]. Markers of indefinite specific NPs simply have the constraint:

$$\begin{bmatrix}\text{SYNSEM}|\text{LOCAL}|\text{CAT}|\text{HEAD}|\text{MARKER}|\text{SELECT}|\text{LOCAL}|\text{CONT}|\text{KEYS}|\text{QUANT-REL}|\text{LBL}|\text{SCOPE}\ \ \textit{wide}\end{bmatrix}$$

If a marker of indefinite specifics is not present, this feature will simply have a more general type (allowing for more scope resolution possibilities, as desired).

The type *widest* in Figure 8.15 is used to constrain the LBL of the *proper_q_rel*, which is used with proper names. It is meant to ensure that proper names receive widest scope. The value *wide* is given to the LBL of quantifier relations associated with an NP where a marker of indefinite specifics is present. The RSTR and BODY features of all quantifier relations are constrained with the SCOPE value *non-widest*, except in *proper_q_rel* relation, where they are not constrained. Quantifier relations of determiners and predeterminers that cannot occur with markers of indefinite specifics in the same NP (e.g. "todo" — *all*) have the SCOPE under these

[3]The items that introduce quantifier relations simply unify this relation with the value of their head's QUANT-REL feature. The feature KEYS is unified between the mother node and the head daughter in all headed constructions.

$$\begin{tikzcd}
 & \textit{scope} & & \\
\textit{widest} \arrow[ur] & & \textit{non-widest} \arrow[ul] & \\
 & & \textit{wide} \arrow[u] & \textit{narrow} \arrow[ul]
\end{tikzcd}$$

Figure 8.15: Type hierarchy under *scope*

two features (RSTR and BODY) further constrained to be *narrow*. This set of items contains the predeterminer "todo" (*all*), the definite articles and the demonstratives:

(71) a.

* (Todos)	os	determinados	homens	leram	um	livro.
all	the	certain	men	have read	a	book

b.

* Esses	determinados	homens	leram	um	livro.
those	certain	men	have read	a	book

The scope ordering is thus the following: *widest* > *wide* > *narrow*.
Consider the following example:

(72)

Todos	os	filhos	da	Ana	leram	um	certo	livro.
all	the	children	of the	Ana	have read	a	certain	book

All of Ana's children have read a certain book.

These constraints license only one quantifier scope possibility:

$$proper\_q(x8, named(x, \text{``}Ana''), \_um\_q(x16, \_livro\_n(x16), \_todo\_q(x4, \_filho\_n\_-de-(x4, x8), \_ler\_v(e2, x4, x16))))$$

The *proper_q_rel* relation cannot be embedded under any of the other quantifier relations, because its LBL has SCOPE of the type *widest*, but the handle arguments of the other quantifier relations have the value *non-widest* or *narrow* for their SCOPE features, and any of these types is incompatible with the type *widest*. The *_um_q_rel* relation in this example also cannot be under the scope of the *_todo_q_rel*, like in the previous example.

8.6.5 Cardinals and Markers of Indefinite Specifics as Determiners

As was mentioned in Section 8.6.4, cardinals and items like "certo" and "determinado" (*certain*) that mark indefinite specific NPs can themselves introduce an NP. Some examples from (62) are repeated below in (73).

(73) a.

dois	certos	capítulos
two	certain	chapters

two certain chapters

b.

certos	dois	capítulos
certain	two	chapters

two certain chapters

Ordinals cannot occur in NP initial position, though:

(74) a. um DVD com dois primeiros episódios dessa série
a DVD with two first episodes of that series
a DVD with two first episodes of that series

b. * um DVD com primeiros dois episódios dessa série
a DVD with first two episodes of that series

When these elements are preceded by a determiner, it is the determiner that introduces quantifier semantics. Assuming that quantifier semantics is always introduced by a determiner or a bare NP construction, there are two ways of introducing quantifiers in these NPs: considering them instances of bare NPs or analyzing the first element as a determiner.

The first possibility is not very attractive for Portuguese, for a number of factors. Preverbal bare NP subjects have a very constrained distribution in European Portuguese. It is interesting to note that NPs introduced by a cardinal (75c) do not pattern with bare NPs (75a) in the following examples, but rather with NPs introduced by determiners (75b).

(75) a. */?? Cartas chegaram.
letters have arrived

b. Algumas cartas chegaram.
some letters arrived
Some letters arrived.

c. Duas cartas chegaram.
two letters arrived
Two letters arrived.

The example in (75c) sounds as good as the one in (75b), which is introduced by the item “alguns”, which can only occur in NP initial position and is thus not a bare NP.

Second, bare NPs tend to have non-specific readings in Portuguese: they cannot scope over negation (76), universal quantifiers (77) or intensional verbs (78). These examples are Brazilian Portuguese, from [Munn and Schmitt, 1998] (corresponding logical formulas added for ease of exposition), but the same observations hold for European Portuguese. We also bracketed the relevant bare NPs in these examples.

(76) a. João não viu uma mancha no chão.
João not saw a spot on the floor
João didn't see a spot on the floor.

1. $\neg\exists x[spot\_on\_the\_floor(x) \wedge saw(João, x)]$
2. $\exists x[spot\_on\_the\_floor(x) \wedge \neg saw(João, x)]$

b. João não viu [manchas no chão.]
João not saw spots on the floor
João didn't see spots on the floor.

1. $\neg\exists x[spot\_on\_the\_floor(x) \wedge saw(João, x)]$

(77) a. Todo mundo leu um livro sobre girafas.
everyone read a book on giraffes
Everyone read a book on giraffes.

1. $\forall x[person(x) \rightarrow \exists y[book\_on\_giraffes(y) \wedge read(x, y)]]$
2. $\exists y[book\_on\_giraffes(y) \wedge \forall x[person(x) \rightarrow read(x, y)]]$

b. Todo mundo leu [livros sobre girafas.]
everyone read books on giraffes
Everyone read books on giraffes.
1. $\forall x[person(x) \rightarrow \exists y[book\_on\_giraffes(y) \wedge read(x, y)]]$

(78) a. Pedro quer encontrar um policial.
Pedro wants to meet a policeman
Pedro wants to meet a policeman.
1. $\exists x[policeman(x) \wedge want(Pedro, meet(Pedro, x))]$
2. $want(Pedro, \exists x[policeman(x) \wedge meet(Pedro, x)])$

b. Pedro quer encontrar [policiais.]
Pedro wants to meet policemen
Pedro wants to meet policemen.
1. $want(Pedro, \exists x[policeman(x) \wedge meet(Pedro, x)])$

NPs introduced by cardinals do not pattern with bare NPs in this respect and allow both readings. An example with negation is in (79), in which the semantics of the cardinal "duas"/*two* is represented by $\lambda P.\lambda Q.\exists x_1 \exists x_2[x_1 \neq x_2 \wedge P(x_1) \wedge P(x_2) \wedge Q(x_1) \wedge Q(x_2)]$. Ambiguity can be found in the other two contexts as well.

(79) João não viu duas manchas no chão.
João not saw two spots on the floor
João didn't see two spots on the floor.
1. $\neg\exists x_1 \exists x_2[x_1 \neq x_2 \wedge spot\_on\_the\_floor(x_1) \wedge spot\_on\_the\_floor(x_2)$
$\wedge saw(João, x_1) \wedge saw(João, x_2)]$
2. $\exists x_1 \exists x_2[x_1 \neq x_2 \wedge spot\_on\_the\_floor(x_1) \wedge spot\_on\_the\_floor(x_2)$
$\wedge \neg saw(João, x_1) \wedge \neg saw(João, x_2)]$

Furthermore, NPs introduced by markers of indefinite specifics should obviously not be analyzed as bare NPs if the latter are constrained to take non-specific readings:

(80) a. O João não viu certa mancha no chão.
the João not saw certain spot on the floor
João didn't see a certain spot on the floor.
1. $\exists x[spot\_on\_the\_floor(x) \wedge \neg saw(João, x)]$

b. Todas as pessoas leram certo livro sobre girafas.
all the people read certain book on giraffes
Everyone read a certain book on giraffes.
1. $\exists y[book\_on\_giraffes(y) \wedge \forall x[person(x) \rightarrow read(x, y)]]$

c. Pedro quer encontrar certo polícia.
Pedro wants to meet certain policeman
Pedro wants to meet a certain policeman.
1. $\exists x[policeman(x) \wedge want(Pedro, meet(Pedro, x))]$

Bare NPs do not co-occur with the "cada" (*each*) of (81), but NPs introduced by cardinals do (examples from [Müller, 2002]):

(81) a.

Os	países	da	UE	mandaram	um/dois/vários	delegado(s)	cada.
the	countries	of the	EU	sent	a/two/various	delegate(s)	each

The EU countries sent a/two/various delegate(s) each.

b. *

Os	países	da	UE	mandaram	delegados	cada.
the	countries	of the	EU	sent	delegates	each

We conclude that cardinals and markers of indefinite specifics at NP initial position are best treated as determiners. They introduce indefinite NPs and cannot co-occur with prenominal possessives. The constraints on the *marking* features must therefore be different from the ones on elements of Position IV. Therefore, the constraints on their HEAD must differ. NP initial cardinals also carry quantifier semantics, which the elements of Position IV arguably do not. We will be calling them cardinal determiners and indefinite specific determiners from now on.

The HEAD of an NP initial cardinal looks like this:

$$\begin{bmatrix} \textit{cardinal-det} & \\ \textsc{marker} & \begin{bmatrix} \textit{pre-only-marker} & \\ \textsc{mark} & \begin{bmatrix} \textit{saturated-marking} & \\ \textsc{mk-val} & \begin{bmatrix} \textsc{cardinal} & \textit{present} \\ \textsc{demonstrative} & \boxed{1} \\ \textsc{possessive} & \boxed{2} \\ \textsc{qualquer} & \boxed{3} \\ \textsc{tal} & \boxed{4} \\ \textsc{outro} & \boxed{5} \\ \textsc{indef-spec} & \boxed{6} \\ \textsc{ordinal} & \boxed{7} \end{bmatrix} \end{bmatrix} \\ \textsc{select}|\textsc{local}|\textsc{cat} & \begin{bmatrix} \textsc{head}\ \textit{noun} & \\ \textsc{marking} & \begin{bmatrix} \textit{no-det-marking} & \\ \textsc{mk-val} & \begin{bmatrix} \textsc{cardinal} & \textit{absent} \\ \textsc{demonstrative} & \boxed{1}\ \textit{absent} \\ \textsc{possessive} & \boxed{2}\ \textit{absent-or-posthead-present} \\ \textsc{qualquer} & \boxed{3}\ \textit{present-or-absent} \\ \textsc{tal} & \boxed{4}\ \textit{present-or-absent} \\ \textsc{outro} & \boxed{5}\ \textit{present-or-absent} \\ \textsc{indef-spec} & \boxed{6}\ \textit{present-or-absent} \\ \textsc{ordinal} & \boxed{7}\ \textit{present-or-absent} \end{bmatrix} \end{bmatrix} \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

The constraint on MK-VAL|CARDINAL of the selected synsem prevents cardinal determiners from iterating and from combining with the elements of Position IV.

The constraints for indefinite specific determiners are similar, with the obvious differences regarding the features CARDINAL and INDEF-SPEC.

There are no determiner versions of ordinals, as they cannot initiate an NP.

There are two questions to address: the relation between these determiners and the items of Position IV, and preventing bare NPs from being formed from NPs starting with an element in Position IV.

There are two possibilities for the first issue: to produce the determiner version from the postdeterminer one via a unary rule, or to have multiple lexical entries. In LXGram indefinite specifics receive multiple lexical entries, but cardinals do not. This is for reasons related to the composition of semantics and is explained in Section 8.6.5.1.

In LXGram bare NPs are produced by a unary syntactic rule that adds quantifier semantics, imposes a value of marking on the mother node subsumed by *saturated-marking* and requires the

$$\begin{bmatrix} \textit{mrs} & \\ \text{LTOP} & \boxed{h1}\ h \\ \text{INDEX} & \boxed{e2}\ e \\ \text{RELS} & \left\langle \begin{array}{l} \begin{bmatrix} \textit{\_o\_q\_rel} & \\ \text{LBL} & \boxed{h3}\ h \\ \text{ARG0} & \boxed{x4}\ x \\ \text{RSTR} & \boxed{h6}\ h \\ \text{BODY} & \boxed{h5}\ h \end{bmatrix}, \begin{bmatrix} \textit{cardinal\_rel} & \\ \text{LBL} & \boxed{h7}\ h \\ \text{ARG0} & \boxed{e9}\ e \\ \text{ARG1} & \boxed{h8}\ h \\ \text{ARG2} & \boxed{j10}\ j \end{bmatrix}, \begin{bmatrix} \textit{greater-or-equal\_rel} & \\ \text{LBL} & \boxed{h7} \\ \text{ARG0} & \boxed{j10} \\ \text{ARG1} & \boxed{j11}\ j \end{bmatrix}, \\ \begin{bmatrix} \textit{int-equals\_rel} & \\ \text{LBL} & \boxed{h7} \\ \text{ARG0} & \boxed{j11} \\ \text{CARG} & \textit{2} \end{bmatrix}, \begin{bmatrix} \textit{\_carro\_n\_rel} & \\ \text{LBL} & \boxed{h8} \\ \text{ARG0} & \boxed{x4} \end{bmatrix}, \begin{bmatrix} \textit{\_avariar\_v\_rel} & \\ \text{LBL} & \boxed{h12} \\ \text{ARG0} & \boxed{e2} \\ \text{ARG1} & \boxed{x4} \end{bmatrix} \end{array} \right\rangle \\ \text{HCONS} & \left\langle \begin{bmatrix} \textit{qeq} & \\ \text{HARG} & \boxed{h1} \\ \text{LARG} & \boxed{h12} \end{bmatrix}, \begin{bmatrix} \textit{qeq} & \\ \text{HARG} & \boxed{h6} \\ \text{LARG} & \boxed{h7} \end{bmatrix} \right\rangle \end{bmatrix}$$

Figure 8.16: MRS of a sentence with a postdeterminer cardinal. The sentence is "os dois carros avariaram" (*the two cars broke down*).

daughter to be a noun headed sign with a MARKING subsumed by *no-det-marking* (see Section 8.8). In order to prevent bare NPs to be built from constituents that include a postdeterminer cardinal, ordinal or marker of indefinite specifics, the daughter is also constrained to have the features ORDINAL, CARDINAL and INDEF-SPEC under MARKING|MK-VAI of type *absent*.

8.6.5.1 Cardinal Determiners and the Semantics of Cardinals

In LXGram we chose to relate cardinal determiners and cardinal postdeterminers via unary syntactic rules. In particular, the determiner versions are produced from the postdeterminer versions. This is tied to issues of composition of semantics.

For the postdeterminer cardinals, an example of the MRSs produced is in Figure 8.16. The cardinal corresponds to three relations in this MRS: the *cardinal_rel* relation, the *greater-or-equal_rel* relation, and the *int-equals_rel* relation.

In the literature, there are several approaches to the semantics of cardinals: they have been given the semantic types $\langle e,t\rangle$ (a set of entities),[4] $\langle\langle e,t\rangle,\langle e,t\rangle\rangle$ (a function from sets to sets) or $\langle\langle e,t\rangle,\langle\langle e,t\rangle,t\rangle\rangle$ (a determiner). We did not choose to give cardinals quantifier semantics, because they can occur after determiners, as in expressions like *all three*. When they do appear in NP initial position, quantifier semantics must be added, though. We opted for the $\langle\langle e,t\rangle,\langle e,t\rangle\rangle$ treatment (i.e. consider them modifiers), and do not commit to saying that cardinals are intersective modifiers.[5] Therefore the *cardinal_rel* scopes over the relation introduced

[4]Or rather its characteristic function, a function from entities to truth values yielding true for all members of that set and for them only, i.e. a function of the form $\lambda x_{\in D_e}.P(x)$.

[5]We do not commit to saying that this function is $\lambda P_{\in D_{\langle e,t\rangle}}.\lambda x_{\in D_e}.P(x) \wedge Q(x)$, for some lexically given set Q. For example, if the denotation of *car* is $\lambda x_{\in D_e}.car'(x)$, the denotation of *two cars* would be $(\lambda P_{\in D_{\langle e,t\rangle}}.\lambda x_{\in D_e}.P(x) \wedge 2(x))(\lambda x_{\in D_e} car'(x)) = \lambda x_{\in D_e}.car'(x) \wedge 2(x)$ if intersective semantics is given to cardinals. This only makes sense if we consider the existence of plural (non-atomic) entities, whose atoms can be counted, in which case the set denoted by 2 in the above formula is the set of all plural entities with two atoms. There are many views on the semantics of cardinals, and we remain neutral with respect to the status of plural entities. In [Ionin and Matushansky, 2006]

by the head noun in MRSs. This is compatible with intersective semantics but does not enforce it. Our representation for cardinals is similar to $\lambda i_{\in I}.\lambda P_{\in D_{\langle e,t\rangle}}.\lambda x_{\in D_e}.cardinal(e, P(x), i)$, where the integer argument i is supplied in each lexical entry for cardinals. However, we do not define the meaning of the *cardinal* relation. It can be intersective if we posit that $\lambda i.\lambda P.\lambda x.cardinal(e, P(x), i) = \lambda i.\lambda P.\lambda x.P(x) \wedge count(P, i)$, where $\lambda i.\lambda x.count(x, i)$ is true if x is a plural entity with i atoms. Its meaning can however be defined differently, not necessarily in an intersective way.

Note that a definition that constrains the cardinality of the set denoted by the noun does not work. For instance a sentence like *three cars broke down* does not mean that the cardinality of the set of cars is three, but rather that the cardinality of the intersection of the set of cars and the set of things that broke down is three. Using plural entities, this sentence would be assigned a representation that says that there is a plural entity consisting of three cars that also belongs to the set of things that broke down, i.e. it is simple existential quantification, which is the semantics we will assume for cardinals occupying a determiner position (see below).

The other relations describe the integer argument of the *cardinal_rel* relation. It is widely assumed that an expression like *two children* means *at least two children* and not *exactly two children*. In the following example, the answer would be contradictory if *two children* meant *exactly two children*:

(82) — Do you have two children?
— Yes. In fact I have three.

The relevant piece of semantics is $cardinal(e_9, \_carro\_n(x_4), j_{10}) \wedge greater\text{-}or\text{-}equal(j_{10}, j_{11}) \wedge int\text{-}equals(j_{11}, 2)$. Variables of the form jn, where $n > 0$ are integer variables.[6] We can view these integer variables as existentially quantified by convention, so we do not explicitly include these quantifiers in the MRSs.

It would be more simple to produce $cardinal(e9, \_carro\_n(x4), 2)$, assuming that the relation greater or equal is part of the meaning of *cardinal_rel*.

The motivation for introducing the *greater-or-equal_rel* relation is that in certain contexts we do not want it to appear in the MRSs. This is the case of expressions like "exactly two" or "at most two". For an expression like "no máximo dois"/*at most two*, we can think of the semantics $\lambda P.\lambda x.cardinal(e, P(x4), j1) \wedge less\text{-}or\text{-}equal(j1, j2) \wedge int\text{-}equals\_rel(j2, 2)$. In order to factor out the similarity with the representation for an unmodified "dois"/*two*, we explicitly introduce *greater-or-equal_rel* relation when a cardinal is not modified.[7]

The use of the *int-equals_rel* relation is a matter of convenience. It is not necessary, because, instead of the piece of semantics $cardinal(e9, \_carro\_n(x4), j10) \wedge greater\text{-}or\text{-}equal(j10, j11) \wedge int\text{-}equals(j11, 2)$, we could simply use $cardinal(e9, \_carro\_n(x4), j10) \wedge greater\text{-}or\text{-}equal(j10, 2)$.

It is more convenient for the generation algorithm in the LKB to associate at least one relation with every lexical item. If we did not include this relation in the lexical entry for cardinals, their only semantic content would be the integer constant that is an argument of relations like *greater-or-equal_rel* or *less-or-equal_rel*. The implementation in LXGram associates to lexical items for cardinals only the *int-equals_rel* relations. All other relations are introduced in syntax, via rules that add semantics.

there are references to the main pieces of work in this field.

[6]They can be created by manipulating the LKB configuration files, namely by redefining the function **determine-variable-type** in the file `mrsglobals.lisp`.

[7]If we assumed that the *greater-or-equal_rel* relation is part of the meaning of *cardinal_rel* so that we would not have to include it in MRSs when a cardinal is not modified, expressions like "at most two" would not receive the correct semantics, or we could not use the *cardinal_rel* in these cases.

The first set of rules allows these expressions to combine with cardinal modifiers like *exactly*, *at most*, etc. Only one modifier is allowed, and if no modifier is present, a unary syntactic rule is used to add the *greater-or-equal_rel*. A cardinal modifier like *at most* introduces a *less-or-equal_rel*, a modifier like *exactly* introduces no relation.

Immediately up the tree, a unary rule is used to add the *cardinal_rel* relation and producing a node with a HEAD of type *cardinal*. After this step the piece of semantics for "dois" (*two*) and for "pelo menos dois" (*at least two*) is like $\lambda P.\lambda x.cardinal(e, P(x), j_1) \wedge \textit{greater-or-equal}(j_1, j_2) \wedge \textit{int-equals}(j_2, 2)$. The semantics for "no máximo dois" (*at most two*) is like $\lambda P.\lambda x.cardinal(e, P(x), j_1) \wedge \textit{less-or-equal}(j_1, j_2) \wedge \textit{int-equals}(j_2, 2)$. The semantics for "exactamente dois" (*exactly two*) is like $\lambda P.\lambda x.cardinal(e, P(x), j_1) \wedge \textit{int-equals}(j_1, 2)$.

An optional rule can apply afterwards, changing the postdeterminer cardinal into a determiner (*cardinal-det* above) and adding quantifier semantics. So cardinal determiners are produced from cardinal postdeterminers via a syntactic rule.

We will not show the details of all these rules since they are relatively trivial. To control order of rule application LXGram uses different values of HEAD for these elements: many of these rules are non-headed. Only the two highest rules create nodes with values of *head* that inherit from *functor* and that can attach to nominal projections. These subtypes of *head* and their definitions (*cardinal* and *cardinal-det*) have already been presented.

This analysis is completely monotonic: we only add relations to an MRS, never remove or alter relations introduced elsewhere. This is a requirement of the LKB: composition of semantics has to be monotonic so that efficient algorithms can be used for generation. Also, every lexical entry for cardinals and every rule used in this process contributes at least one relation to the MRS.

Although a large number of dedicated rules is involved, they are used to build the semantics little by little and factor out the commonalities between the various pieces of MRS that are related to cardinals.

We assume that complex cardinal expressions like "vinte e um"/*twenty one* are recognized by a Named Entity Recognizer (NER) in a preprocessing step, and for the purposes of the grammar behave just like atomic cardinals like "vinte"/*twenty*. There is one NER developed in the University of Lisbon [Ferreira et al., 2007], that can be integrated with LXGram. Since NERs are not necessarily bidirectional, we can parse these expressions but we cannot generate them so far.

8.6.6 Modifying Adjectives

On a first approximation, adjectives select for a constituent with [MARKING *n-marking*] and produce a node with the same level of saturation:

$$\begin{bmatrix} \textit{adjective} & \\ \text{MARKER} & \begin{bmatrix} \text{SELECT|LOCAL|CAT} & \begin{bmatrix} \text{HEAD} & \textit{noun} \\ \text{MARKING} & \textit{n-marking} \end{bmatrix} \\ \text{MARK}\ \textit{n-marking} & \end{bmatrix} \end{bmatrix}$$

As a consequence, they are allowed to recur.

Portuguese has prenominal and postnominal adjectives. Potentially spurious attachment ambiguities will be produced for a sequence AP_1-Noun-AP_2: [AP_1 [Noun AP_2]] and [[AP_1 Noun] AP_1]. Although spurious ambiguity is innocuous, it is also a source of inefficiency, as it causes the parser to perform more computations than needed. In LXGram the type hierarchy of *marking* is used to control this, too.

Examples like the one in (83) argue in favor of the structure [AP_1 [Noun AP_2]], since this NP can describe someone who is not Chinese. Accordingly, we want to provide to such NP semantics like $\lambda P.\_um\_q(x, \_falso\_a(e_1, \_médico\_n(x) \wedge \_chinês\_a(e_2, x)), P(x))$.[8] It does not describe a Chinese person who is a fake doctor (i.e. $\lambda P.\_um\_q(x, \_falso\_a(e_1, \_médico\_n(x)) \wedge \_chinês\_a(e_2, x), P(x)))$. Assuming syntactic scope and semantic scope match, the structure [AP_1 [Noun AP_2]] is justified.[9]

(83) um falso médico chinês
a fake doctor Chinese
a fake Chinese doctor

Prenominal adjectives are specified to have the constraint [MARK *prenom-adj-marking*] and select for nominal projections with [MARKING *prenom-adj-or-n-marking*], while postnominal adjectives select for sister nodes with [MARKING *n-marking*] and also bear the value *n-marking* for their MARK attribute. The type of HEAD in adjectives has the following constraints:

$$\begin{bmatrix} \textit{adjective} & \\ \text{MARKER} & \begin{bmatrix} \text{SELECT|LOCAL|CAT} & \begin{bmatrix} \text{HEAD} & \textit{noun} \\ \text{MARKING} & \textit{prenom-adj-or-n-marking} \end{bmatrix} \\ \text{MARK } \textit{prenom-adj-or-n-marking} & \end{bmatrix} \end{bmatrix}$$

In the lexical types for adjectives, we distinguish between the adjectives that can only precede the noun, the ones that can only follow it and the ones that can occur in either position. The following examples illustrate these three classes. An adjective like "mero" (*mere*) can only precede the noun, an adjective like "japonês" (*Japanese*) can only follow the noun, and an adjective like "falso" (*false*) can precede or follow it.

(84) a. Atacaram um mero inspector.
they attacked a mere inspector
They attacked a mere inspector.

b. * Atacaram um inspector mero.
they attacked an inspector mere

c. * Atacaram um japonês inspector.
they attacked a Japanese inspector

d. Atacaram um inspector japonês.
they attacked an inspector Japanese
They attacked a Japanese inspector.

e. Atacaram um falso inspector.
they attacked a false inspector
They attacked a false inspector.

f. Atacaram um inspector falso.
they attacked an inspector false
They attacked a false inspector.

[8] We can assume that the semantic representation of "falso" (*fake*), $\lambda P_{\in D_{\langle e,t\rangle}}.\lambda x_{\in D_e}.\_falso\_a(e, P(x))$, means $\lambda P.\lambda x.\neg P(x)$.

[9] It is not required that syntactic and semantic scope match, because it is possible to manipulate feature structures, but it is desirable that they do, since implementation becomes more straightforward if they match. We thus assume that syntax and semantics match in the absence of a compelling argument against it.

The lexical types for the adjectives that can precede the noun have the constraints:

$$\left[\textsc{synsem}|\textsc{local}|\textsc{cat}|\textsc{head} \begin{bmatrix} \textit{adjective} \\ \textsc{marker}|\textsc{prehead}|\textsc{mark}\ \textit{prenom-adj-marking} \end{bmatrix}\right]$$

The lexical types for the ones that can follow the noun are constrained with:

$$\left[\textsc{synsem}|\textsc{local}|\textsc{cat}|\textsc{head} \begin{bmatrix} \textit{adjective} \\ \textsc{marker}|\textsc{posthead} \begin{bmatrix} \textsc{select}|\textsc{local}|\textsc{cat}|\textsc{marking}\ \textit{n-marking} \\ \textsc{mark}\ \textit{n-marking} \end{bmatrix} \end{bmatrix}\right]$$

The adjectives that can follow or precede the noun inherit all these constraints. The ones that can only precede it are given a lexical type that inherits from the type where the constraints on PREHEAD are stated and is further constrained with:

$$\left[\textsc{synsem}|\textsc{local}|\textsc{cat}|\textsc{head}|\textsc{marker}\ \textit{pre-only-marker}\right]$$

Likewise, the lexical type for the adjectives that can only follow the noun inherits from the type above that has a constrained POSTHEAD feature and is defined to also bear:

$$\left[\textsc{synsem}|\textsc{local}|\textsc{cat}|\textsc{head}|\textsc{marker}\ \textit{post-only-marker}\right]$$

In this implementation, nouns are given the same syntactic distribution as noun-adjective sequences: they can combine with another adjective to their right, or with a prenominal adjective. Nouns have a syntactic distribution different from adjective-noun sequences, as the latter cannot combine with an adjective to their right.

With this system of constraints, the noun phrase "um médico chinês falso" receives a semantic representation equivalent to "um falso médico chinês", equivalent to the lambda formula presented above. On the other hand, a noun phrase like "um médico falso Chinês" (*a fake doctor who is Chinese*) receives semantics equal to $\lambda P.\_um\_q(x, \_falso\_a(e_1, \_médico\_n(x)) \wedge \_chinês\_a(e_2, x), P(x))$, based on the syntactic structure ["um" [["médico falso"] "chinês"]].

Adjectives are allowed to iterate in both positions (prenominal and postnominal). This is borne out by data like:

(85) a.

Era	um	grande,	grande filme.
it was	a	great	great movie

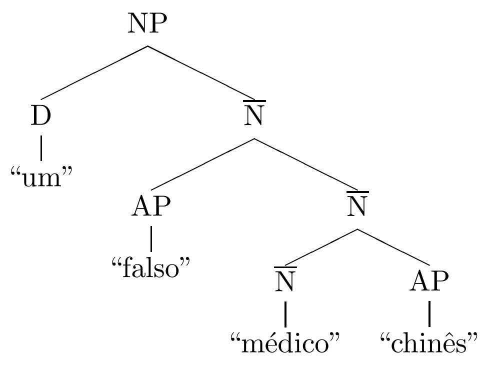
It was a great, great movie.

b.

Era	um	filme	chato,	chato.
it was	a	movie	boring	boring

It as a boring, boring movie.

It is worth pointing out that we cannot properly capture the meaning difference between an $\overline{\text{N}}$ like "filme chato" (*boring movie*), which receives semantics equivalent to $\lambda x.\_filme\_n(x) \wedge \_chato\_a(e1, x)$, and an $\overline{\text{N}}$ like "filme chato, chato" (*boring, boring movie*), which is assigned an MRS representation equivalent to $\lambda x.\_filme\_n(x) \wedge \_chato\_a(e1, x) \wedge \_chato\_a(e2, x)$: the two formulas are logically equivalent due to idempotence of conjunction if we ignore the different event variables. It is not clear that the difference is truly semantic, anyway. It may simply be a pragmatic effect.



$$
\left[\begin{array}{ll}
\multicolumn{2}{l}{\textit{functor-head-phrase}}\\
\text{STEM} & \left\langle \text{“um”, “falso”, “médico”, “chinês”} \right\rangle\\
\text{SYNSEM}|\text{LOCAL}|\text{CAT} & \left[\begin{array}{lll}\text{HEAD} & \boxed{1} & \textit{noun}\\ \text{MARKING} & \boxed{2} & \textit{saturated-marking}\end{array}\right]\\
\text{NON-HEAD-DTR} & \left[\begin{array}{ll}\text{STEM} & \left\langle \text{“um”} \right\rangle\\ \text{SYNSEM}|\text{LOCAL}|\text{CAT}|\text{HEAD} & \left[\begin{array}{ll}\multicolumn{2}{l}{\textit{determiner}}\\ \text{MARKER} & \left[\begin{array}{ll}\text{SELECT} & \boxed{3}\\ \text{MARK} & \boxed{2}\end{array}\right]\end{array}\right]\end{array}\right]\\
\text{HEAD-DTR} & \left[\begin{array}{ll}
\multicolumn{2}{l}{\textit{functor-head-phrase}}\\
\text{STEM} & \left\langle \text{“falso”, “médico”, “chinês”} \right\rangle\\
\text{SYNSEM } \boxed{3} & \left[\text{LOCAL}|\text{CAT} \left[\begin{array}{lll}\text{HEAD} & \boxed{1} & \\ \text{MARKING} & \boxed{4} & \textit{prenom-adj-marking}\end{array}\right]\right]\\
\text{NON-HEAD-DTR} & \left[\begin{array}{ll}\text{STEM} & \left\langle \text{“falso”} \right\rangle\\ \text{SYNSEM}|\text{LOCAL}|\text{CAT}|\text{HEAD} & \left[\begin{array}{ll}\multicolumn{2}{l}{\textit{adjective}}\\ \text{MARKER} & \left[\begin{array}{ll}\text{SELECT} & \boxed{5}\\ \text{PREHEAD} & \left[\begin{array}{ll}\text{SELECT} & \boxed{5}\\ \text{MARK} & \boxed{4}\end{array}\right]\end{array}\right]\end{array}\right]\end{array}\right]\\
\text{HEAD-DTR} & \left[\begin{array}{ll}
\multicolumn{2}{l}{\textit{head-functor-phrase}}\\
\text{STEM} & \left\langle \text{“médico”, “chinês”} \right\rangle\\
\text{SYNSEM } \boxed{5} & \left[\text{LOCAL}|\text{CAT} \left[\begin{array}{lll}\text{HEAD} & \boxed{1} & \\ \text{MARKING} & \boxed{6} & \textit{n-marking}\end{array}\right]\right]\\
\text{NON-HEAD-DTR} & \left[\begin{array}{ll}\text{STEM} & \left\langle \text{“chinês”} \right\rangle\\ \text{SYNSEM}|\text{LOCAL}|\text{CAT}|\text{HEAD} & \left[\begin{array}{ll}\multicolumn{2}{l}{\textit{adjective}}\\ \text{MARKER} & \left[\begin{array}{ll}\text{SELECT} & \boxed{7}\\ \text{POSTHEAD} & \left[\begin{array}{ll}\text{SELECT} & \boxed{7}\\ \text{MARK} & \boxed{6}\end{array}\right]\end{array}\right]\end{array}\right]\end{array}\right]\\
\text{HEAD-DTR} & \left[\begin{array}{ll}\text{STEM} & \left\langle \text{“médico”} \right\rangle\\ \text{SYNSEM } \boxed{7} & \left[\text{LOCAL}|\text{CAT} \left[\begin{array}{ll}\text{HEAD} & \boxed{1}\\ \text{MARKING} & \textit{n-marking}\end{array}\right]\right]\end{array}\right]
\end{array}\right]
\end{array}\right]
\end{array}\right]
$$

Figure 8.17: Syntactic analysis for an NP with a prenominal and a postnominal adjective

The syntactic analysis produced by LXGram for the NP in (83) ("um falso médico chinês" — *a false Chinese doctor*) is in Figure 8.17, with abridged feature structures.

The structure [[AP_{1} N] AP_{2}] is blocked, because the phrase with the form [AP_{1} N] has MARKING with the value *prenom-adj-marking* but postnominal adjectives select for a sister node with the value *n-marking* for that feature. There is no unifier for *n-marking* and *prenom-adj-marking*, as can be seen in Figure 8.9.

8.6.7 Argumental Adjectives

Semantics

Adjectives that are used as an argument of nouns (86a) display drastically different semantics from adjectives that modify a noun (86b). Consider the two examples:

(86) a. Viram [$_{\mathrm{NP}}$ a alunagem americana] na televisão.
they saw the moon landing American on the television
They saw the American moon landing on TV.

b. Viram [$_{\mathrm{NP}}$ um carro americano] naquela rua.
they saw a car American on that street
They saw an American car on that street.

The NP in the first example has semantics quite similar to an NP like "a alunagem pelos americanos" ("the moon landing of the Americans"). The semantics for this NP could be

$$\lambda P_{\in D_{\langle e,t\rangle}}.\_o\_q(x, \_o\_q(y, \_americano\_n(y)), \_alunagem\_n(x,y)), P(x))$$

For the NP in the first example ("a alunagem americana" — *the American moon landing*) we could thus think of similar semantics. The semantics for the noun "alunagem" is

$$\lambda \mathcal{Q}_{\in D_{\langle\langle e,t\rangle t\rangle}}.\lambda x_{\in D_e}.\mathcal{Q}(\lambda y_{\in D_e}.alunagem(x,y))$$

Assuming that semantically, the noun is the functor and the adjective is the argument, the semantics for the argumental adjective in (86a) would have to be

$$\lambda P_{\in D_{\langle e,t\rangle}}.\_o\_q(z, \_americano\_n(z), P(z))$$

The most simple semantics for the modifying adjective "americano" in (86b) is

$$\lambda P_{\in D_{\langle e,t\rangle}}.\lambda x_{\in D_e}.P(x) \wedge \_americano\_a(e,x)$$

The semantics for the NP in (86b) is thus

$$\lambda P_{\in D_{\langle e,t\rangle}}.\_um\_q(x, \_carro\_n(x) \wedge \_americano\_a(e,x), P(x))$$

The same adjective in these two contexts presents very different semantics. There are two options: to have multiple lexical entries for the adjectives that can occur as modifiers and as arguments; to use an optional lexical rule to change the meaning and syntactic properties of such adjectives, producing one of the versions from the other, which would be in the lexicon.

The lexical rule approach is certainly more appealing, since adjectives that can occur as arguments would simply receive a special lexical type in their lexical entry, denoting this property. The problem is that we cannot produce one of the semantic representations from the other with the machinery in the LKB, because we cannot manipulate strings, and the mapping between the relation names *_americano_n* and *_americano_a* requires string manipulation.

We address this by providing a slightly different semantics to these adjectives when they are used as modifiers:

$$\lambda P_{\in D_{\langle e,t\rangle}}.\lambda x_{\in D_e}.udef\_q(y, americano\_n(y), P(x) \wedge abstract\_a(e,x,y))$$

Using the MRS format, this semantics can be easily produced from the semantics the adjective displays when it occurs as an argument. Therefore, the lexical entries for the adjectives that can occur in both positions have argumental semantics, and an optional lexical rule adds the *abstract_a* relation.

Under this model, the NP "o carro americano" receives the semantics:

$$\lambda P_{\in D_{\langle e,t\rangle}}.\_o\_q(x, udef\_q(y, americano\_n(y), \_carro\_n(x) \wedge abstract\_a(e,x,y), P(x))$$

The relation named *abstract_a* denotes a relation we cannot determine systematically. In this example, it links "car" with "Americans" and can be understood as "produced by". The relation $\lambda x_{\in D_e}.\lambda y_{\in D_e}.abstract\_a(e,x,y)$ is intended to mean $\lambda x_{\in D_e}.\lambda y_{\in D_e}.\exists R_{\in D_{\langle e,\langle e,t\rangle\rangle}} R(x,y)$.

Adjectives that cannot be used as arguments of nouns (e.g. "amarelo" — *yellow*) still receive standard adjective semantics:

$$\lambda P_{\in D_{\langle e,t\rangle}}.\lambda x_{\in D_e}.P(x) \wedge \_amarelo\_a(e,x)$$

This implementation of argumental adjectives is experimental, and a final solution will be sought in the next development phases.

Syntax

A PP complement cannot intervene between a noun and an adjectival complement:

(87)	a.		a	invasão	americana	do	Iraque
			the	invasion	American	of the	Iraq
			The American invasion of Iraq				
	b.	*	a	invasão	do	Iraque	americana
			the	invasion	of the	Iraq	American

Also, the remaining elements in Position VIII cannot appear before an adjective argument either. An example with a PP adjunct follows:

(88)	a.		a	alunagem	americana	de	1969
			the	moon landing	American	of	1969
			the American moon landing of 1969				
	b.	*	a	alunagem	de	1969	americana
			the	moon landing	of	1969	American

In LXGram, we use a dedicated syntactic rule similar to Head-Complement constructions that requires the head daughter to be a noun that selects for a PP complement (cf. "the American moon landing" with "the moon landing by the Americans") and the non-head daughter to be an adjective with argumental semantics.

We resort to two important subtypes of *synsems*: *lex-synsem* and *phrase-synsem*. The SYNSEM feature of all words (terminal symbols and lexical rules) is of type *lex-synsem*, and the SYNSEM of phrases is of type *phrase-synsem*. These types are incompatible: they have no common subtype.

In order to force strict adjacency between the noun head and the adjective argument, all that is necessary is that the head daughter of this syntactic rule dedicated to project adjectival arguments to the right of a noun be constrained to have a SYNSEM of type *lex-synsem*. Because of this constraint, nothing can intervene between the noun and this type of adjective, because, if that happened, a phrasal node would have to be the head daughter of this construction. This constraint also has the nice side effect of blocking two adjectival arguments of the same noun. This blocks the following ungrammatical example:

(89) a. *a invasão americana iraquiana
the invasion American Iraqi

In this special rule, the MARKING value of the mother node is also *n-marking*.

8.6.8 Noun Complementation

Many nouns subcategorize for one or more complements, that can be of different kinds. For the sake of illustration, here we will focus only on nouns with a single PP complement.

The standard HPSG approach to project complements is assumed: subcategorized for complements are members of a list-valued attribute COMPS in the lexical entry of the corresponding head, and a syntactic rule projects elements in that list, producing a mother node with a reduced COMPS.

Following many computationally implemented HPSGs, like the LinGO English Resource Grammar or the LinGO Grammar Matrix, strict binary branching is assumed — in the case of multiple complements, they are discharged one at a time. The Head-Complement syntactic rule or rules therefore unify the SYNSEM of the non-head daughter with the first element in the COMPS of the head daughter, and the COMPS of the mother node is the tail of the COMPS of the head daughter.

An issue in focus here is the relative scope between complements and the various functors. In Portuguese, the relative order between complements and several adnominal constituents (the ones in Position VIII in Table 8.3) is free. Consider the examples in (90).

(90) a. o consumo galopante$_{\text{AP}}$ [$_{\text{PP}}$ de petróleo]
the consumption ever increasing of oil
the ever increasing consumption of oil

b. o consumo [$_{\text{PP}}$ de petróleo] galopante$_{\text{AP}}$
the consumption of oil ever increasing
the ever increasing consumption of oil

These examples show that word order between postnominal adjunct adjectives and PP complements is arbitrary. Similar data can be presented for the other elements in Position VIII.

With other functors, however, word order is not free. Indeed, PP complements surface before restrictive relative clauses (but see Section 8.6.10):

(91) a. o consumo [$_{\text{PP}}$ de petróleo] [$_{\text{RelCl}}$ que continua a crescer]
the consumption of oil that continues to increase
the consumption of oil that continues to increase

b. *o consumo [$_{\text{RelCl}}$ que continua a crescer] [$_{\text{PP}}$ de petróleo]
the consumption that continues to grow of oil

The exact constraints on the position of relative clauses within an NP, as they are implemented in LXGram, are presented in Section 8.6.10.

Since complements occupy the same word order slot as the functors that give rise to constituents with *n-marking*, the relative syntactic scope between complements and the remaining functors must be the same as the relative scope between *n-marking* functors and the rest.

If complement placement is not constrained, many attachment ambiguities will surface. There will be no corresponding differences in the semantics produced, because the semantic constraints that link the MRS representation for the noun and the MRS representation for its complements are completely lexical (given in the lexical entries for nouns) and not affected by syntax.

There are two possible solutions to prevent this spurious overgeneration. The first one is to have all functors except the ones that occur in Position VIII select for a projection with saturated COMPS. Since the functors that occur in Position VIII are the most deeply embedded ones, if a complement is projected, it can only occur also in this position.

The second solution involves constraining the value of MARKING in the mother node of Head-Complement rules to be of the type *basic-marking* (see Figure 8.9; *basic-marking* is a supertype of other values of *marking*, not shown in that figure, for phrases not headed by a noun). For instance, if the head daughter of a Head-Complement constructions is a phrase introduced by a cardinal, that phrase will have a MARKING value that is the unifier of the type *no-det-marking* (the type of the feature SYNSEM|LOCAL|CAT|HEAD|MARKER|MARK of cardinals; see Section 8.6.4) with the type *basic-marking* (this is the contraint on Head-Complement rules just mentioned). This unifier is the type *n-marking* accoording to the hierarchy in Figure 8.9. This type *n-marking* is defined to have a feature MK-VAL|CARDINAL with the value *absent* (see Section 8.6.1). However the value for this feature will be *present* for such a phrase, because of the contraints on cardinals (cardinals have the value *present* for their feature SYNSEM|LOCAL|CAT|HEAD|MARKER|MARK, as presented in Section 8.6.4). The types *absent* and *present* are incompatible (Figure 8.10). This shows how cardinals are prevented to attach lower than complements. The control on the attachment position of the other functors with respect to complements is similar. In particular, all functors with a MARK feature that includes constraints incompatible with the constraints on the type *n-marking* are forced to attach higher than complements.

In either case, Head-Complement rules unify the MARKING value of the head daughter with the MARKING value of the mother node:

$$\begin{bmatrix} \text{SYNSEM}|\text{LOCAL}|\text{CAT}|\text{MARKING } \boxed{1} \\ \text{HEAD-DTR}|\text{SYNSEM}|\text{LOCAL}|\text{CAT}|\text{MARKING } \boxed{1} \end{bmatrix}$$

The second solution has an important advantage over the first one: the level of saturation at which complements attach is stated in a single place, a type for Head-Complement constructions. This is the solution used in LXGram. Note that *basic-marking* results in *n-marking* when unified with any type under *no-det-marking* (except for *prenom-adj-marking* and *rel-marking*, which are never selected for by a functor; see Figure 8.9).

Figure 8.18 shows the analysis of an NP with an adjective intervening between the head and the complement. The NP is "um membro provável do IRA" (*a probable member of the IRA*).

In this example the node labeled $\overline{\text{N}}$ is produced via a Head-Complement construction. The remaining phrasal nodes are produced via Head-Functor constructions.

The semantic representation for this NP produced by LXGram is equivalent to

$$\lambda P_{\in D_{\langle e,t\rangle}}.proper\_q(x_1, named(x_1, \text{“}IRA''), \_um\_q(x_2, \_provável(e, \_membro\_n(x_2, x_1)), P(x_2)))$$

8.6.9 PPs and AdvPs

PPs and some AdvPs can modify a noun on their left. Some examples are given in (92).

(92) a. pessoas [$_{\text{PP}}$ com mobilidade reduzida]
people with mobility reduced
people with reduced mobility

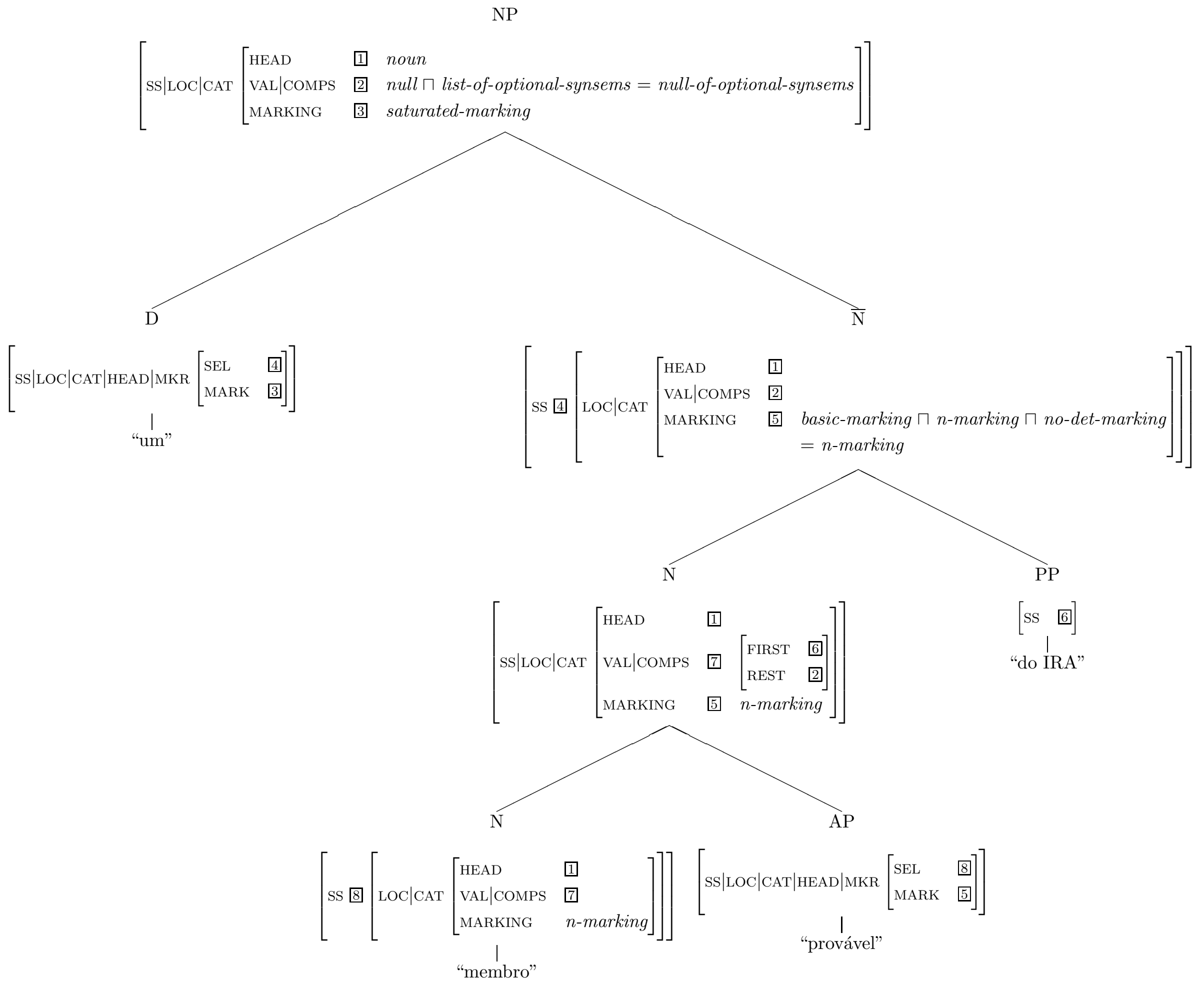


Figure 8.18: Syntactic analysis for "um membro provável do IRA" (*a probable member of the IRA*). SS abbreviates SYNSEM, LOC abbreviates LOCAL, MKR abbreviates MARKER, and SEL abbreviates SELECT.

b.

[$_{NP}$ Aquele	carro	ali$_{AdvP}$	que	tem	um	pneu	furado]	[$_{VP}$ estava	estacionado	aqui	ontem.	]
that	car	there	that	has	a	tire	flat	was	parked	here	yesterday	

That car over there that has a flat tire was parked here yesterday.

The adverb "ali" (*there*) in (92b) must be analyzed as part of the subject of the main clause, since it occurs in an NP internal position (following the noun "carro" and preceding the relative clause).

PPs and AdvPs cannot precede the noun, as shown in (93).

(93) a.

* [$_{PP}$ com	mobilidade	reduzida]	pessoas	
with	mobility	reduced	people	

b.

* Aquele	ali$_{AdvP}$	carro	estava	estacionado	aqui	ontem.
that	there	car	was	parked	here	yesterday

Inside the NP, they have the syntactic distribution of postnominal adjectives (Position VIII in Table 8.3). In fact, because PPs and APs can be interspersed, ambiguity can arise concerning adjective attachment — consider (94).

(94)

carros	sem	assentos	vermelhos
cars	without	seats	red

red cars with no seats/cars without red seats

This example has two interpretations depending on the attachment site of the adjective: [[carros [$_{PP}$ sem assentos]] vermelhos$_{AP}$] (*red cars with no seats*) and [carros [$_{PP}$ sem assentos vermelhos]] (*cars without red seats*). This fact leads one to posit constraints on the head types of prepositions and adverbs similar to the ones on postnominal adjectives, as far as the feature POSTHEAD is concerned.

The constraints on the head of prepositions and adverbs that can modify nouns (as well as verbs) thus look like:

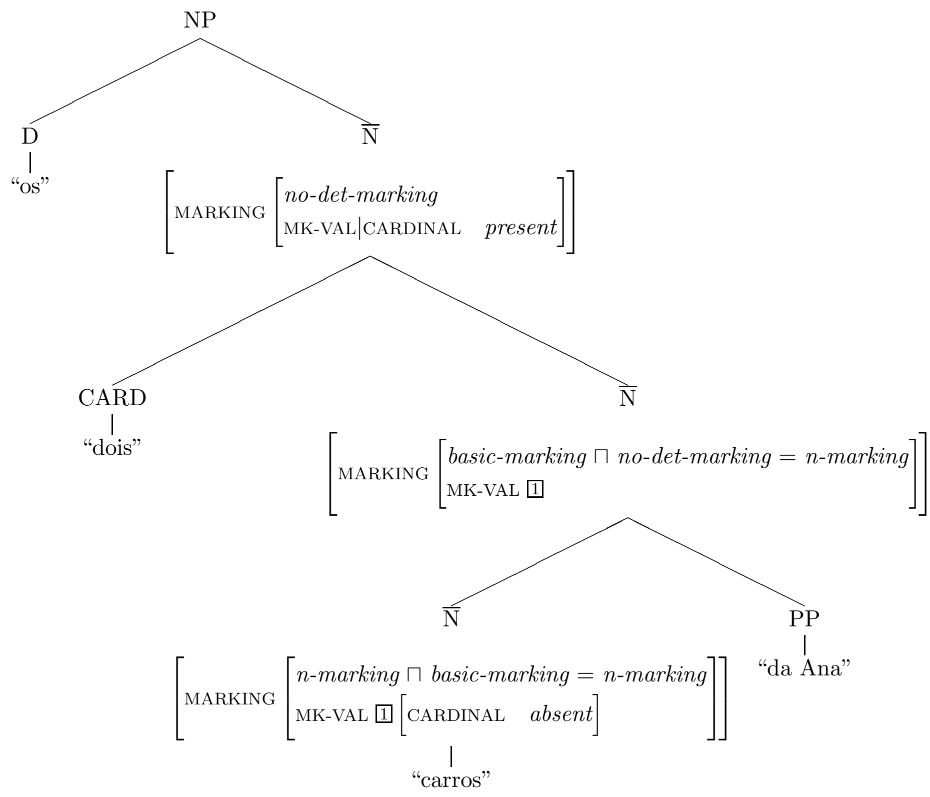
$$\begin{bmatrix} \text{MARKER} & \begin{bmatrix} \text{MARK} & \begin{bmatrix} \textit{basic-marking} \\ \text{MK-VAL} \ \boxed{1} \end{bmatrix} \\ \text{SELECT|LOCAL|CAT} & \begin{bmatrix} \text{HEAD} & \textit{noun-or-verb} \\ \text{MARKING} & \begin{bmatrix} \textit{basic-marking} \\ \text{MK-VAL} \ \boxed{1} \end{bmatrix} \end{bmatrix} \\ \text{PREHEAD|SELECT|LOCAL|CAT|HEAD} & \textit{verb} \end{bmatrix} \end{bmatrix}$$

The analysis for the NP in (95) is shown below.

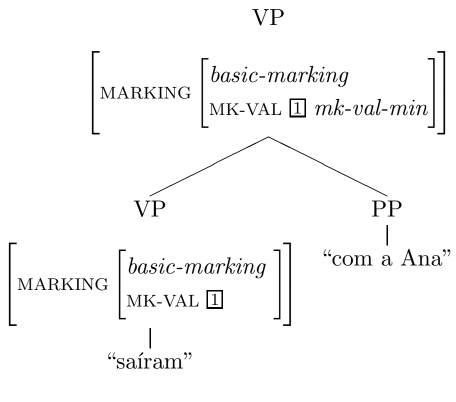
(95)

os	dois	carros	da	Ana
the	two	cars	of the	Ana

Ana's two cars


NP
D
"os"
N̄
MARKING [no-det-marking, MK-VAL|CARDINAL present]
CARD
"dois"
N̄
MARKING [basic-marking ⊓ no-det-marking = n-marking, MK-VAL 1]
N̄
MARKING [n-marking ⊓ basic-marking = n-marking, MK-VAL 1 [CARDINAL absent]]
"carros"
PP
"da Ana"

The types subsumed by *no-det-marking* are not used to constrain the MARKING of verbal projections, because the items where they are employed only attach to nouns. This means that when a PP attaches to a verbal constituent, the MARKING value of that constituent remains *basic-marking*. The result is never *n-marking*. This way, features under MK-VAL like CARDINAL are not present in verb headed elements. Consider the analysis for the VP "saíram com a Ana" (*left with Ana*):


VP
MARKING [basic-marking, MK-VAL 1 mk-val-min]
VP
MARKING [basic-marking, MK-VAL 1]
"saíram"
PP
"com a Ana"

The general type *basic-marking* helps in hiding the implementation of NP structure. The information about the exact position within the NP where PPs and AdvPs attach — namely the constraints on the absence of the items in Position IV — is encapsulated in a more specific type, *n-marking*. This more specific type is kept separate from the definitions of prepositions and adverbs.

8.6.10 Relative Clauses

In LXGram, relative clauses are not headed by a verb, and a dedicated head type is used for them (*relative-complementizer*), i.e. constructions that project relative pronouns to the left of a clause are assumed to be non-headed structures. This is compatible with the LinGO Grammar Matrix. We abstain from developing on the analysis of relative clauses here, as their implementation is still experimental.

There is a type in the hierarchy under *marking*, *rel-marking*, used to model relative clause attachment.

Restrictive relative clauses should be allowed to iterate, but they are more peripheral than APs and PPs inside an NP, and they always follow the noun:

$$\begin{bmatrix} \textit{relative-complementizer} & \\ \text{MARKER} & \begin{bmatrix} \textit{post-only-marker} & \\ \text{SELECT}|\text{LOCAL}|\text{CAT} & \begin{bmatrix} \text{HEAD} & \textit{noun} \\ \text{MARKING} & \begin{bmatrix} \textit{no-det-marking} \\ \text{MK-VAL } \boxed{1} \end{bmatrix} \end{bmatrix} \\ \text{MARK} & \begin{bmatrix} \textit{rel-marking} \\ \text{MK-VAL } \boxed{1} \end{bmatrix} & \end{bmatrix} \end{bmatrix}$$

The type hierarchy under *marking* and the constraints presented so far mean that restrictive relative clauses outscope prenominal adjectives. Semantically, this is borne out by the data. Consider the NP in (96) below. It describes an entity as being in fact Chinese. That is, the piece of semantics for that NP will be equivalent to $\lambda P.|((D_e - D) \cap C) \cap P| > 0$ (where D_e is the model's domain, D the set of doctors and C the set of Chinese entities, giving the adjective "falso" the semantics $\lambda Q.D_e - Q$), with no mismatch between syntactic and semantic scope. It will not be $\lambda P.|(D_e - (D \cap C)) \cap P| > 0$, the semantics for the example in (83) ("um falso médico chinês" — *a false Chinese doctor*).

(96) um falso médico que é chinês
a fake doctor who is Chinese
a fake doctor that is Chinese

Semantically, restrictive relative clauses are under the scope of cardinals. Consider the sentence in (97). If M is the set of movies, BM is the set of bad movies and S is the set of entities that "I saw there", the meaning of (97) is $M \cap S \subseteq BM \wedge |M \cap S| = 3$, not $M \cap S \subseteq BM \wedge |M| = 3$.

(97) Todos os exactamente três filmes que lá vi eram maus.
all the exactly three movies that there I saw were bad
All exactly three movies I saw there were bad.

Under the assumption that, if there is no reason to assume the contrary, syntactic scope matches semantic scope, we therefore want relative clauses to attach lower than cardinals. Similar data can be envisaged for ordinals, but we will not present them for the sake of brevity. Semantic considerations cannot help us determine the relative scope between relative clauses and markers of indefinite specifics, because the latter contribute no relations to the resulting semantics. Since cardinals, ordinals and markers of indefinite specifics all occupy the same NP slot, we assume that relative clauses attach lower than all these elements. To force this attachment, we can simply add the following constraints to the type *rel-marking*:

$$\begin{bmatrix} \textit{rel-marking} & \\ \text{MK-VAL} & \begin{bmatrix} \text{CARDINAL} & \textit{absent} \\ \text{ORDINAL} & \textit{absent} \\ \text{INDEF-SPEC} & \textit{absent} \\ \text{TAL} & \textit{absent} \\ \text{POSSESSIVE} & \textit{absent-or-posthead-present} \\ \text{QUALQUER} & \textit{absent-or-posthead-present} \\ \text{OUTRO} & \textit{absent-or-posthead-present} \end{bmatrix} \end{bmatrix}$$

Recall that the feature MK-VAL is unified between MARKER|MARK and MARKER|SELECT|LOCAL|CAT|MARKI
under the HEAD attribute of relative clauses, so the constraints just presented on *rel-marking* effectively make relative clauses select for constituents that lack all of these elements.

The constraint on the feature POSSESSIVE also makes prenominal possessives outscope relative clauses, since prenominal possessives precede cardinals. The constraint on the feature TAL is because the element “tal” (*such*) must also precede cardinals.

It is worth mentioning that there are data that this analysis does not contemplate:

(98)

a	sugestão$_N$	[$_{\text{RelCl}}$ que	foi	mencionada]	[$_{\text{COMP}}$ de	que	seria	impossível
the	suggestion	that	was	mentioned		that	it would be	impossible

proceder	de outro modo	]
to act	in a different way	

the suggestion that was mentioned that it would be impossible to act in a different way

In (98) there is a relative clause, bracketed with RelCl, intervening between the head noun and its sentential complement, bracketed with COMP, for complement. However, according to our system of constraints, relative clauses must attach to projections with already saturated complements. This is because Head-Complement constructions constrain the head daughter to have MARKING of type *basic-marking*, as presented in Section 8.6.8. This constraint was necessary in order to force PP complements to precede relative clauses, as explained in that section.

We believe that this sort of situation only arises in specific cases (sentential complements) and that they should receive a special treatment, which we do not develop here.

8.6.11 Postnominal Demonstratives and Postnominal Possessives

In Position VIII we can find other elements besides adverbial PPs, AdvPs and APs and complements, which have been covered. These other elements are postnominal demonstratives, possessives and universal quantifiers:

(99) a.

A	bicicleta	essa	está	estragada.
the	bicycle	that	is	broken

That bicycle is broken.

b.

Chegaram	várias	cartas	tuas.
arrived	several	letters	yours

There arrived several letters of yours.

c.

Desapareceram	as	cartas	todas.
disappeared	the	letters	all

All the letters disappeared.

We will not address postnominal universal quantifiers. They motivate a more complicated composition of semantics, because they introduce a quantifier relation with scope over the rest of the semantic material of the NP they are in and yet occur in a position (Position VIII) that does not have syntactic scope over that material. Consider the following example, bracketed according to the syntactic structure that is assumed:

(100)	Desapareceram [	as	[[cartas	todas]	da	Ana.]]
	disappeared	the	letters	all	of the	Ana

All of Ana's letters disappeared.

We leave this issue to future work.

8.6.11.1 Postnominal Demonstratives

Postnominal demonstratives are possible in some dialects of Portuguese. They are confined to NPs introduced by a definite article. They do not co-occur with prenominal demonstratives and do not iterate:

(101) a.	A	bicicleta	essa	está	estragada.
	the	bicycle	that	is	broken

That bicycle is broken.

b.	* Uma	bicicleta	essa	está	estragada.
	a	bicycle	that	is	broken

c.	* Essa	bicicleta	essa	está	estragada.
	that	bicycle	that	is	broken

d.	* Esta	bicicleta	essa	está	estragada.
	this	bicycle	that	is	broken

e.	* A	bicicleta	essa	essa	está	estragada.
	the	bicycle	that	that	is	broken

The feature DEMONSTRATIVE under MK-VAL is used to block the co-occurrence of prenominal and postnominal demonstratives and to prevent demonstratives from iterating. This feature is used as it can be expected from the use of the other features of MK-VAL as presented before: prenominal and postnominal demonstratives select for sisters with MARKING|MK-VAL|DEMONSTRATIVE of type *absent* and have MARK|MK-VAL|DEMONSTRATIVE with the value *present*, the remaining functors unify the DEMONSTRATIVE attributes under the paths SELECT|LOCAL|CAT|MARKING|MK-VAL and MARK|MK-VAL. In order to block the co-occurrence of indefinite determiners with postnominal demonstratives, indefinite determiners also select for sisters with an *absent* DEMONSTRATIVE.

Prenominal demonstratives and postnominal demonstratives must come in the lexicon in different entries, or related by lexical rules, because the prenominal ones are determiners and carry quantifier semantics. The constraints on MARKING and MARK are also different between these two sets of items. Prenominal demonstratives have a HEAD of type *determiner*, while postnominal ones must have *marking* constraints almost identical to the other elements in the same slot (postnominal adjectives, etc.). The word order between functor and head is also different. The head of postnominal demonstratives looks like:

$$\left[\begin{array}{ll} \multicolumn{2}{l}{\textit{postnominal-demonstrative}} \\ \text{MARKER} & \left[\begin{array}{ll} \multicolumn{2}{l}{\textit{post-only-marker}} \\ \text{MARK} & \left[\begin{array}{ll} \multicolumn{2}{l}{\textit{n-marking}} \\ \text{MK-VAL} & \left[\begin{array}{ll} \text{DEMONSTRATIVE} & \textit{present} \\ \text{POSSESSIVE} & \boxed{1} \\ \text{QUALQUER} & \boxed{2} \\ \text{TAL} & \boxed{3} \\ \text{OUTRO} & \boxed{4} \\ \text{CARDINAL} & \boxed{5} \\ \text{ORDINAL} & \boxed{6} \\ \text{INDEF-SPEC} & \boxed{7} \end{array}\right] \end{array}\right] \\ \text{SELECT}|\text{LOCAL}|\text{CAT} & \left[\begin{array}{ll} \text{HEAD} & \textit{noun} \\ \text{MARKING} & \left[\begin{array}{ll} \multicolumn{2}{l}{\textit{basic-marking}} \\ \text{MK-VAL} & \left[\begin{array}{ll} \text{DEMONSTRATIVE} & \textit{absent} \\ \text{POSSESSIVE} & \boxed{1} \\ \text{QUALQUER} & \boxed{2} \\ \text{TAL} & \boxed{3} \\ \text{OUTRO} & \boxed{4} \\ \text{CARDINAL} & \boxed{5} \\ \text{ORDINAL} & \boxed{6} \\ \text{INDEF-SPEC} & \boxed{7} \end{array}\right] \end{array}\right] \end{array}\right] \end{array}\right] \end{array}\right]$$

To make the composition of semantics easier with demonstratives, we view determiner demonstratives as carrying two semantic relations: a quantifier relation and an intersective relation in the restrictor of the quantifier. When a demonstrative is used deictically, the second relation can be semantically considered to be roughly similar to the relation of adverbs like *here* or *there*. In this case a noun phrase like *that car* is considered semantically close to a noun phrase like *the car there*, and *this car* to *the car here*. There is some empirical support to this analysis, as the demonstrative and the adverb must agree with respect to deixis:

(102) a. Esta bicicleta aqui está estragada.
this bicycle here is broken
This bicycle here is broken. (the bicycle near me/us)

b. Essa bicicleta aí está estragada.
that bicycle there is broken
That bicycle there is broken. (the bicycle near you)

c. Aquela bicicleta ali está estragada.
that bicycle there is broken
That bicycle there is broken. (the bicycle away from me and you)

d. * Esta bicicleta aí está estragada.

e. * Esta bicicleta ali está estragada.

f. * Essa bicicleta aqui está estragada.

g. * Essa bicicleta ali está estragada.

h. * Aquela bicicleta aqui está estragada.

i. * Aquela bicicleta aí está estragada.

$$\left[\begin{array}{ll}
\textit{mrs} & \\
\text{LTOP} & \boxed{h1}\ h \\
\text{INDEX} & \boxed{e2}\ e \\
\text{RELS} & \left\langle \begin{bmatrix} \textit{\_o\_q\_rel} & \\ \text{LBL} & \boxed{h3}\ h \\ \text{ARG0} & \boxed{x4}\ x \\ \text{RSTR} & \boxed{h6}\ h \\ \text{BODY} & \boxed{h5}\ h \end{bmatrix}, \begin{bmatrix} \textit{\_esse\_a\_rel} & \\ \text{LBL} & \boxed{h7}\ h \\ \text{ARG0} & \boxed{e8}\ e \\ \text{ARG1} & \boxed{x4} \end{bmatrix}, \begin{bmatrix} \textit{\_carro\_n\_rel} & \\ \text{LBL} & \boxed{h7} \\ \text{ARG0} & \boxed{x4} \end{bmatrix}, \begin{bmatrix} \textit{\_avariar\_v\_rel} & \\ \text{LBL} & \boxed{h9} \\ \text{ARG0} & \boxed{e2} \\ \text{ARG1} & \boxed{x4} \end{bmatrix} \right\rangle \\
\text{HCONS} & \left\langle \begin{bmatrix} \textit{qeq} & \\ \text{HARG} & \boxed{h1} \\ \text{LARG} & \boxed{h9} \end{bmatrix}, \begin{bmatrix} \textit{qeq} & \\ \text{HARG} & \boxed{h6} \\ \text{LARG} & \boxed{h7} \end{bmatrix} \right\rangle
\end{array}\right]$$

Figure 8.19: MRS of a sentence with a demonstrative. The corresponding sentences are "esse carro avariou" and "o carro esse avariou" (*that car broke down*).

The names of these predicates in the restrictor of the quantifier are the lemma of the demonstrative — we do not commit to saying that they are identical to adverbial relations. They are obviously not so when demonstratives are employed anaphorically, in which case these predicates are assumed to take a different meaning. We cannot detect automatically with the grammar whether a demonstrative is being used anaphorically or deictically, so the relations visible in the MRSs are meant to be underspecifications.

Postnominal demonstratives introduce a single, plain intersective, relation in the MRS, and the quantifier relation comes from the definite article. Prenominal demonstratives introduce two relations in the MRS representation, a similar intersective one and a quantifier relation. We give them semantics similar to that of a postnominal demonstrative co-occurring with a definite article. That is, the quantifier relation of prenominal demonstratives is the same as that of definite articles, thus treating the following examples as paraphrases of one another:

(103) a.

o	carro	esse
the	car	that

that car

b.

esse	carro
that	car

that car

MRSs for these two cases are shown in Figure 8.19. The LBL of the *_esse_a_rel* relation is the LTOP of the demonstrative determiner's sister: demonstrative determiners unify this LBL with the path SELECT|LOCAL|CONT|HOOK|LTOP under their HEAD.

The attribution of two relations to prenominal demonstratives is also useful in the context of a predeterminer, as in the following example:

(104)

[$_{\text{NP}}$ Todos	esses	carros]	avariaram.
all	those	cars	broke down

All those cars broke down.

In Section 8.6.2 it was stated that determiners co-occurring with predeterminers contribute no quantifier semantics. The fact that demonstratives introduce two relations in the MRS gives us

$$\begin{bmatrix} \textit{mrs} & \\ \text{LTOP} & \boxed{h1}\ h \\ \text{INDEX} & \boxed{e2}\ e \\ \text{RELS} & \left\langle \begin{bmatrix} \textit{\_todo\_q\_rel} & \\ \text{LBL} & \boxed{h3}\ h \\ \text{ARG0} & \boxed{x4}\ x \\ \text{RSTR} & \boxed{h6}\ h \\ \text{BODY} & \boxed{h5}\ h \end{bmatrix}, \begin{bmatrix} \textit{\_esse\_a\_rel} & \\ \text{LBL} & \boxed{h7}\ h \\ \text{ARG0} & \boxed{e8}\ e \\ \text{ARG1} & \boxed{x4} \end{bmatrix}, \begin{bmatrix} \textit{\_carro\_n\_rel} & \\ \text{LBL} & \boxed{h7} \\ \text{ARG0} & \boxed{x4} \end{bmatrix}, \begin{bmatrix} \textit{\_avariar\_v\_rel} & \\ \text{LBL} & \boxed{h9} \\ \text{ARG0} & \boxed{e2} \\ \text{ARG1} & \boxed{x4} \end{bmatrix} \right\rangle \\ \text{HCONS} & \left\langle \begin{bmatrix} \textit{qeq} & \\ \text{HARG} & \boxed{h1} \\ \text{LARG} & \boxed{h9} \end{bmatrix}, \begin{bmatrix} \textit{qeq} & \\ \text{HARG} & \boxed{h6} \\ \text{LARG} & \boxed{h7} \end{bmatrix} \right\rangle \end{bmatrix}$$

Figure 8.20: MRS of a sentence with predeterminer and a demonstrative. The sentence is "todos esses carros avariaram" (*all those cars broke down*).

a simple way to distinguish between semantic representations of sentences with a predeterminer and a demonstrative determiner from semantic representations of sentences with a predeterminer and a definite article. More specifically, there have to be versions of prenominal demonstratives that do not introduce quantifier semantics as well, with *marking* constraints similar to the ones on the vacuous definite articles presented in Section 8.6.2. These determiners are however not semantically vacuous, they still introduce the special predicate in the restrictor of the quantifier. The quantifier relation is introduced by the predeterminer, as before.

In the presence of a predeterminer, NPs with postnominal demonstratives and NPs with prenominal demonstratives have similar MRSs, too. An example MRS is in Figure 8.20. It is the MRS for the sentences "todos esses carros avariaram" and "todos os carros esses avariaram" (*all those cars broke down*).

To control the co-occurrence restrictions between the set of determiners and postnominal demonstratives in (101), the relevant determiners are constrained to select for a sister with DEMONSTRATIVE of type *absent*.

8.6.11.2 Postnominal Possessives

Postnominal possessives are constrained to occur in indefinite NPs or NPs introduced by a demonstrative (Section 8.6.3).

We use the feature POSSESSIVE under MK-VAL to control the distribution of possessives.

The HEAD of possessives has the additional constraints under the POSTHEAD feature (in addition to the constraints presented for this type in Section 8.6.3):

$$\begin{bmatrix} \textit{possessive} & \\ \text{MARKER}|\text{POSHEAD} & \begin{bmatrix} \text{SELECT}|\text{LOCAL}|\text{CAT}|\text{MARKING}\ \textit{n-marking} & \\ \text{MARK} & \begin{bmatrix} \textit{n-marking} & \\ \text{MK-VAL}|\text{POSSESSIVE} & \textit{posthead-present} \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

The co-occurrence of prenominal and postnominal possessives is prevented because they all select for a sister with POSSESSIVE *absent*, they all have MARK|MK-VAL|POSSESSIVE with the value *present*.

The same attributes (the ones under MK-VAL) are also used to constrain the different distribution of prenominal and postnominal possessives, since the types appropriate for these features include information relating to realization and also word order.

Possessives have *posthead-present* under HEAD|MARKER|POSTHEADIMARK|MK-VAL|POSSESSIVE and the value *prehead-present* under HEAD|MARKER|PREHEADIMARK|MK-VAL|POSSESSIVE. Definite articles, which cannot co-occur with postnominal possessives, select for sisters with SYNSEM|LOCAL|CAT|MARKING|MK-VAL|POSSESSIVE *absent-or-prehead-present*. As usual, all functors that are not possessives must percolate this feature by unifying their MARKER|MARK|MK-VAL|POSSESSIVE with their MARKER|SELECT|LOCAL|CAT|MARKING|MK-VAL|POSSESSIVE.

Indefinite determiners select a sister with MARKING|MK-VAL|POSSESSIVE of type *absent-or-posthead-present*, since they cannot co-occur with prenominal possessives. This is also true of the cardinal determiners presented in Section 8.6.5.

Prenominal demonstratives, which can co-occur with both prenominal and postnominal possessives, do not constrain the feature POSSESSIVE.

8.6.12 Gerunds

Gerund forms that occur as modifiers are derived from the lexical entries for verbs via two inflectional rules.

One of these rules produces gerund forms that must modify a noun-headed constituent, the other rule produces gerund forms that must modify a verb-headed constituent.

In the implementation of LXGram, all modifying gerunds are given intersective semantics.

For gerunds that modify nominal constituents, the ARG1 of the gerund's relation (the argument corresponding to the subject of that verb) is the index of the head noun, accessible in the INDEX attribute of the modified element.

The semantics for the example NP below ("um livro descrevendo a Irlanda", *a book describing Ireland*) is thus similar to $\lambda P.\ proper\_q(x,\ named(x,\ "Irlanda"),\ \_um\_q(y,\ \_livro\_n(y) \wedge \_descrever\_v(y,x),\ \ P(y)))$: this NP is given semantics similar to "um livro que descreve a Irlanda" (*a book that describes Ireland*).

(105) um livro descrevendo a Irlanda
a book describing the Ireland
a book describing Ireland

For gerunds that modify verb phrases, the ARG1 feature of the gerund's relation is identified with the index of the modified verb's subject (the subject of the gerund is the subject of the main verb). This value is visible in the XARG feature of the modified constituent. The following example thus receives semantics similar to $\_pronoun\_q(x, pronoun\_n(x), \_sair\_v(e_1, x) \wedge \_correr(e_2, x))$.

(106) a. Saíram correndo.
they left running
They left running.

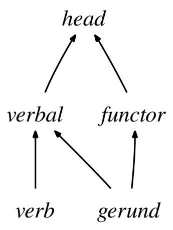
This analysis does not contemplate expressions like "considerando que..." (*considering that...*), where the subject of the gerund form may be independent of the main verb's subject.

Because the value of the ARG1 feature of the gerund's relation is obtained from different attributes of the modified constituent (XARG if it is a verb phrase, INDEX if it is nominal), two separate rules are used for gerunds modifying nouns and verbs.

For gerunds modifying verb phrases, we add an additional relation, called *gerund_rel*, in order to make it explicit which relation corresponds to the gerund. For our previous example, "saíram correndo" (*they left running*), LXGram produces a semantic representation similar to

$\_pronoun\_q(x, pronoun\_n(x), \_sair\_v(e_1, x) \wedge gerund(e_2, e_1) \wedge \_correr\_v(e_2, x))$. The fact that e_2 (the first argument of the *gerund_rel* relation) is the event variable of the *_correr_v_rel* relation indicates that this is the relation corresponding to the gerund.

The HEAD type of gerunds has to be different from the HEAD type of all other verbal forms: gerunds are modifiers (they must have the MARKER feature under head, and therefore must inherit from the type *functor*), whereas non gerund verb forms cannot head modifiers (and so they must not inherit from *functor*). The following simplified type hierarchy presents the relevant types employed in LXGram to control this:



The lexical entries for verbs come with the value *verbal* for the HEAD attribute. This type is specialized as either *gerund* or *verb* in all inflectional rules. The type *gerund* is used in the rules that produce gerund forms, and contains the constraints that control attachment of phrases headed by a gerund form. The type *verb* is used in all other morphological rules. It does not inherit from *functor*, making it impossible for non-gerund verb forms to occur as the functor daughter of head-functor constructions.

The constraints on the features MARKING and MARK of gerunds are similar to the constraints for prepositions and adverbs.

This implementation of gerunds is experimental, and a final solution will be sought in the next development phases.

8.6.13 Other NP Elements with a Special Treatment

LXGram gives a special treatment to other NP elements, namely the elements "outro" (*other*), "qualquer" (*any*) and "tal" (*such*). These elements have a special syntax. They do not fit cleanly in the NP positions identified in Section 8.6. None of these elements can occur multiple times in an NP; therefore there is an attribute under MK-VAL for each one of them, recording their presence (the features OUTRO, QUALQUER and TAL).

The element "outro" can appear in NP initial position, but also in positions that are internal to the NP. In the latter case, its distribution is different from that of adjectives, because it can precede cardinals, for instance. The following sentences illustrate this point.

(107) a. outra bicicleta
another bicycle
another bicycle

b. a outra bicicleta
the other bicycle
the other bicycle

c. as duas outras bicicletsa
the two other bicycles
the two other bicycles

d. as outras duas bicicletas
the other two bicicles

the other two bicicles

The order between “outro” and prenominal possessives is also free:

(108) a. a minha outra bicicleta
the my other bicycle
my other bicycle

b. a outra minha bicicleta
the other my bicycle
my other bicycle

The element “tal” also exhibits special behavior:

(109) a. a tal bicicleta
the such bicycle
that bicycle

b. essa tal bicicleta
that such bicycle
that one bicycle

c. a minha tal bicicleta
the my such bicycle
that bicycle of mine

d. a tal minha bicleta
the such my bicycle
that bicycle of mine

e. a tal bicicleta minha
the such bicycle of mine
that bicycle of mine

f. tal bicicleta minha
such bicycle of mine
that bicycle of mine

g. * tal minha bicicleta
such my bicycle

h. tais duas bicicletas
such two bicycles
two such bicycles

i. duas tais bicicletas
two such bicycles
two such bicycles

j. as tais duas bicicletas
the such two bicycles
those two bicycles

k. as duas tais bicicletas
the two such bicycles
those two bicycles

l. * a tal tal bicicleta
the such such bicycle

The item “qualquer” is also idyossincratic:

(110) a. qualquer bicicleta minha
any bicycle of mine
any bicycle of mine

b. * qualquer qualquer bicicleta minha
any any bicycle of mine

c. quaisquer duas bicicletas
any two bicycles
any two bicycles

d. duas quaisquer bicicletas
two any bicycles
any two bicycles

e. duas bicicletas quaisquer
two bicycles any
any two bicycles

f. * quaisquer bicycletas quaisquer
any bicycles any

g. * as quaisquer bicicletas
the any bicycles

h. * as bicicletas quaisquer
the bicycles any

These features OUTRO, TAL and QUALQUER are used like the other features under MK-VAL: they are unified between MARKING and MARK in the other functors, and constrained with the values *present* or *absent* in these elements. They are used in a way analoguous to the way the other features are used, as presented before, by placing constraints on these feature in the various functors.

Some of these constraints have been presented before. For instance, markers of indefinite specific NPs select for a sister node with the value *absent* for the feature TAL (see Section 8.6.4), because of ungrammatical examples like the following one:

(111) * certas tais bicicletas
certain such bicycles

The item “tal” also selects a sister with INDEF-SPEC of type *absent*, in order to block the following example:

(112) * tais certas bicicletas
such certain bicycles

8.6.14 Postponed Coverage or Known Limitations

Partitives are not contemplated by the current implementation.

LXGram contains no implementation yet of noun-noun compounds, like “palavra-chave” (*key-word*), “governo fantoche” (lit. *puppet government*), “experiência piloto” (*pilot experiment*).

8.7 Noun Ellipsis

In some NPs there is no overt head noun. Some examples are in (113).

(113) a. a casa azul e [a - verde]
the house blue and the green
the blue house and the green one

b. algumas crianças com chapéu e [algumas - com boné]
some children with hat and some with cap
some children in hats and some in caps

c. Os que podem ajudar nunca ajudam.
the who can help never help
The people who can help never do so.

The examples in (113a) and (113b) are however very different from the one in (113c) with respect to the semantics of the missing nouns. In (113a) the noun form "casa" (*house*) is recoverable from context, but in (113c) the missing noun (something close to *people*) is independent of context and its semantics can be described as generic.

For this reason, the phenomenon in (113c) has been referred to as *people* deletion [Pullum, 1975]) or null-N generics [Nerbonne and Mullen, 2000]. Here we will adopt the designation *missing-N generics*, in order to remain neutral to the status of non-realized elements.

On the other hand, the phenomenon in (113a) is known as noun ellipsis.

In LXGram these constructions are handled via unary rules that produce a mother node with a HEAD feature with the type *noun*. The daughter of these rules is any element that selects for a noun headed constituent via their SELECT feature. Since we cannot distinguish between missing noun generics and noun ellipsis on the basis of syntax, this construction adds a relation to the MRS with the name *ellipsis-or-generic_n_rel*.

Implementation details are provided in the next sections.

8.7.1 ARGS, HEAD-DTR and NON-HEAD-DTR

The order of the daughters of phrasal constituents is denoted in the LKB by the order of elements in the list-valued feature ARGS. Furthermore, attributes like HEAD-DTR and NON-HEAD-DTR are merely pointers to these elements, useful when one wants to abstract from word order.

Nothing requires that these daughter features point to an existing element of ARGS, though. That is, it is possible to have constructions with the two features, HEAD-DTR and NON-HEAD-DTR, but with an ARGS list of less than two elements.

This is a way to model a class of missing syntactic constituents. Assuming that phrases are binary at most (this is enforced in LXGram and several other computational HPSGs), these constructions are prototypically unary, but have semantic or syntactic properties of some other binary constructions.

The difference between ARGS and daughter features (HEAD-DTR and NON-HEAD-DTR) has no theoretical status in HPSG, and the attribute ARGS is specific to the LKB. But we can make a conceptual distinction between them, and give them a theoretical status. The feature ARGS denotes the realized daughters of a phrase, whereas the daughter features (like HEAD-DTR and NON-HEAD-DTR) include them as well as elements that correspond to empty constituents.

There are thus two dimensions: the HEAD-DTR and NON-HEAD-DTR level, which abstracts from the possibility of non-realized constituents, and the ARGS level, which is more superficial in this respect. ARGS is also the best place where the Principle of Canonicality can be enforced (all elements of ARGS are required to have SYNSEMs of type *canonical-synsem*). This is what is done in LXGram (see Section 8.6.1).

The implementation of the missing noun phrases is an interesting case to justify these two levels. It is explained in Section 8.7.2.

8.7.2 Implementation

The phrase types for missing noun constructions are descendants of *basic-missing-noun-phrase* where both the HEAD-DTR and the NON-HEAD-DTR features are present but ARGS has a single element in it, corresponding to the non-head daughter. The type *basic-missing-noun-phrase* is a subtype of *basic-head-functor-phrase*, where several constraints for the Head-Functor constructions are stated. These constraints are inherited by *basic-missing-noun-phrase.* They have been presented in Section 5.3.

In the type *basic-missing-noun-phrase* it must be specified which daughter is realized. We chose to do it in a supertype, *head-missing*, assuming that there can be other constructions with a singleton ARGS but with both HEAD-DTR and NON-HEAD-DTR features, which could be defined to also inherit from *head-missing*:

$$\begin{bmatrix} \textit{head-missing} \\ \text{HEAD-DTR}|\text{SYNSEM } \textit{non-canonical-synsem} \\ \text{NON-HEAD-DTR } \boxed{1} \\ \text{ARGS } \left\langle \boxed{1} \right\rangle \end{bmatrix}$$

The attribute SYNSEM of the missing daughter is constrained to be of the type *non-canonical-synsem*, in view of the fact that it is not realized.

Its counterpart type, *non-head-missing*, is also part of the hierarchy of phrse types and is specified to have the expected constraints, namely:

$$\begin{bmatrix} \textit{non-head-missing} \\ \text{HEAD-DTR } \boxed{1} \\ \text{NON-HEAD-DTR}|\text{SYNSEM } \textit{non-canonical-synsem} \\ \text{ARGS } \left\langle \boxed{1} \right\rangle \end{bmatrix}$$

The interesting part of this design is that, in order to add noun semantics and constrain the type of the HEAD and MARKING features (to be *noun* and *n-marking* respectively) that the functor feeding the *basic-missing-noun-phrase* will see under its SELECT feature, one can put this information under the HEAD-DTR attribute. The basic machinery put in place to percolate syntactic information from the daughters in headed phrases and Head-Functor schemata fills the appropriate values in the mother node — it is completely inherited from supertypes. All that is required is that the supertypes never constrain ARGS, and use HEAD-DTR and NON-HEAD-DTR instead.

This approach is taken even further. Since the constraints on the HEAD-DTR feature effectively consist of the definition of a noun, HEAD-DTR can simply be constrained to be of a type that is a supertype of lexical items for nouns.

In LXGram there is a type, *noun-common-0comps-3p-sign*, that is the supertype of all lexical types for count nouns with no complements that do not accept second-person readings (see Section 8.3). It includes constraints that determine the SYNSEM|LOCAL|CAT|HEAD feature to be of type *common-noun* (see Section 8.1) and the attribute SYNSEM|LOCAL|CAT|MARKING must have a value that denotes that this element is not a full NP. Under SYNSEM|LOCAL|CONT, HCONS is empty and RELS includes a single relation with an ARG0 of type *ref-index* (the real type name of the variables that show up in MRSs with type *x*) structure-shared with HOOK|SARG (see Section 5.11) and HOOK|LTOP is unified with the LBL of that relation:

$$\begin{bmatrix} \textit{noun-common-0comps-3p-sign} & \\ \text{SYNSEM}|\text{LOCAL} & \begin{bmatrix} \text{CAT} & \begin{bmatrix} \text{HEAD} & \begin{bmatrix} \textit{common-noun} & \\ \text{AGR}|\text{PNG}|\text{PERSON} & \textit{3rd} \end{bmatrix} \\ \text{VAL}|\text{COMPS} & \langle\rangle \\ \text{MARKING} & \begin{bmatrix} \textit{non-saturated-marking} & \\ \text{DEMONSTRATIVE} & \textit{absent} \\ \text{POSSESSIVE} & \textit{absent} \\ \text{TAL} & \textit{absent} \\ \text{QUALQUER} & \textit{absent} \\ \text{OUTRO} & \textit{absent} \\ \text{CARDINAL} & \textit{absent} \\ \text{ORDINAL} & \textit{absent} \\ \text{INDEF-SPEC} & \textit{absent} \end{bmatrix} \end{bmatrix} \\ \text{CONT} & \begin{bmatrix} \text{HOOK} & \begin{bmatrix} \text{LTOP} & \boxed{1} \\ \text{SARG} & \boxed{2} \end{bmatrix} \\ \text{RELS} & \left\{ \begin{bmatrix} \text{LBL} & \boxed{1} & \textit{handle} \\ \text{ARG0} & \boxed{2} & \textit{ref-index} \end{bmatrix} \right\} \\ \text{HCONS} & \{\} \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

The value of the feature MARKING in this type is explained in Section 8.7.3. The lexical types that inherit from *noun-common-0comps-3p-sign* and correspond to overt nouns further constrain this feature with the value *n-marking*, as presented above.

A descendant of *noun-sign* is *covert-noun-sign*, representing a noun that has no phonetic realization. The semantics specific to missing noun constructions (LXGram does not resolve the antecedent of noun ellipsis) is specified here:

$$\begin{bmatrix} \textit{covert-noun-sign} & \\ \text{SYNSEM} & \begin{bmatrix} \textit{unexpressed-synsem} & \\ \text{LOCAL} & \begin{bmatrix} \text{CONT}|\text{RELS} & \left\{ \begin{bmatrix} \text{PRED} & \textit{ellipsis-or-generic\_n\_1\_rel} \end{bmatrix} \right\} \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

The constraint on the type of its SYNSEM attribute (type *unexpressed-synsem*) denotes the fact that this noun is not realized.

This lexical type is not used in lexical entries, since our analysis does not resort to null constituents, but it can be used in the definition of the constructions with missing nouns. These constructions specify their head daughter to be a *covert-noun-sign*:

$$\begin{bmatrix} \textit{basic-missing-noun-phrase} & \\ \text{HEAD-DTR} & \textit{covert-noun-sign} \end{bmatrix}$$

All the properties specific to *basic-missing-noun-phrase* follow immediately from the constraints inherited from its supertypes and the constraints on *covert-noun-sign*, and do not have to be stated as specific constraints in the *basic-missing-noun-phrase* type.

In particular, the daughter of this construction must be an element that selects for a noun headed constituent, since the constraints inherited from *basic-head-functor-phrase* unify the SELECT feature of the non-head daughter with the SYNSEM under HEAD-DTR.

The MARK feature of the non-head daughter is also unified with the MARKING of the mother node via the inherited constraints. Therefore, if a determiner feeds this rule, a full NP is produced, but if e.g. an adjective is the daughter then the resulting node must combine with another element before forming a full NP.

The HEAD feature of the mother node is also constrained to be of the type *common-noun*, since it is unified with the HEAD feature of the HEAD-DTR in supertypes.

The advantages of this implementation are:

- No *ad hoc* constraints on missing noun phrases are needed to add noun semantics or to constrain the value of the HEAD feature of the mother node or of the SELECT feature of the functor daughter. The constraints necessary to compose semantics are also inherited from very general supertypes.

- The constraints common to overt nouns and the missing head are stated in a single place. Furthermore, these constraints basically define what a noun is. This makes it easier to change the implementation. For instance, changes in the type hierarchy of *marking* that require changes in the value of MARKING of nouns do not require changes both in the lexical types of nouns and in the definition of missing noun phrases.

- The constraints that define what a noun is are encapsulated in the type used to constrain the HEAD-DTR feature and not directly stated in the type for missing noun phrases.

The main disadvantage is that the feature structures for missing noun phrases will be substantially larger, since the feature structure for an entire lexical item will be present under HEAD-DTR. Note however that it does not imply more unification operations at run time, since no node will be unified with the entire HEAD-DTR attribute, as it is not an element of ARGS.

8.7.3 Predeterminers in Missing Noun Constructions

In LXGram all nouns come in the lexicon with the value *n-marking* for the feature MARKING. However, in the syntax rules for missing nouns, the daughter of those rules is constrained to be a functor selecting for a noun-headed constituent with MARKING of type *non-saturated-marking* (as presented above), rather than *n-marking*. The reason for this mismatch is the peculiar behavior of the predeterminer "todos" (*all*). This element can appear in noun ellipsis (114a) or missing-N generic constructions (114b).

(114) a. O João comprou maçãs e todas estavam podres.
the João bought apples and all were rotten
João bought apples and all (of them) were rotten.

b. Todos são livres.
all are free
All (people) are free.

When this element appears in NPs with an overt head, a determiner must also be present in the case of European Portuguese, as reported in Section 8.6.2. The analysis presented in that section resorted to two lexical entries for this item, one of them common to European and Brazilian Portuguese, where the presence of a determiner is required, and another specific to Brazilian Portuguese, where it is not.

For the first entry, a lexical specification was employed in predeterminers according to which they select for a nominal projection with a value of MARKING subsumed by *non-saturated-det-marking.* This constraint makes it incompatible with constituents that bear the value *n-marking* for this attribute, but it is compatible with the value *non-saturated-marking.* This is the reason why the value of MARKING of missing nouns is more abstract than that of overt nouns.

The second entry, specific to Brazilian Portuguese, is also allowed in the missing noun constructions, but sentences like the ones in (114) are possible in European Portuguese, too. Therefore, we would like the item exclusive to Brazilian Portuguese to be blocked, since that would just multiply parses. The item exclusive to Brazilian Portuguese is constrained to select for a constituent with a synsem of type *canonical-synsem.* This prevents it from feeding the rules for missing nouns, since the HEAD-DTR of these constructions has a SYNSEM of type *unexpressed-synsem* (see the constraints on *covert-noun-sign* above), that is incompatible with *canonical-synsem.*

8.8 Bare NPs

The implementation of bare NPs (NPs lacking a determiner) is very similar to the implementation of constructions with missing nouns. Some examples of bare NPs are given below.

(115) a. Desapareceram livros da biblioteca.
disappeared books from the library
There disappeared books from the library.

b. Compraram novos livros para a biblioteca.
they bought new books for the library
They bought new books for the library.

c. ??/* Livros desapareceram da biblioteca.
books disappeared from the library

As the last example shows, bare NPs are not generally acceptable as preverbal subjects. Currently, this possibility is not blocked by LXGram.

In LXGram, bare NPs are implemented as inheriting from a type for Head-Functor constructions and also from *non-head-missing* (see the definition of this type in Section 8.7.2). The missing functor is constrained to be a determiner, in a way similar to the way that the head daughter of missing noun constructions is constrained to be a noun. The head daughter of bare-NP constructions is therefore automatically constrained to be headed by a noun and bear an appropriate value for the feature MARKING. Also, the semantics of a quantifier relation that this rule must introduce comes from the constraints on the missing daughter. There are additional constraints on the features under MK-VAL of the head daughter, which have been stated in Section 8.6.5.

8.9 Postponed Coverage or Known Limitations

The implementation of appositive modification is postponed to a phase after punctuation has been implemented more substantially, since appositive modifiers are generally written between commas.

Chapter 9

Syntax Rules

We present the phrase-structure rules in LXGram relevant in the current implementation phase. They are defined in the files `syntax.tdl` and `rules.tdl`.

- Root
  Most root nodes of syntax trees are produced by this rule. This is a unary rule whose daughter is constrained to be headed by a verb and have saturated valence lists. This rule is responsible for adding a *qeq* constraint between the global top and the handle of the main verb. In LXGram complementizers also select for a clausal complement and introduce a similar handle constraint in the MRS representation. For this reason, we use a feature root-comp whose value is a lexical type for complementizers: the relevant attributes are then simply unified (for instance, the constraints on the comps element under root-comp are unified with the synsem of the daughter of this rule).

  See Figure 9.1.

Ex.:	Choveu.
	rained.THIRD-PERSON-SINGULAR
	It rained.

- Pre_Root-Sentence
  This rule rewrites a start symbol (a sentence) as a pre-root element and a start symbol. The pre-root element can be an interjection, a greeting element ("olá" - *hi*) or a kind of adverbial ("bom" - *well*). This rule can iterate if there are multiple pre-root elements. Currently, the pre-root element is constrained to have a comma attached to it.

  See Figure 9.2.

Ex.:	Olá,	estás	bem?
	hi	are.SECOND-PERSON-SINGULAR	okay
	Hi, are you okay?		

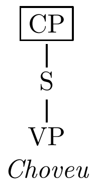


Figure 9.1: The Root syntactic rule (corresponding node(s) in a box).

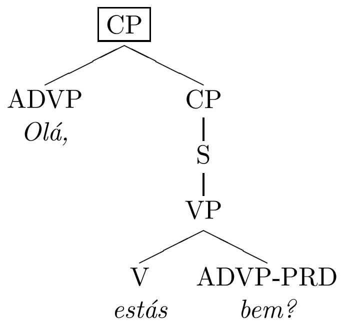


Figure 9.2: The PRE_ROOT-SENTENCE syntactic rule (corresponding node(s) in a box).

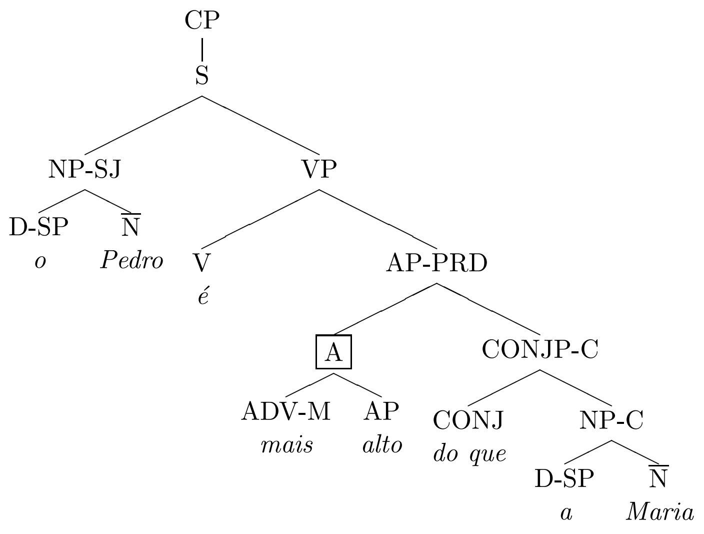


Figure 9.3: The FUNCTOR-HEAD-FCOMPS-ISECT syntactic rule (corresponding node(s) in a box).

- FUNCTOR-HEAD-FCOMPS-ISECT
  This rule projects a functor to the left of its head and unifies the COMPS of the resulting node with those of the functor. See Section 5.3 for details. This rule applies to functors that are given intersective semantics and unifies the LTOP features of both daughters with that of the mother node (see Section 5.3.1). The scopal version of this rule is FUNCTOR-HEAD-FCOMPS-SCOPAL, which only unifies the LTOP of the mother with that of the functor daughter.

  See Figure 9.3.

  Ex.: O Pedro é mais alto do que a Maria.
  the Pedro is more tall than the Maria
  Pedro is taller than Maria.

- FUNCTOR-HEAD-HCOMPS-ISECT
  This rule projects a functor to the left of its head and unifies the COMPS of the resulting node with those of the head. See Section 5.3 for details. This rule applies to functors that are given intersective semantics and unifies the LTOP features of both daughters with that of the mother node (see Section 5.3.1). The scopal version of this rule is FUNCTOR-HEAD-HCOMPS-SCOPAL, which only unifies the LTOP of the mother with that of the functor daughter.

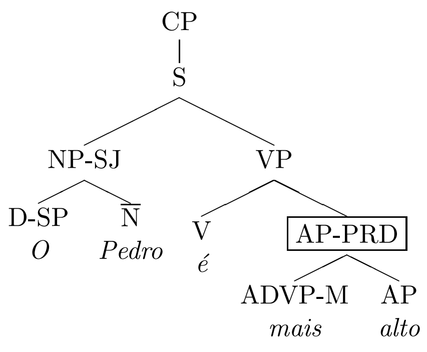


Figure 9.4: The FUNCTOR-HEAD-HCOMPS-ISECT syntactic rule (corresponding node(s) in a box).

See Figure 9.4.

Ex.:	O	Pedro	é	mais	alto.
	the	Pedro	is	more	tall

Pedro is taller.

- HEAD-FUNCTOR-ISECT
  This rule projects a functor to the right of its head and unifies the COMPS of the resulting node with those of the head. See Section 5.3 for details. This rule applies to functors that are given intersective semantics and unifies the LTOP features of both daughters with that of the mother node (see Section 5.3.1). The scopal version of this rule is HEAD-FUNCTOR-SCOPAL, which only unifies the LTOP of the mother with that of the functor daughter.

  See Figure 9.5.

Ex.:	O	Pedro	era	então	um	consumidor	compulsivo	de	cinema.
	the	Pedro	was	then	a	consumer	compulsive	of	cinema

At that time Pedro was a compulsive consumer of movies.

- SUBJECT-HEAD
  This rule is responsible for producing pre-verbal subjects, saturating the verb's SUBJ valence. It will project whichever type of subject a verb selects for. It inherits from *basic-head-subj-phrase* and *head-final*.

  See Figure 9.6.

Ex.:	O	carro	da	Maria	é	amarelo.
	the	car	of the	Maria	is	yellow

Maria's car is yellow.

- HEAD-SUBJECT
  This rule is responsible for rewriting a finite sentence as a subject and a VP. Currently the subject can only be an NP, because verbs selecting for a different kind of subject are not implemented yet, but the rule will rewrite as the the left daughter any kind of element in the head's SUBJ list. It inherits from *basic-head-subj-phrase* in the LinGO Grammar Matrix.

  See Figure 9.7.

Ex.:	Chegou	o	pai	da	Maria.
	arrived	the	father	of the	Maria

Maria's father arrived.

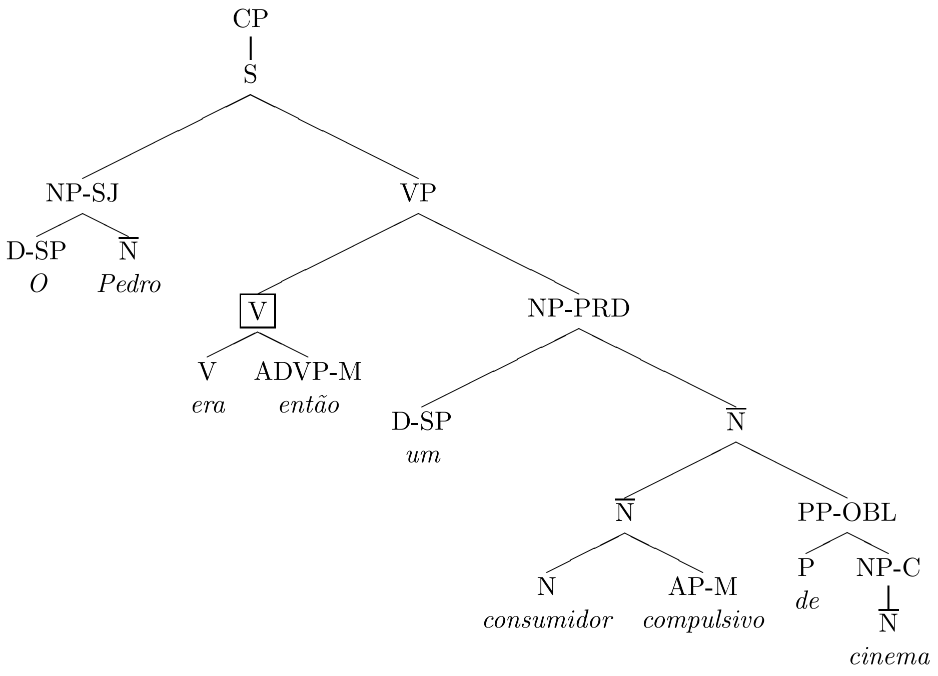


Figure 9.5: The HEAD-FUNCTOR-ISECT syntactic rule (corresponding node(s) in a box).

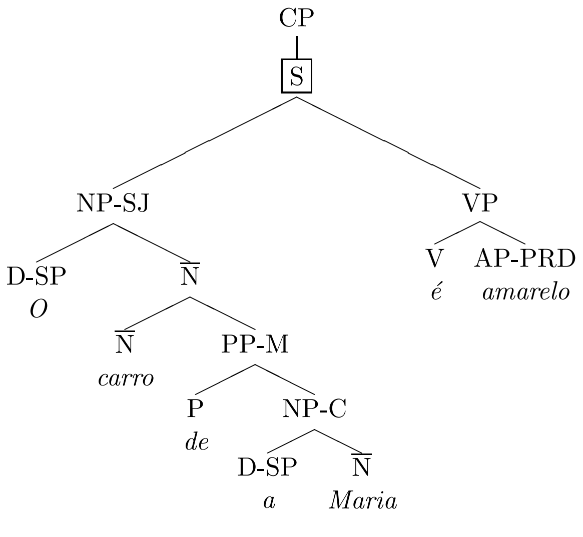


Figure 9.6: The SUBJECT-HEAD syntactic rule (corresponding node(s) in a box).

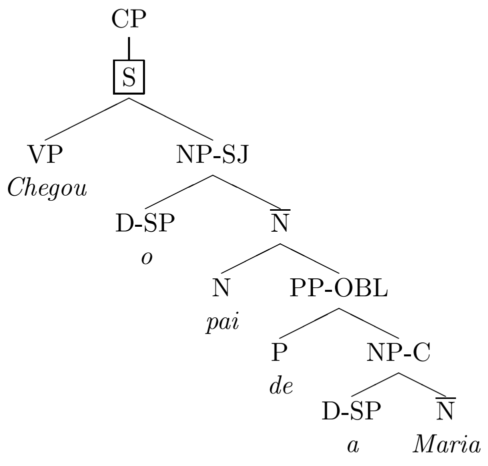


Figure 9.7: The HEAD-SUBJECT syntactic rule (corresponding node(s) in a box).

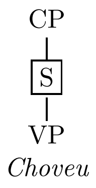


Figure 9.8: The NULL_EXPLETIVE_SUBJECT syntactic rule (corresponding node(s) in a box).

- NULL_EXPLETIVE_SUBJECT
  This rule is responsible for discharging an expletive subject from a verb's valence in the cases where the expletive ("ele") is not realized.

  See Figure 9.8.

  Ex.: Choveu.
  rained.THIRD-PERSON-SINGULAR
  It rained.

- NULL_SUBJECT-HEAD
  This rule is responsible for producing non-expletive null subjects. It empties the SUBJ list of a verb and it also adds the semantics of a personal pronoun linked to the argument in the verbal relation that corresponds to the subject. This personal pronoun's PNG features are specified only with the information provided by the inflection for subject agreement on the verb.

  See Figure 9.9.

  Ex.: Avariaram.
  broke down.THIRD-PERSON-PLURAL
  They broke down.

- HEAD-COMP_NOTCLITIC
  This is a binary rule projecting the first complement of any head, and passing the remaining complements up to the mother node. It only applies to non-clitic complements. Currently, there are rules for clitics implemented in LXGram. They are not listed here, because the current implementation of clitics is still experimental.

  See Figure 9.10.

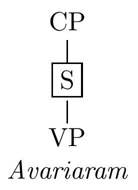


Figure 9.9: The NULL_SUBJECT-HEAD syntactic rule (corresponding node(s) in a box).

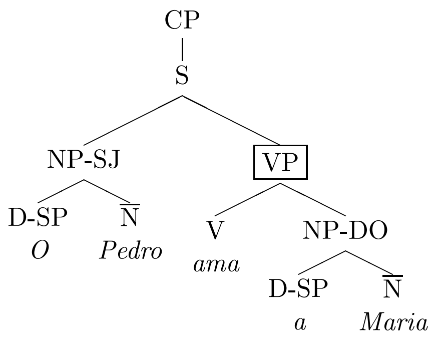


Figure 9.10: The HEAD-COMP_NOTCLITIC syntactic rule (corresponding node(s) in a box).

Ex.:	O	Pedro	ama	a	Maria.
	the	Pedro	loves	the	Maria

Pedro loves Maria.

- BARE-NP
  This rule produces bare NPs. See Section 8.8

  See Figure 9.11.

Ex.:	Ele	bebe	vinho.
	he	drinks	wine

  He drinks wine.

- MISSING-NOUN-HEAD-INITIAL-ISECT
  This unary rule produces noun-headed projections from a constituent that can modify a noun. There are four such rules in LXGram: MISSING-NOUN-HEAD-FINAL-ISECT, MISSING-NOUN-HEAD-FINAL-SCOPAL, MISSING-NOUN-HEAD-INITIAL-ISECT, MISSING-NOUN-HEAD-INITIAL-SCOPAL. The first two rules correspond to the cases when the missing noun would follow the modifier that feeds the rule; the other two are for the cases when the miss-

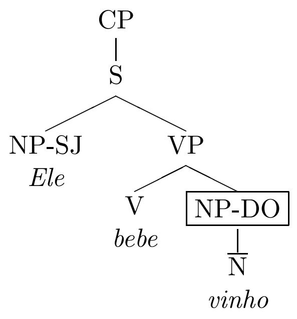


Figure 9.11: The BARE-NP syntactic rule (corresponding node(s) in a box).

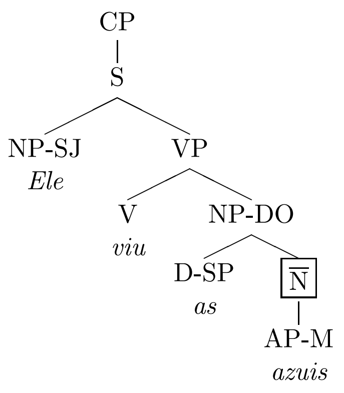


Figure 9.12: The Missing-Noun-Head-Initial-Isect syntactic rule (corresponding node(s) in a box).

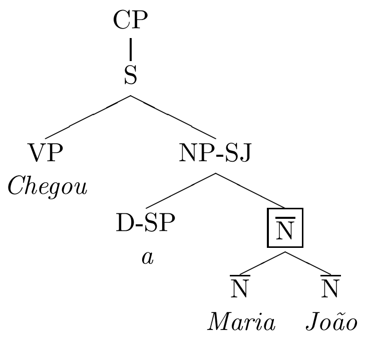


Figure 9.13: The Name-Name syntactic rule (corresponding node(s) in a box).

ing noun would precede it. There are two rules for intersective modifiers and two rules for scopal modifiers and specifiers, just like for the Head-Functor rules above. See Section 8.7

See Figure 9.12.

Ex.:	Ele	viu	as	azuis.
	he	saw	the	blue
	He saw the blue ones			

- Name-Name

  This binary rule takes a proper name as its left daughter and either a proper name or another instance of the name-name construction as its right daughter. See Section 8.5.3

  See Figure 9.13.

Ex.:	Chegou	a	Maria João
	arrived	the	Maria João
	Maria João arrived.		

- NBar-Name

  This binary rule takes a constituent headed by a common noun as its left daughter and an element headed by a proper name as its right daughter. It combines the semantics of the two elements in an intersective way.

  See Figure 9.14.

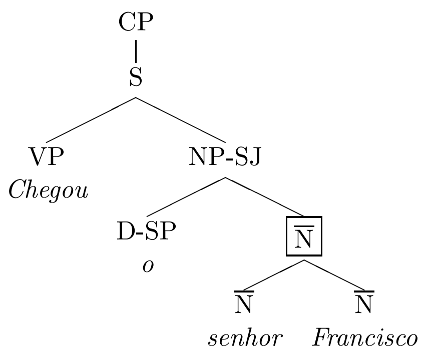


Figure 9.14: The NBar-Name syntactic rule (corresponding node(s) in a box).

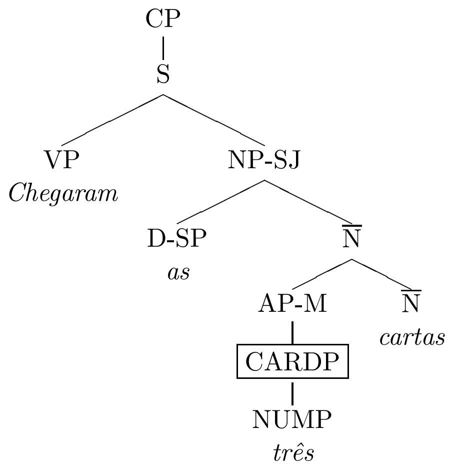


Figure 9.15: The Null-Precardinal syntactic rule (corresponding node(s) in a box).

Ex.:	Chegou	o	senhor	Francisco
	arrived	the	Mr.	Francisco

Mr. Francisco arrived.

- Null-Precardinal
  This rule takes as daughter a cardinal word and adds *at least* semantics to that cardinal (in the absence of cardinal modifiers like "exactamente"/*exactly*, etc.). The mother has a different head type, so that cardinal modifiers can no longer attach. See Section 8.6.5.1.

  See Figure 9.15.

Ex.:	Chegaram	as	três	cartas.
	arrived	the	three	letters

  The three letters arrived.

- Number-Expression-To-Cardinal
  This rule takes as daughter a cardinal word (possibly modified by cardinal modifiers) and adds on the mother node constraints that allow it to modify a noun-headed element as a post-determiner. It adds the *cardinal_rel* relation to the semantics. See Section 8.6.5.1.

  See Figure 9.16.

Ex.:	Chegaram	as	três	cartas.
	arrived	the	three	letters

  The three letters arrived.

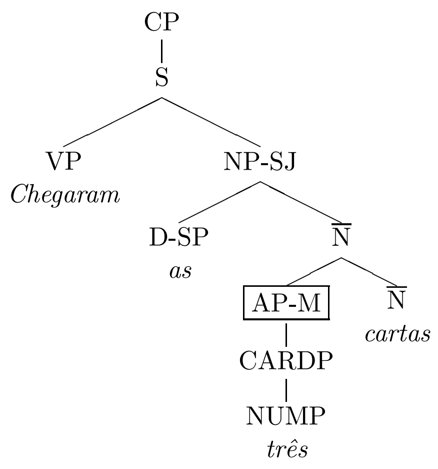


Figure 9.16: The Number-Expression-To-Cardinal syntactic rule (corresponding node(s) in a box).

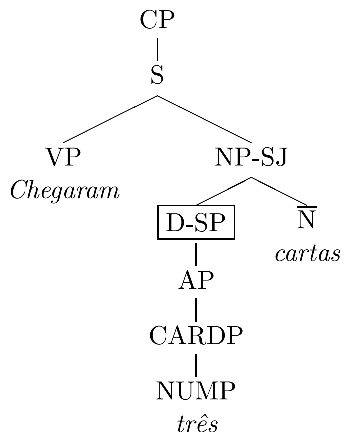


Figure 9.17: The Cardinal-to-Determiner syntactic rule (corresponding node(s) in a box).

- Cardinal-to-Determiner

  This rule takes as daughter a cardinal post-determiner and, in the absence of a realized determiner, constrains the mother node to attach as a determiner. It adds a quantifier relation to the semantics. See Section 8.6.5.1.

  See Figure 9.17.

  Ex.: Chegaram três cartas.
  arrived three letters
  Three letters arrived.

Chapter 10

Lexical Rules

The lexical rules are defined in the files `irules.tdl` (inflectional rules)and `lrules.tdl` (non-inflectional rules). The file `morphology.tdl` contains the supertypes of the inflectional rules, and `lexical-rules.tdl` is where the supertypes of the non-inflectional lexical rules are defined. The rules for punctuation are defined in the file `punctuation.tdl`. This chapter lists the lexical rules relevant in the current implementation phase.

10.1 Verbal Inflection

In the case of verbal inflection, two sets of rules are applied to recognize fully inflected forms. The first set of rules specifies tense, aspect and mood (TAM) information, the second set is responsible for person and number (PN) agreement with the subject. Verbal forms that do not exhibit subject agreement are produced in a single step, via inflectional rules that produce words (this is controlled via subtypes of *sign*, as all items that can be the daughter of a syntactic rule must be compatible with the type *syntactic-sign*, as presented in Section 8.6.1). Verbal forms that show subject agreement are each produced via a TAM rule that outputs a lexeme (they are of a subtype of *sign* that is incompatible with *syntactic-sign*, namely the type *morphological-sign*) and a PN rule that produces a word.

Some form of underspecification is also employed for ambiguous forms, although this is not done in a fully systematic manner (e.g. third person singular indicative present and second person singular affirmative imperative virtually always exhibit the same surface form, but there is no point in providing a similar, underspecified representation for these, since imperative sentences will have a value for the feature SF different from that of the declarative sentences — see Section 5.4).

LXGram is lacking a rule to produce negative imperative forms (e.g. "corre" – *run* – vs. "não corras" - *don't run*), and the participles used in passives. Only gerunds that are VP or $\overline{\mathrm{N}}$ adjuncts are produced at the moment (see Section 8.6.12).

Morphology is not exhaustively implemented. There are, for instance, verbal inflection paradigms that are not covered. This is an explicit decision, because it is planned that LXGram will be interfaced with external shallow tools which take care of morphology (see Section 6.1 for references).

All TAM rules but the rule for the past participle of compound tenses share the mood value under the verbal event with the feature VFORM of the attribute HEAD. All TAM rules constrain this VFORM attribute appropriately. VFORM is constrained by the elements that select for verbal projections (for instance, verbs selecting for VPs headed by an infinitive form constrain the VFORM of that complement with the value *infinitivo*).

The TAM rules that produce lexemes that are to be further inflected for subject agreement

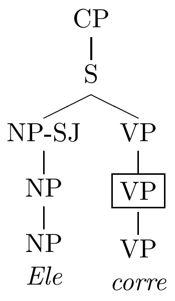


Figure 10.1: The Pres-Ind-Verb lexical rule (corresponding node(s) in a box).

are listed next. They all constrain the feature synsem|local|cont|hook|index|e|aspect|perf with + if the input is a perfect auxiliary and - otherwise (it is also structure shared with the feature alts|simple-tense-perf; see Section 7.1).

- Pres-Ind-Verb
  This rule adds the constraint
  $$\left[\textsc{synsem}|\textsc{local}|\textsc{cont}|\textsc{hook}|\textsc{index}|\textsc{e}\ \begin{bmatrix}\textsc{tense} & \textit{presente}\\ \textsc{mood} & \textit{indicativo}\end{bmatrix}\right]$$
  See Figure 10.1.

  Ex.: Ele corre.
  he runs
  He runs.

- Pret-Imp-Ind-Verb
  This rule adds the constraint
  $$\left[\textsc{synsem}|\textsc{local}|\textsc{cont}|\textsc{hook}|\textsc{index}|\textsc{e}\ \begin{bmatrix}\textsc{tense} & \textit{pretérito-imperfeito}\\ \textsc{mood} & \textit{indicativo}\end{bmatrix}\right]$$

  Ex.: Ele corria.
  he ran
  He ran/would run/used to run.

- Pret-Perf-Ind-Verb
  This rule adds the constraint
  $$\left[\textsc{synsem}|\textsc{local}|\textsc{cont}|\textsc{hook}|\textsc{index}|\textsc{e}\ \begin{bmatrix}\textsc{tense} & \textit{pretérito-perfeito}\\ \textsc{mood} & \textit{indicativo}\end{bmatrix}\right]$$

  Ex.: Ele correu.
  he ran
  He ran/has run.

- Pres-Or-Pret-Perf-Ind-Verb
  This rule adds the constraint
  $$\left[\textsc{synsem}|\textsc{local}|\textsc{cont}|\textsc{hook}|\textsc{index}|\textsc{e}\ \begin{bmatrix}\textsc{tense} & \textit{presente-or-pretérito-perfeito}\\ \textsc{mood} & \textit{indicativo}\end{bmatrix}\right]$$

  Ex.: Nós corremos.
  we run/ran
  We run/ran.

- PLU-IND-VERB
  This rule adds the constraint
  $$\begin{bmatrix}\textsc{synsem}|\textsc{local}|\textsc{cont}|\textsc{hook}|\textsc{index}|\textsc{e} & \begin{bmatrix}\textsc{tense} & \textit{pretérito-mais-que-perfeito}\\ \textsc{mood} & \textit{indicativo}\end{bmatrix}\end{bmatrix}$$
  Ex.: Ele correra.
  he had run
  He had run.

- PRET-PERF-OR-PLUPERF-IND-VERB
  This rule adds the constraint
  $$\begin{bmatrix}\textsc{synsem}|\textsc{local}|\textsc{cont}|\textsc{hook}|\textsc{index}|\textsc{e} & \begin{bmatrix}\textsc{tense} & \textit{pretérito-perfeito-or-pretérito-mais-que-perfeito}\\ \textsc{mood} & \textit{indicativo}\end{bmatrix}\end{bmatrix}$$
  Ex.: Eles correram.
  they ran
  They ran/have run/had run.

- FUT-IND-VERB
  This rule adds the constraint
  $$\begin{bmatrix}\textsc{synsem}|\textsc{local}|\textsc{cont}|\textsc{hook}|\textsc{index}|\textsc{e} & \begin{bmatrix}\textsc{tense} & \textit{futuro}\\ \textsc{mood} & \textit{indicativo}\end{bmatrix}\end{bmatrix}$$
  Ex.: Ele correrá.
  he will run
  He will run.

- FUT-PRET-IND-VERB
  This rule adds the constraint
  $$\begin{bmatrix}\textsc{synsem}|\textsc{local}|\textsc{cont}|\textsc{hook}|\textsc{index}|\textsc{e} & \begin{bmatrix}\textsc{tense} & \textit{futuro-do-pretérito}\\ \textsc{mood} & \textit{indicativo}\end{bmatrix}\end{bmatrix}$$
  Ex.: Ele correria.
  he would run
  He would run.

- PRES-SUBJ-VERB
  This rule adds the constraint
  $$\begin{bmatrix}\textsc{synsem}|\textsc{local}|\textsc{cont}|\textsc{hook}|\textsc{index}|\textsc{e} & \begin{bmatrix}\textsc{tense} & \textit{presente}\\ \textsc{mood} & \textit{conjuntivo}\end{bmatrix}\end{bmatrix}$$
  Ex.: Não tenho nenhum que funcione.
  not I have any that works.PRESENT-SUBJUNCTIVE
  I don't have any that works.

- PRET-IMP-SUBJ-VERB
  This rule adds the constraint
  $$\begin{bmatrix}\textsc{synsem}|\textsc{local}|\textsc{cont}|\textsc{hook}|\textsc{index}|\textsc{e} & \begin{bmatrix}\textsc{tense} & \textit{pretérito-imperfeito}\\ \textsc{mood} & \textit{conjuntivo}\end{bmatrix}\end{bmatrix}$$

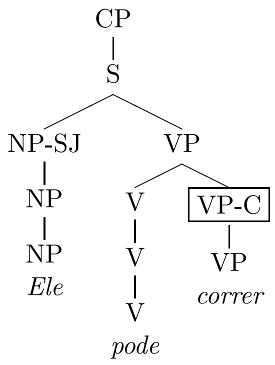


Figure 10.2: The Inf-Verb lexical rule (corresponding node(s) in a box).

Ex.: Não tinha nenhum que funcionasse.
not I had any that worked.IMPERFECT-SUBJUNCTIVE
I didn't have any that worked.

- Fut-Subj-Verb
  This rule adds the constraint
  $$\left[\text{SYNSEM}|\text{LOCAL}|\text{CONT}|\text{HOOK}|\text{INDEX}|\text{E}\ \begin{bmatrix}\text{TENSE} & \textit{futuro}\\ \text{MOOD} & \textit{conjuntivo}\end{bmatrix}\right]$$
  Ex.: Comprarei o primeiro que chegar.
  I will buy the first that arrives.FUTURE-SUBJUNCTIVE
  I'll buy the first one to arrive.

- Pres-Imp-Verb
  This rule adds the constraint
  $$\left[\text{SYNSEM}|\text{LOCAL}|\text{CONT}|\text{HOOK}|\text{INDEX}|\text{E}\ \begin{bmatrix}\text{TENSE} & \textit{presente}\\ \text{MOOD} & \textit{imperativo}\end{bmatrix}\right]$$
  Ex.: Corre.
  run
  (You) run.

- Infl-Inf-Verb
  (The grammar does not yet produce syntactic contexts where these forms occur.)

The TAM rules that produce words are these:

- Inf-Verb
  This rule adds the constraint
  $$\left[\text{SYNSEM}|\text{LOCAL}|\text{CONT}|\text{HOOK}|\text{INDEX}|\text{E}|\text{MOOD}\quad \textit{infinitivo}\right]$$
  See Figure 10.2.

  Ex.: Ele pode correr.
  he can run
  He can run.

- Part-Pass-Inv
  This rule adds the constraint

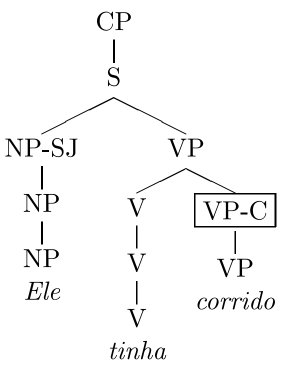


Figure 10.3: The Part-Pass-Inv lexical rule (corresponding node(s) in a box).

$$\left[\textsc{synsem}|\textsc{local}\ \left[\begin{array}{ll}\textsc{cat}|\textsc{head}|\textsc{vform} & \textit{particípio-passado-inv}\\ \textsc{cont}|\textsc{hook}|\textsc{index}|\textsc{e}|\textsc{aspect}|\textsc{perf} & +\end{array}\right]\right]$$

See Figure 10.3.

Ex.: Ele tinha corrido.
he had run
He had run.

- Noun-Modifying-Gerund
  This rule adds the constraint
  $\left[\textsc{synsem}|\textsc{local}|\textsc{cont}|\textsc{hook}|\textsc{index}|\textsc{e}|\textsc{mood}\ \ \textit{gerúndio}\right]$
  See Figure 10.4.

  Ex.: Ele comprou um livro descrevendo a Irlanda.
  he bought a book describing the Ireland
  He bought a book describing Ireland.

- Verb-Modifying-Gerund
  This rule adds the constraint
  $\left[\textsc{synsem}|\textsc{local}|\textsc{cont}|\textsc{hook}|\textsc{index}|\textsc{e}|\textsc{mood}\ \ \textit{gerúndio}\right]$
  See Figure 10.5.

  Ex.: Ele saiu correndo.
  he left running
  He left running.

The PN rules are the following:

- 1Sg-OR-3Sg-Verb
  This rule adds the constraint

$$\left[\textsc{synsem}|\textsc{local}|\textsc{cat}|\textsc{val}|\textsc{subj}\ \left\langle\left[\textsc{local}|\textsc{cat}|\textsc{head}|\textsc{agr}|\textsc{png}\ \left[\begin{array}{ll}\textsc{person} & \textit{first-or-third}\\ \textsc{number} & \textit{singular}\end{array}\right]\right]\right\rangle\right]$$

  See Figure 10.6.

  Ex.: Corria.
  ran.FIRST-OR-THIRD-PERSON-SINGULAR
  I/he/she/it ran.

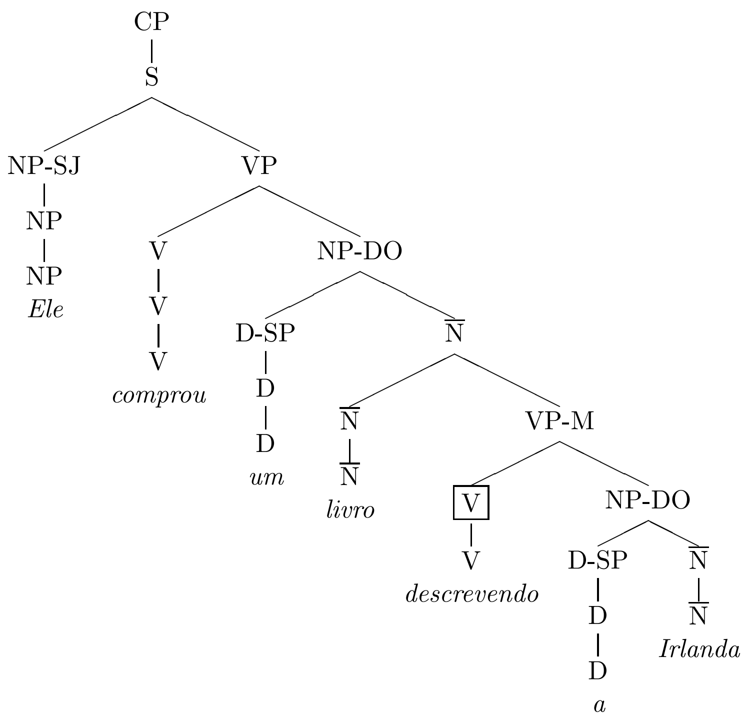


Figure 10.4: The Noun-Modifying-Gerund lexical rule (corresponding node(s) in a box).

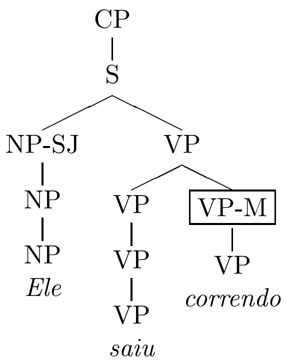


Figure 10.5: The Verb-Modifying-Gerund lexical rule (corresponding node(s) in a box).

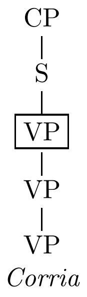


Figure 10.6: The 1Sg-OR-3Sg-Verb lexical rule (corresponding node(s) in a box).

- 1Sg-Verb
  This rule adds the constraint
  $$\left[\textsc{synsem}|\textsc{local}|\textsc{cat}|\textsc{val}|\textsc{subj}\left\langle\left[\textsc{local}|\textsc{cat}|\textsc{head}|\textsc{agr}|\textsc{png}\begin{bmatrix}\textsc{person} & \textit{first}\\ \textsc{number} & \textit{singular}\end{bmatrix}\right]\right\rangle\right]$$
  Ex.: Eu corro.
  I run
  I run.

- 2Sg-Verb
  This rule adds the constraint
  $$\left[\textsc{synsem}|\textsc{local}|\textsc{cat}|\textsc{val}|\textsc{subj}\left\langle\left[\textsc{local}|\textsc{cat}|\textsc{head}|\textsc{agr}|\textsc{png}\begin{bmatrix}\textsc{person} & \textit{second}\\ \textsc{number} & \textit{singular}\end{bmatrix}\right]\right\rangle\right]$$
  Ex.: Tu corres.
  you run
  You run.

- 3Sg-Verb
  This rule adds the constraint
  $$\left[\textsc{synsem}|\textsc{local}|\textsc{cat}|\textsc{val}|\textsc{subj}\left\langle\left[\textsc{local}|\textsc{cat}|\textsc{head}|\textsc{agr}|\textsc{png}\begin{bmatrix}\textsc{person} & \textit{third}\\ \textsc{number} & \textit{singular}\end{bmatrix}\right]\right\rangle\right]$$
  Ex.: Ele corre.
  he runs
  He runs.

- 1Pl-Verb
  This rule adds the constraint
  $$\left[\textsc{synsem}|\textsc{local}|\textsc{cat}|\textsc{val}|\textsc{subj}\left\langle\left[\textsc{local}|\textsc{cat}|\textsc{head}|\textsc{agr}|\textsc{png}\begin{bmatrix}\textsc{person} & \textit{first}\\ \textsc{number} & \textit{plural}\end{bmatrix}\right]\right\rangle\right]$$
  Ex.: Nós corremos.
  we run
  We run.

- 2Pl-Verb
  This rule adds the constraint
  $$\left[\textsc{synsem}|\textsc{local}|\textsc{cat}|\textsc{val}|\textsc{subj}\left\langle\left[\textsc{local}|\textsc{cat}|\textsc{head}|\textsc{agr}|\textsc{png}\begin{bmatrix}\textsc{person} & \textit{second}\\ \textsc{number} & \textit{plural}\end{bmatrix}\right]\right\rangle\right]$$
  Ex.: Vós correis.
  you run
  You run.

- 3Pl-Verb
  This rule adds the constraint
  $$\left[\textsc{synsem}|\textsc{local}|\textsc{cat}|\textsc{val}|\textsc{subj}\left\langle\left[\textsc{local}|\textsc{cat}|\textsc{head}|\textsc{agr}|\textsc{png}\begin{bmatrix}\textsc{person} & \textit{third}\\ \textsc{number} & \textit{plural}\end{bmatrix}\right]\right\rangle\right]$$

Ex.:	Eles	correm.
	they	run

They run.

10.1.1 Postponed Coverage or Known Limitations

Passives have not been implemented yet, and accordingly the past participle forms that inflect for gender and number are not produced by the grammar.

There is work in progresss with the objective of integrating shallow tools with LXGram. The inflectional rules are likely to change as a result of this. Also, not all verbal paradigms are implemented, because these external tools can fill the gaps in the implementation.

10.2 Nominal Inflection

Four rules are used to produce all gender and number inflected forms of nouns, pronouns and all noun modifiers/specifiers that inflect for these two categories (determiners, adjectives, possessives, etc.). They are also organized in two sets. The rules in the first one are responsible to constrain gender information and their output is a lexeme, the rules in the second set inflect these elements in number and produce words.

Not all of these elements undergo the gender rules, however, viz. they only do if they show gender distinctions. This information must of course be lexically specified (e.g. "gato" – *cat* – has a feminine version "gata"; "carro" – *car* does not). This is the current implementation of nominal morphology in LXGram. This implementation will change in future versions, in order to better accommodate the integration of an external morphological analyzer.

For adjectives there is another rule to produce "-íssimo" suffixed forms (a kind of superlative), that applies before any of the other rules.

The inflectional rules for nominals are the following:

- Masc-Nominal
  This rule produces the masculine forms. For nouns and adjectives, it applies only to those items that show gender distinctions (which must be stated in the lexicon). In the example below it applies neither to the noun "sofá" (*couch*), which, referring to an inanimate entity, has a single, lexically specified gender, nor to the adjective "verde" (*green*), which does not overtly show gender agreement.

  See Figure 10.7.

Ex.:	Os	meus	dois	gatos	pretos	destruíram	o	sofá	verde.
	the	my	two	cats.MASCULINE	black	destroyed	the	couch.MASCULINE	green

  My two black (male) cats destroyed the green couch.

- Fem-Nominal
  This rule produces the feminine forms. For nouns and adjectives, it applies only to those items that show gender distinctions (which must be stated in the lexicon). In the example below it applies neither to the noun "mesa" (*table*), which, referring to an inanimate entity, has a single, lexically specified gender, nor to the adjective "verde" (*green*), which does not overtly show gender agreement.

  See Figure 10.8.

Ex.:	As	minhas	duas	gatas	pretas	destruíram	a	mesa	verde.
	the	my	two	cats.FEMININE	black	destroyed	the	table.FEMININE	green

  My two black (female) cats destroyed the green table.

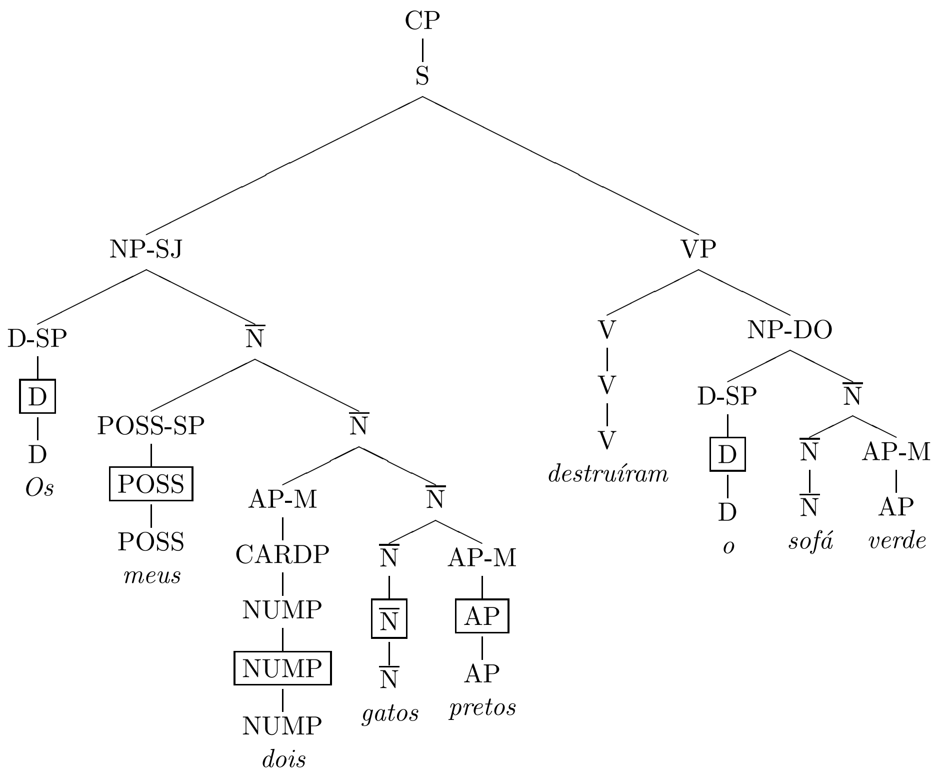


Figure 10.7: The Masc-Nominal lexical rule (corresponding node(s) in a box).

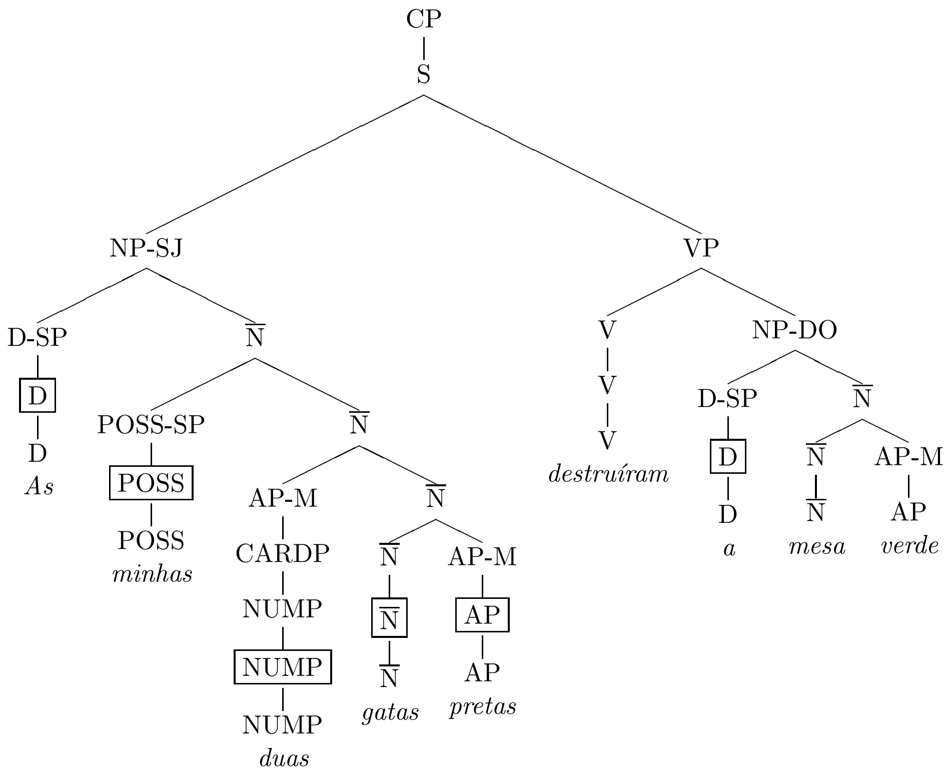


Figure 10.8: The Fem-Nominal lexical rule (corresponding node(s) in a box).

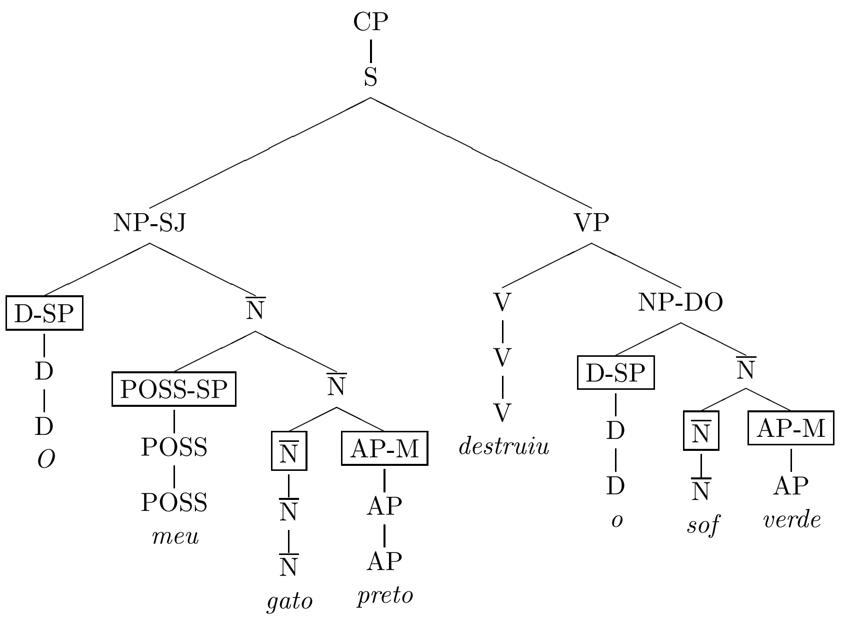


Figure 10.9: The Sg-Nominal lexical rule (corresponding node(s) in a box).

- Sg-Nominal
  This rule produces the singular forms from a gender specified lexeme. It will apply after inflectional rules for gender (most of the nominals in the example below) or directly to a lexical item if it does not inflect for gender ("sofá" and "verde" in the example below).

  See Figure 10.9.

Ex.:	O	meu	gato	preto	destruiu	o	sofá	verde.
	the	my	cat	black	destroyed	the	couch	green

  My black cat destroyed the green couch.

- Pl-Nominal
  This rule produces the plural form from a gender specified lexeme. Since number inflection is required to be more peripheral than gender inflection, items that inflect for gender but do not possess a singular form (some cardinals, like "dois"/ "duas" in the example below) must nevertheless undergo this rule as well. Interestingly, they show the `-s` suffix typical of plural nominals, and thus, as far as this rule is concerned, are not irregular.

  See Figure 10.10.

Ex.:	Os	meus	dois	gatos	pretos	miaram.
	the	my	two	cats	black	meowed

  My two black cats meowed.

- Superlative-Adjective
  This rule produces the `-íssimo` form of adjectives. It applies before gender and number inflectional rules, because this suffix is less peripheral and because the `-íssimo` form can have morphological properties different from the ones of its base form, as the example below illustrates (where the base "contente" shows no gender agreement but the resulting form does).

  See Figure 10.11.

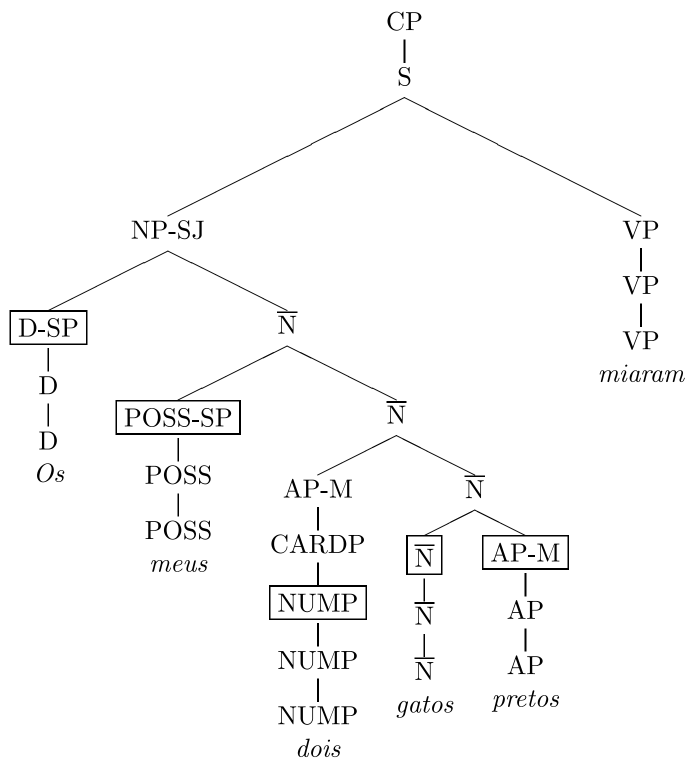


Figure 10.10: The Pl-Nominal lexical rule (corresponding node(s) in a box).

Ex.: A Maria está contentíssima.
the Maria is very happy
Maria is very happy.

10.2.1 Postponed Coverage or Known Limitations

LXGram has no support for diminutive/ evaluative forms of nouns and adjectives, like "coisinha" (*thingy*), "asseadinho" (*clean*, in a depreciative sense).

The list of irregular forms of nouns and adjectives is not exhaustive, since LXGram is intended to be integrated with an external morphological component that contains such a list.

10.3 Punctuation

There are also four punctuation rules in LXGram (see Section 5.9). They are more peripheral than any of the inflectional rules. They are defined in the file `punctuation.tdl`.

- Punctuation-Period

  This rule has the constraints:

$$\begin{bmatrix} \text{PUNCT}|\text{R-PUNCT} & \begin{bmatrix} \textit{period} & \\ \text{S-FORCE} & \textit{proposition-or-command} \\ \text{ELLIPTICAL} & - \end{bmatrix} \\ \text{DTR}|\text{PUNCT}|\text{R-PUNCT} & \textit{no-punctuation-mark} \end{bmatrix}$$

  See Figure 10.12.

  Ex.: Ele corre.
  he runs
  He runs.

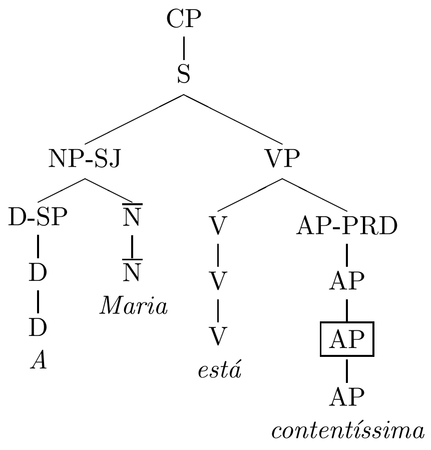


Figure 10.11: The Superlative-Adjective lexical rule (corresponding node(s) in a box).

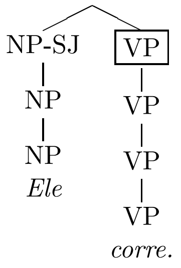


Figure 10.12: The Punctuation-Period lexical rule (corresponding node(s) in a box).

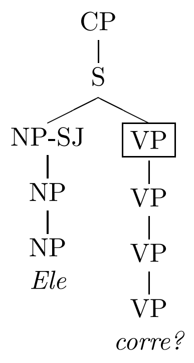


Figure 10.13: The PUNCTUATION-QUESTION-MARK lexical rule (corresponding node(s) in a box).

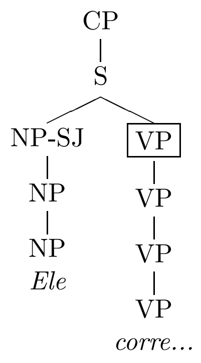


Figure 10.14: The PUNCTUATION-ELLIPSIS lexical rule (corresponding node(s) in a box).

- PUNCTUATION-QUESTION-MARK
  This rule has the constraints:

$$\begin{bmatrix} \text{PUNCT}|\text{R-PUNCT} & \begin{bmatrix} \textit{question-mark} & \\ \text{S-FORCE} & \textit{question} \\ \text{ELLIPTICAL} & - \end{bmatrix} \\ \text{DTR}|\text{PUNCT}|\text{R-PUNCT} & \textit{no-punctuation-mark} \end{bmatrix}$$

  See Figure 10.13.

  Ex.: Ele corre?
  he runs
  Does he run?

- PUNCTUATION-ELLIPSIS
  This rule has the constraints:

$$\begin{bmatrix} \text{PUNCT}|\text{R-PUNCT} & \begin{bmatrix} \textit{ellipsis} & \\ \text{S-FORCE} & \textit{proposition-or-command} \\ \text{ELLIPTICAL} & + \end{bmatrix} \\ \text{DTR}|\text{PUNCT}|\text{R-PUNCT} & \textit{no-punctuation-mark} \end{bmatrix}$$

  See Figure 10.14.

  Ex.: Ele corre...
  he runs
  He runs...

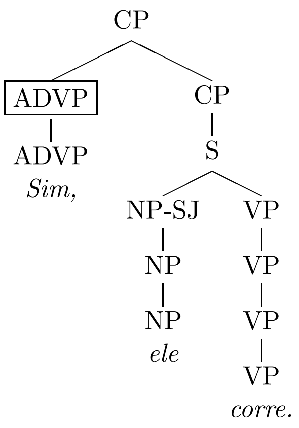


Figure 10.15: The Punctuation-Comma lexical rule (corresponding node(s) in a box).

- Punctuation-Comma
  This rule has the constraints:
  $$\begin{bmatrix} \textsc{punct}|\textsc{r-punct} & \textit{comma} \\ \textsc{dtr}|\textsc{punct}|\textsc{r-punct} & \textit{no-punctuation-mark} \end{bmatrix}$$
  See Figure 10.15.

  Ex.: Sim, ele corre.
  yes he runs
  Yes, he runs.

The remaining lexical rules do not have an impact on the orthographic representation of words. They are responsible for alternations in grammatical properties of words.

- Arg-Adj-To-Mod-Adj
  This rule converts adjectives that can be arguments of nouns into adjectives that modify nouns, adding an extra *abstract_a_rel* relation to their semantics. See Section 8.6.7.

  See Figure 10.16.

  Ex.: Tenho um carro americano.
  I have a car American
  I have an American car.

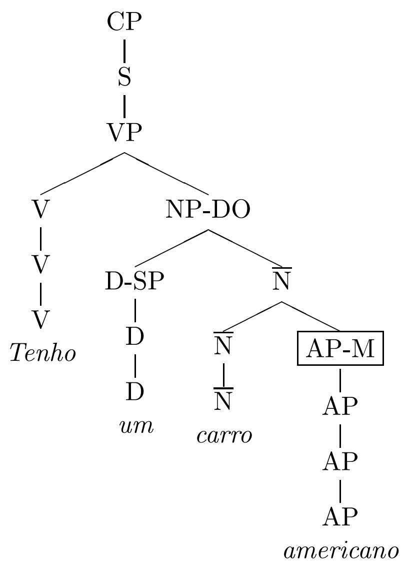


Figure 10.16: The ARG-ADJ-TO-MOD-ADJ lexical rule (corresponding node(s) in a box).

Chapter 11

Test Suite

This chapter describes the test suites used in LXGram.

The test suites attempt to attest many combinations of phenomena and word patterns systematically. Figure 11.1 shows an example of this. This makes it easier to debug if needed, since it allows one to understand better where the problem is. It also enables one to check more thoroughly whether changes in one place in the grammar have made damage in other analyses.

```
Ele mora perto do centro.
Ele mora perto.
Ele mora muito perto do centro.
Ele mora muito perto.
Ele mora mais perto do centro.
Ele mora mais perto.
Ele mora muito mais perto do centro.
Ele mora muito mais perto.
```

Figure 11.1: Example test suite sentences

In the test suite for the phase A.1 (see Chapter 4), the test items are grouped by phenomenon. Additionally, each test item is tagged with several pieces of information, as depicted in Figure 11.2.

The fields that are filled in for each test item are the following:

- TestItem
  The test sentence, with additional | symbols for purposes of alignment with the gloss
- Gloss
  A gloss in English
- English
  The English translation
- Tagged
  The test sentence with part of speech tags as well as inflectional tags. These tags are produced manually.
- OtherPhenomena
  Other phenomena that the test item attests, besides the main phenomena under the section of which the test example appears

- TSNLPhenomenon
  Classification of this test item according to the phenomena list presented in [Oepen et al., 1997] and used in [incr tsdb()].

Some figures on the test suites developed so far as well as on grammar coverage and efficiency when parsing them can be found in Chapter 4.

Currently, the test suite for the phase A.1 is annotated and contains 202 items. The test suite for the phase A.4 is not annotated (it contains 851 items).

```
; @BeginPhenomenon Basic_Phrase_Structure_AdvP
...

; @TestItem   Ele | mora | perto | do | centro.
; @Gloss      he | lives | near | of the | center
; @English    He lives near the center.
; @Tagged     Ele/PRS#ms3 mora/MORAR/V/pi-3s perto/ADV de_/PREP o/DA#ms centro/CENTRO/CN#ms
; @OtherPhenomena Basic_Phrase_Structure_VP Basic_Phrase_Structure_PP Basic_Phrase_Structure
; @TSNLPPhenomenon  C_Complementation
Ele mora perto do centro.
```

Figure 11.2: Example test suite item with annotations

11.1 Ontology of Phenomena

Besides the classification of phenomena presented in [Oepen et al., 1997], which is roughly the one available in [incr tsdb()], test items are also being classified using a different ontology of phenomena, using the tags BeginPhenomenon and OtherPhenomena presented above.

This ontology follows the implementation agenda defined in Section 2.2 and is expanded as progress is made. Currently the ontology is the following:

- Basic_Phrase_Structure_VP
- Basic_Phrase_Structure_PP
- Basic_Phrase_Structure_AP
- Basic_Phrase_Structure_AdvP
- Auxiliaries
- Basic_Phrase_Structure_NP

Chapter 12

Appendix I - Lexical Types

This appendix contains the exhaustive list of the types that are used in LXGram for lexical entries.

- Common nouns
  - *noun-common-masc-count_or_mass-0comps-lex*
  - *noun-common-masc-count_or_mass-1comp-lex*
  - *noun-common-masc-count-0comps-lex*
  - *noun-common-masc-count-2nd_or_3rd-0comps-lex*
  - *noun-common-masc-mass-0comps-lex*
  - *noun-common-masc-mass-1comp-lex*
  - *noun-common-masc-count-1comp-lex*
  - *noun-common-masc-count-2nd_or_3rd-1comp-lex*
  - *noun-common-masc-count-2comps_de+de_por-lex*
  - *noun-common-masc-count-2comps_de+por-lex*
  - *noun-common-masc-count_or_mass-2comps_de+de_por-lex*
  - *noun-common-masc-pl-count-0comps-lex*
  - *noun-common-masc-sg_or_pl-count_or_mass-0comps-lex*
  - *noun-common-masc-sg_or_pl-count-0comps-lex*
  - *noun-common-masc-sg_or_pl-mass-0comps-lex*
  - *noun-common-fem-count_or_mass-0comps-lex*
  - *noun-common-fem-count_or_mass-1comp-lex*
  - *noun-common-fem-count_or_mass-2comps_de+por-lex*
  - *noun-common-fem-count-0comps-lex*
  - *noun-common-fem-count-2nd_or_3rd-0comps-lex*
  - *noun-common-fem-mass-0comps-lex*
  - *noun-common-fem-mass-1comp-lex*
  - *noun-common-fem-mass-2comps_de+de_por-lex*
  - *noun-common-fem-count-1comp-lex*
  - *noun-common-eventive-fem-count-1comp-lex*

  - *noun-common-fem-count-2nd_or_3rd-1comp-lex*
  - *noun-common-fem-count-2comps_de+de_por-lex*
  - *noun-common-fem-count-2comps_de+por-lex*
  - *noun-common-fem-count_or_mass-2comps_de+de_por-lex*
  - *noun-common-fem-pl-count-0comps-lex*
  - *noun-common-fem-sg_or_pl-count_or_mass-0comps-lex*
  - *noun-common-fem-sg_or_pl-count-0comps-lex*
  - *noun-common-fem-sg_or_pl-mass-0comps-lex*
  - *noun-common-masc_or_fem-count-0comps-lex*
  - *noun-common-masc_or_fem-count-1comp-lex*
  - *noun-common-masc_or_fem-count-2nd_or_3rd-0comps-lex*
  - *noun-common-masc_or_fem-count-2nd_or_3rd-1comp-lex*
  - *noun-common-masc_with_fem-count-0comps-lex*
  - *noun-common-masc_with_fem-count-2nd_or_3rd-0comps-lex*
  - *noun-common-masc_with_fem-count_or_mass-0comps-lex*
  - *noun-common-pretitle-masc_with_fem-count-2nd_or_3rd-0comps-lex*
  - *noun-common-masc_with_fem-count-1comp-lex*
  - *noun-common-masc_with_fem-count-2nd_or_3rd-1comp-lex*

- Proper nouns
  - *noun-proper-masc-opt_det-lex*
  - *noun-proper-fem-opt_det-lex*
  - *noun-proper-masc-opt_det-2nd_or_3rd-lex*
  - *noun-proper-masc-sg_or_pl-opt_det-2nd_or_3rd-lex*
  - *noun-proper-fem-opt_det-2nd_or_3rd-lex*
  - *noun-proper-fem-sg_or_pl-opt_det-2nd_or_3rd-lex*
  - *noun-proper-masc_or_fem-opt_det-2nd_or_3rd-lex*
  - *noun-proper-masc_or_fem-sg_or_pl-opt_det-2nd_or_3rd-lex*
  - *noun-proper-masc-obl_det-lex*
  - *noun-proper-masc-sg_or_pl-obl_det-lex*
  - *noun-proper-fem-obl_det-lex*
  - *noun-proper-fem-sg_or_pl-obl_det-lex*
  - *noun-proper-masc-pl-obl_det-lex*
  - *noun-proper-fem-pl-obl_det-lex*
  - *noun-proper-masc_or_fem-obl_det-lex*
  - *noun-proper-masc_or_fem-sg_or_pl-obl_det-lex*
  - *noun-proper-masc_or_fem-pl-obl_det-lex*
  - *noun-proper-masc-mod_det-lex*
  - *noun-proper-masc-sg_or_pl-mod_det-lex*

  - *noun-proper-fem-mod_det-lex*
  - *noun-proper-fem-sg_or_pl-mod_det-lex*
  - *noun-proper-masc_or_fem-mod_det-lex*
  - *noun-proper-masc_or_fem-sg_or_pl-mod_det-lex*

- Name Particles
  - *name-particle-lex*
- Verbs
  - Auxiliaries for compound tenses
    - *verb-compound_tense_aux-lex*
  - Copular verbs
    - *verb-individual_lvl_copula-lex*
    - *verb-stage_lvl_copula-lex*
    - *verb-identity_copula-lex*
  - Zero-place
    - *verb-0place-lex*
  - Intransitive and unaccusative
    - *verb-intrans-lex*
  - Direct transitive
    - *verb-dir_trans-lex*
    - *verb-dir_trans-indef_null_obj-lex*
    - *verb-dir_trans-opaque-lex*
  - Indirect transitive
    - *verb-ind_trans-lex*
  - Ditransitive
    - *verb-ditrans-lex*
  - Prepositional intransitive
    - *verb-intrans-prep_acercade_de_em_sobre-lex*
    - *verb-intrans-prep_acercade_de_sobre-lex*
    - *verb-intrans-prep_com_de-lex*
    - *verb-intrans-prep_com-lex*
    - *verb-intrans-prep_de-lex*
    - *verb-intrans-prep_em-lex*
    - *verb-intrans-prep_sobre-lex*
  - Oblique intransitive
    - *verb-intrans-obl_direction-lex*
    - *verb-intrans-obl_location-lex*
  - With a clausal argument
    - *verb-comp_cp-lex*
    - *verb-comp_cp-comp_indir-lex*

        * *verb-comp_cp_declarative-lex*
        * *verb-comp_cp_declarative-comp_indir-lex*
        * *verb-comp_cp_interrogative-lex*
        * *verb-comp_cp_interrogative-comp_indir-lex*
    - Subject Raising
        * *verb-subj_raising-comp_vp-lex*
        * *verb-subj_raising-comp_pp_a-lex*
        * *verb-subj_raising-comp_pp_de-lex*
        * *verb-subj_raising-comp_pp_por-lex*
    - Subject Control
        * *verb-subj_control-lex*

- Adjectives
    - *adjective-isect-gender_infl-0comps-lex*
    - *adjective-isect-gender_uninfl-0comps-lex*
    - *adjective-isect-uninfl-0comps-lex*
    - *adjective-isect-postnom-gender_infl-0comps-lex*
    - *adjective-isect-postnom-gender_uninfl-0comps-lex*
    - *adjective-isect-postnom-uninfl-0comps-lex*
    - *adjective-isect-postnom-gender_uninfl-1comp_pp_de-lex*
    - *adjective-isect-postnom-synth-gender_infl-lex*
    - *adjective-isect-postnom-synth-gender_uninfl-lex*
    - *adjective-isect-gender_infl-1comp_pp_a_com-lex*
    - *adjective-isect-gender_infl-1comp_pp_a-lex*
    - *adjective-isect-gender_uninfl-1comp_pp_a-lex*
    - *adjective-isect-gender_infl-1comp_pp_com-lex*
    - *adjective-isect-gender_uninfl-1comp_pp_com-lex*
    - *adjective-isect-gender_infl-1comp_pp_de-lex*
    - *adjective-isect-gender_uninfl-1comp_pp_de-lex*
    - *adjective-isect-gender_infl-1comp_pp_para-lex*
    - *adjective-isect-gender_uninfl-1comp_pp_para-lex*
    - *adjective-scopal-pred_exist_sem-gender_infl-0comps-lex*
    - *adjective-scopal-pred_exist_sem-gender_uninfl-0comps-lex*
    - *adjective-scopal-pred_subj_n_sem-gender_infl-0comps-lex*
    - *adjective-scopal-pred_subj_n_sem-gender_uninfl-0comps-lex*
    - *adjective-scopal-pred_subj_n_sem-synth-gender_infl-0comps-lex*
    - *adjective-scopal-nonpred-ungradable-prenom-gender_infl-0comps-lex*
    - *adjective-scopal-nonpred-ungradable-prenom-gender_uninfl-0comps-lex*
    - *adjective-scopal-nonpred-ungradable-prenom-uninfl-0comps-lex*
    - *adjective-scopal-nonpred-prenom-gender_infl-0comps-lex*

  - *adjective-scopal-nonpred-prenom-synth-gender_uninfl-0comps-lex*
  - *adjective-isect-comparative-lex*
  - *adjective-isect-postnom-comparative-lex*
  - *adjective-scopal-comparative-lex*
  - *adjective-scopal-nonpred-prenom-comparative-lex*
  - *adjective-qualquer-lex*
  - *adjective-noun_argument-gender-infl-lex*
  - *adjective-noun_argument-gender-uninfl-lex*
  - *adjective-isect-nonpred-ungradable-postnom-gender_infl-no_plural-0comps-lex*
  - *adjective-isect-nonpred-ungradable-postnom-gender_uninfl-no_plural-0comps-lex*
  - *adjective-superlative-lex*
  - *adjective-noun_argument-gender_infl-lex*
  - *adjective-noun_argument-gender_uninfl-lex*

- Adverbs
  - *adverb-isect-0comps-lex*
  - *adverb-isect-0comps-proctrigger-lex*
  - *adverb-isect-subj-0comps-lex*
  - *adverb-isect-subj-0comps-proctrigger-lex*
  - *adverb-isect-subj-0comps-postverbal-lex*
  - *adverb-isect-subj-1comp_com-lex*
  - *adverb-isect-subj-1comp_de-lex*
  - *adverb-scopal-lex*
  - *adverb-scopal-proctrigger-lex*
  - *adverb-scopal-temporal-proctrigger-lex*
  - *adverb-scopal-proctrigger-preverbal-lex*
  - *adverb-quantificational-lex*
  - *adverb-np-adjunct-lex*
  - *adverb-degree-mais_do_que-lex*
  - *adverb-degree-menos_do_que-lex*
  - *adverb-degree-1comp_como_quanto-lex*
  - *adverb-degree-ungradable-lex*
  - *adverb-degree-ungradable-infl-masc-or-fem-lex*
  - *adverb-degree-ungradable-infl-neut-lex*
  - *adverb-degree-gradable-lex*
  - *adverb-degree_of_comparison-lex*
  - *adverb-np_semantics-locative-0comps-lex*
  - *adverb-np_semantics-locative-1comp_a_de-lex*
  - *adverb-np_semantics-locative-1comp_de-lex*

  - *adverb-temporal-lex*
  - *adverb-temporal-proctrigger-lex*
  - *adverb-locative-or-temporal-subj-1comp-lex*
  - *adverb-isect-ungradable-postnom-uninfl-1comp_s_or_como_s-lex*
  - *neg-lex*

- Pre-root elements
  - *discourse-element-lex*

- Prepositions
  - With semantic content
    - *preposition-predicational-lex*
    - *preposition-predicational-nom-lex*
    - *preposition-predicational_locative-lex*
    - *preposition-predicational_locative-nom_or_obl-lex*
    - *preposition-predicational_non_locative-lex*
    - *preposition-predicational_non_locative-com_case-lex*
    - *preposition-predicational-comp_advp-lex*
  - Argumental prepositions
    - *preposition-nonpredicational-comp_np-a-lex*
    - *preposition-nonpredicational-comp_np_or_vp-a-lex*
    - *preposition-nonpredicational-comp_np-acercade-lex*
    - *preposition-nonpredicational-comp_np-com-lex*
    - *preposition-nonpredicational-comp_np-de-lex*
    - *preposition-nonpredicational-comp_np_or_vp-de-lex*
    - *preposition-nonpredicational-comp_np-em-lex*
    - *preposition-nonpredicational-comp_np-para-lex*
    - *preposition-nonpredicational-comp_np-por-lex*
    - *preposition-nonpredicational-comp_np_or_vp-por-lex*
    - *preposition-nonpredicational-comp_np-sobre-lex*

- Determiners and Predeterminers
  - *definite-article-lex*
  - *definite-article-nosem-lex*
  - *demonstrative-masc-or-fem-lex*
  - *demonstrative-noquant-masc-or-fem-lex*
  - *demonstrative-neut-lex*
  - *demonstrative-noquant-neut-lex*
  - *determiner-indef-div-uninfl-lex*
  - *determiner-indef-div-gender_uninfl-lex*
  - *determiner-indef-div-no_noun_ellipsis-lex*

  - *determiner-def-div-lex*
  - *determiner-mass_or_count-indef-masc_or_fem-lex*
  - *determiner-mass_or_count-indef-neut-lex*
  - *determiner-mass_or_count-indef-mascneut_or_fem-lex*
  - *determiner-mass_or_count-indef-specific-masc_or_fem-lex*
  - *determiner-mass_or_count-indef-mascneut_or_fem-1comp_como_quanto-lex*
  - *univ-quant-over-individuals-masc-or-fem-sg-lex*
  - *univ-quant-over-individuals-neut-sg-lex*
  - *univ-quant-over-individuals-masc-or-fem-pl-lex*
  - *univ-quant-over-individuals-masc-or-fem-nodet-lex*
  - *univ-quant-over-parts-masc-or-fem-sg-lex*
  - *univ-quant-over-parts-neut-sg-lex*
  - *univ-quant-over-parts-masc-or-fem-pl-lex*
  - *determiner-mass_or_count-indef-mascneut_or_fem-pl-multi_poucos-lex*
  - *determiner-mass_or_count-indef-mascneut_or_fem-pl-multi_quantos-lex*
  - *predeterminer-def-quant_sem-uninfl-lex*
  - *determiner-mass_or_count-indef-mascneut_or_fem-pl-multi_tantos-lex*
  - *determiner-mass_or_count-indef-mascneut_or_fem-pl-multi_e_tantos-lex*
  - *determiner-indef-mascneut_or_fem-multi_mais-lex*
  - *determiner-indef-mascneut_or_fem-pl-multi_menos-lex*
  - *determiner-indef-empty_lp-gender_uninfl_non_neuter-multi_um-lex*
  - *determiner-indef-mascneut_or_fem-multi_qualquer-lex*
  - *determiner-indef-non_div-full_lp-uninfl-lex*
  - *determiner-indef-non_div-empty_lp-masc_with_fem-lex*
  - *np-determiner-0comps-def-semantic_head_agr-lex*
  - *determiner-mass_or_count-indef-gender_uninfl_non_neuter-lex*
  - Non-Initial Elements of Multi-Word Determiners
    - *determiner-particle-mais-lex*
    - *determiner-particle-menos-lex*
    - *determiner-particle-poucos-lex*
    - *determiner-particle-qualquer-lex*
    - *determiner-particle-quantos-lex*
    - *determiner-particle-tantos-lex*
    - *determiner-particle-um-lex*
    - *determiner-particle-e_tantos-lex*

- Post-Nominal Demonstratives
  - *demonstrative-postnominal-masc_with_fem-lex*

- Possessives

- *possessive-meu-adjunct-lex*
- *possessive-meu-adjunct-bp-lex*
- *possessive-meu-comp1-lex*
- *possessive-meu-comp1-bp-lex*
- *possessive-meu-comp2-lex*
- *possessive-meu-comp2-bp-lex*
- *possessive-meu_prprio-adjunct-lex*
- *possessive-meu_prprio-adjunct-bp-lex*
- *possessive-meu_prprio-comp1-lex*
- *possessive-meu_prprio-comp1-bp-lex*
- *possessive-meu_prprio-comp2-lex*
- *possessive-meu_prprio-comp2-bp-lex*
- *possessive-teu-adjunct-lex*
- *possessive-teu-adjunct-bp-lex*
- *possessive-teu-comp1-lex*
- *possessive-teu-comp1-bp-lex*
- *possessive-teu-comp2-lex*
- *possessive-teu-comp2-bp-lex*
- *possessive-teu_prprio-adjunct-lex*
- *possessive-teu_prprio-adjunct-bp-lex*
- *possessive-teu_prprio-comp1-lex*
- *possessive-teu_prprio-comp1-bp-lex*
- *possessive-teu_prprio-comp2-lex*
- *possessive-teu_prprio-comp2-bp-lex*
- *possessive-nosso-adjunct-lex*
- *possessive-nosso-adjunct-bp-lex*
- *possessive-nosso-comp1-lex*
- *possessive-nosso-comp1-bp-lex*
- *possessive-nosso-comp2-lex*
- *possessive-nosso-comp2-bp-lex*
- *possessive-nosso_prprio-adjunct-lex*
- *possessive-nosso_prprio-adjunct-bp-lex*
- *possessive-nosso_prprio-comp1-lex*
- *possessive-nosso_prprio-comp1-bp-lex*
- *possessive-nosso_prprio-comp2-lex*
- *possessive-nosso_prprio-comp2-bp-lex*
- *possessive-vosso-adjunct-lex*
- *possessive-vosso-adjunct-bp-lex*
- *possessive-vosso-comp1-lex*

- *possessive-vosso-comp1-bp-lex*
- *possessive-vosso-comp2-lex*
- *possessive-vosso-comp2-bp-lex*
- *possessive-vosso_prprio-adjunct-lex*
- *possessive-vosso_prprio-adjunct-bp-lex*
- *possessive-vosso_prprio-comp1-lex*
- *possessive-vosso_prprio-comp1-bp-lex*
- *possessive-vosso_prprio-comp2-lex*
- *possessive-vosso_prprio-comp2-bp-lex*
- *possessive-seu-third_person-adjunct-lex*
- *possessive-seu-third_person-adjunct-bp-lex*
- *possessive-seu-third_person-comp1-lex*
- *possessive-seu-third_person-comp1-bp-lex*
- *possessive-seu-third_person-comp2-lex*
- *possessive-seu-third_person-comp2-bp-lex*
- *possessive-seu_prprio-third_person-adjunct-lex*
- *possessive-seu_prprio-third_person-adjunct-bp-lex*
- *possessive-seu_prprio-third_person-comp1-lex*
- *possessive-seu_prprio-third_person-comp1-bp-lex*
- *possessive-seu_prprio-third_person-comp2-lex*
- *possessive-seu_prprio-third_person-comp2-bp-lex*
- *possessive-seu-second_person-adjunct-lex*
- *possessive-seu-second_person-adjunct-bp-lex*
- *possessive-seu-second_person-comp1-lex*
- *possessive-seu-second_person-comp1-bp-lex*
- *possessive-seu-second_person-comp2-lex*
- *possessive-seu-second_person-comp2-bp-lex*
- *possessive-seu_prprio-second_person-adjunct-lex*
- *possessive-seu_prprio-second_person-adjunct-bp-lex*
- *possessive-seu_prprio-second_person-comp1-lex*
- *possessive-seu_prprio-second_person-comp1-bp-lex*
- *possessive-seu_prprio-second_person-comp2-lex*
- *possessive-seu_prprio-second_person-comp2-bp-lex*
- *possessive-prprio-adjunct-lex*
- *possessive-prprio-adjunct-bp-lex*
- *possessive-prprio-comp1-lex*
- *possessive-prprio-comp1-bp-lex*
- *possessive-prprio-comp2-lex*
- *possessive-prprio-comp2-bp-lex*

- Vague Quantifiers
  - *vague-quantifier-lex*
  - *vague-quantifier-mass-lex*
  - *vague-quantifier-plural-lex*
  - *vague-quantifier-gender_uninfl-plural-lex*
- Indefinite specifics
  - *indefinite-specific-lex*
- Pronouns
  - Personal pronouns
    * *personal-pronoun-eu-lex*
    * *personal-pronoun-eu_mesmo-lex*
    * *personal-pronoun-eu_prprio-lex*
    * *personal-pronoun-me-lex*
    * *personal-pronoun-mim-lex*
    * *personal-pronoun-mim_mesmo-lex*
    * *personal-pronoun-mim_prprio-lex*
    * *personal-pronoun-migo-lex*
    * *personal-pronoun-migo_mesmo-lex*
    * *personal-pronoun-migo_prprio-lex*
    * *personal-pronoun-tu-lex*
    * *personal-pronoun-tu_mesmo-lex*
    * *personal-pronoun-tu_prprio-lex*
    * *personal-pronoun-te-lex*
    * *personal-pronoun-ti-lex*
    * *personal-pronoun-ti_mesmo-lex*
    * *personal-pronoun-ti_prprio-lex*
    * *personal-pronoun-tigo-lex*
    * *personal-pronoun-tigo_mesmo-lex*
    * *personal-pronoun-tigo_prprio-lex*
    * *personal-pronoun-ele-lex*
    * *personal-pronoun-ele_todo-lex*
    * *personal-pronoun-ele-expletive-lex*
    * *personal-pronoun-ele_mesmo-lex*
    * *personal-pronoun-ele_prprio-lex*
    * *personal-pronoun-o-lex*
    * *personal-pronoun-os-lex*
    * *personal-pronoun-lhe-lex*
    * *personal-pronoun-lhes-lex*
    * *personal-pronoun-se-lex*
    * *personal-pronoun-si-lex*

    - *personal-pronoun-si_mesmo-lex*
    - *personal-pronoun-si_prprio-lex*
    - *personal-pronoun-sigo-lex*
    - *personal-pronoun-sigo_mesmo-lex*
    - *personal-pronoun-sigo_prprio-lex*
    - *personal-pronoun-ns-lex*
    - *personal-pronoun-ns_prprios-lex*
    - *personal-pronoun-ns_mesmos-lex*
    - *personal-pronoun-ns_todos-lex*
    - *personal-pronoun-nos-lex*
    - *personal-pronoun-nosco-lex*
    - *personal-pronoun-nosco_todos-lex*
    - *personal-pronoun-nosco_mesmos-lex*
    - *personal-pronoun-nosco_prprios-lex*
    - *personal-pronoun-vs-lex*
    - *personal-pronoun-vs_todos-lex*
    - *personal-pronoun-vs_mesmos-lex*
    - *personal-pronoun-vs_prprios-lex*
    - *personal-pronoun-vos-lex*
    - *personal-pronoun-vosco-lex*
    - *personal-pronoun-vosco_todos-lex*
    - *personal-pronoun-vosco_mesmos-lex*
    - *personal-pronoun-vosco_prprios-lex*
    - *personal-pronoun-voc-lex*
    - *personal-pronoun-voc_mesmo-lex*
    - *personal-pronoun-voc_prprio-lex*
    - *personal-pronoun-vocs-lex*
    - *personal-pronoun-vocs_todos-lex*
    - *personal-pronoun-vocs_mesmos-lex*
    - *personal-pronoun-vocs_prprios-lex*
  - Relative pronouns
    - *relative-pronoun-lex*
  - Other pronouns
    - *pronoun-particle-mesmo-lex*
    - *pronoun-particle-prprio-lex*
    - *o+mesmo-lex*
    - *o+prprio-lex*

- Cardinals
  - *plus-particle-lex*
  - *cardinal-atomic-sg-lex*
  - *cardinal-atomic-gender_infl-pl-lex*

  - *cardinal-atomic-gender_uninfl-pl-lex*
  - *cardinal-2digit-lex*
  - *cardinal-3digit-gender_infl-lex*
  - *cardinal-3digit-gender_uninfl-lex*
  - *cardinal-4digit-lex*

- Ordinals
  - *ordinal-lex*
- Other adnominals
  - *mesmo-lex*
  - *outro-postdeterminer-lex*
  - *outro-determiner-lex*
  - *tal-determiner-lex*
  - *tal-postdeterminer-lex*
  - *que_no-outro_absent-lex*
  - *que_no-outro_present-lex*
  - *prprio-prenominal-lex*
- Pre-cardinals
  - *precardinal-uninflected-lex*
  - *precardinal-inflected-lex*
  - *precardinal-uninflected-2comps_e-lex*
  - *precardinal-uninflected-2comps_a-at-lex*
  - *precardinal-correlate-a-lex*
  - *precardinal-correlate-at-lex*
  - *precardinal-correlate-e-lex*
- Complementizers
  - *complementizer-decl-lex*
  - *complementizer-interr-lex*
- Conjunctions
  - *conjunction-lex*
- Other
  - *qualitative-particle-lex*
  - *comparative-particle-do_que-lex*
  - *comparative-particle-como_quanto-lex*

References

Allegranza, V. 1998a. Determination and quantification. In Van Eynde, F. and Schmidt, P., editors, *Linguistic Specifications for Typed Feature Structure Formalisms*, pages 281–314. Office for Official Publications of the European Communities, Luxembourg.

Allegranza, V. 1998b. Determiners as functors: Np structure in italian. In Balari, S. and Dini, L., editors, *Romance in Head-driven Phrase Structure Grammar*, volume 75 of *CSLI Lecture Notes*, pages 55–108. CSLI Publications, Stanford.

Bender, E. M., Flickinger, D., and Oepen, S. 2002. The Grammar Matrix: An open-source starter-kit for the development of cross-linguistically consistent broad-coverage precision grammars. In Carroll, J., Oostdijk, N., and Sutcliffe, R., editors, *Procedings of the Workshop on Grammar Engineering and Evaluation at the 19th International Conference on Computational Linguistics*, pages 8–14, Taipei, Taiwan.

Branco, A. and Costa, F. 2007. Identification and handling of dialectal variation with a single grammar. In Dirix, P., Schuurman, I., Vandeghinste, V., and Eynde, F. V., editors, *Proceedings of the 17th Meeting of Computational Linguistics in the Netherlands (CLIN17)*, pages 5–19, Utrecht. LOT.

Branco, A. and Henriques, T. 2003. Aspects of verbal inflection and lemmatization: Generalizations and algorithms. In Mendes, A. and Freitas, T., editors, *Proceedings of XVIII Annual Meeting of the Portuguese Association of Linguistics (APL)*, pages 201–210, Lisbon. APL.

Branco, A. and Silva, J. 2002. EtiFac: A facilitating tool for manual tagging. In *Proceedings of XVIII Annual Meeting of the Portuguese Association of Linguistics (APL)*, pages 81–90, Lisbon. APL.

Branco, A. and Silva, J. 2003a. Contractions: breaking the tokenization-tagging circularity. *Lecture Notes in Artificial Intelligence*, 2721:167–170.

Branco, A. and Silva, J. 2003b. A metric for the efficency of accurate tagging procedures. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2003)*, pages 71–77, Borovets, Bulgaria. Bulgarian Academy of Sciences.

Branco, A. and Silva, J. 2003c. Morpho-syntactic tagging without training corpus or lexicon: How far is it possible to get? In Mendes, A. and Freitas, T., editors, *Proceedings of XVIII Annual Meeting of the Portuguese Association of Linguistics (APL)*, pages 211–222, Lisbon. APL.

Branco, A. and Silva, J. 2003d. Portuguese-specific issues in the rapid development of state of the art taggers. In Branco, A., Mendes, A., and Ribeiro, R., editors, *Tagging and Shallow Processing of Portuguese: Workshop Notes of TASHA'2003*, pages 7–10, Lisbon. University of Lisbon, Faculty of Sciences, Department of Informatics.

Branco, A. and Silva, J. 2003e. Tokenization of Portuguese: resolving the hard cases. Technical Report TR-2003-4, Department of Informatics, University of Lisbon.

Branco, A. and Silva, J. 2004a. Evaluating solutions for the rapid development of state-of-the-art POS taggers for Portuguese. In Lino, M. T., Xavier, M. F., Ferreira, F., Costa, R., and Silva, R., editors, *Proceedings of the 4th International Conference on Language Resources and Evaluation (LREC2004)*, pages 507–510, Paris. ELRA.

Branco, A. and Silva, J. 2004b. Swift development of state of the art taggers for Portuguese. In Branco, A., Mendes, A., and Ribeiro, R., editors, *Language Technology for Portuguese: Shallow Processing Tools and Resources*, pages 29–45, Lisbon. Edições Colibri.

Callmeier, U. 2000. PET — A platform for experimentation with efficient HPSG processing techniques. *Natural Language Engineering*, 6(1):99–108. (Special Issue on Efficient Processing with HPSG).

Copestake, A., Flickinger, D., Pollard, C., and Sag, I. A. 2001. Minimal Recursion Semantics: An introduction. *Language and Computation*, 3:1–47.

Copestake, A., Flickinger, D., Sag, I. A., and Pollard, C. 2005. Minimal Recursion Semantics: An introduction. *Journal of Research on Language and Computation*, 3(2–3):281–332.

Copestake, A. 2002. *Implementing Typed Feature Structure Grammars.* CSLI Publications, Stanford.

Ferreira, E., Balsa, J., and Branco, A. 2007. Combining rule-based and statistical methods for named entity recognition in portuguese. In *Actas do V Workshop em Tecnologia da Informação e da Linguagem Humana TIL.*

Flickinger, D. 2000. On building a more efficient grammar by exploiting types. *Natural Language Engineering*, 6(1):15–28. (Special Issue on Efficient Processing with HPSG).

Ginzburg, J. and Sag, I. A. 2000. *Interrogative Investigations: the Form, Meaning, and Use of English Interrogatives.* CSLI Publications, Stanford, California.

Ionin, T. and Matushansky, O. 2006. The composition of complex cardinals. *Journal of Semantics*, 23:315–360.

Kasper, R. T. 1996. The semantics of recursive modification. Manuscript.

Web Page. LXConjugator. http://lxconjugator.di.fc.ul.pt.

Web Page. LXLemmatizer. http://lxlemmatizer.di.fc.ul.pt.

Web Page. LXSuite. http://lxsuite.di.fc.ul.pt.

Müller, A. 2002. The semantics of generic quantification in Brazilian Portuguese. *Probus: International Journal of Latin and Romance Linguistics*, 14(2):279–298.

Munn, A. and Schmitt, C. 1998. Against the nominal mapping parameter: Bare nouns in Brazilian Portuguese. In *Proceedings of NELS 29*, pages 339–353, Amherst, MA. GLSA, The University of Massachusetts.

Nerbonne, J. and Mullen, T. 2000. Null-headed nominals in German and English. In van Eynde, F., Schuurman, I., and Schelkens, N., editors, *Proc. of Computational Linguistics in the Netherlands 1998*, pages 143–64.

Oepen, S., Netter, K., and Klein, J. 1997. TSNLP – Test suites for natural language processing. *Linguistic Databases, CSLI Lecture Notes*, 77:3–28.

Oepen, S. 2001. [incr tsdb()] — competence and performance laboratory. User manual. Technical report, Computational Linguistics, Saarland University, Saarbrücken, Germany. In preparation.

Pollard, C. and Sag, I. 1987. *Information-Based Syntax and Semantics, Vol. 1.* CSLI Publications, Stanford.

Pollard, C. and Sag, I. 1994. *Head-Driven Phrase Structure Grammar.* Chicago University Press and CSLI Publications, Stanford.

Pullum, G. K. 1975. *People* deletion in English. In *Working Papers in Linguistics*, volume 14, pages 95–101. Ohio State University.

Sag, I. A., Wasow, T., and Bender, E. M. 2003. *Syntactic Theory – A Formal Introduction.* CSLI Publications, Stanford.

Van Eynde, F. 2003a. On the notion 'determiner'. In Müller, S., editor, *Proceedings of the HPSG-2003 Conference, Michigan State University, East Lansing*, pages 391–396, Stanford. CSLI Publications.

Van Eynde, F. 2003b. Prenominals in Dutch. In Kim, J.-B. and Wechsler, S., editors, *The Proceedings of the 9th International Conference on HPSG*, Stanford University. CSLI Publications.