Lexicalist Unification-Based Machine Translation

John Luis BEAVEN

Ph.D.
University of Edinburgh
1992

Declaration

I declare that this thesis has been composed by myself and that the work described is my own.

John Luis BEAVEN

Abstract

A novel approach to Machine Translation (MT), called **Shake-and-Bake**, is presented, which exploits recent advances in Computational Linguistics in terms of the rise of lexicalist unification-based grammar theories. It is argued that it overcomes many deficiencies of current methods, such as those based on transfer rules, interlingual representations, and isomorphic grammars.

The key advantages are a greater modularity of the monolingual components, which can be written with great independence of each other, using purely monolingual considerations. They can be used for parsing and generation, and may be used for multi-lingual translation systems.

The two monolingual components involved in translation are put into correspondence by means of a bilingual lexicon which contains information similar to what one might expect to find in an ordinary bilingual dictionary.

The approach is demonstrated by presenting very different Unification Categorial Grammars for small fragments of English and Spanish. Although their coverage is small, they have been chosen to contain linguistically interesting phenomena known to be difficult in MT, such as word order variation and clitic placement. These monolingual grammars are put into correspondence by means of a bilingual lexicon.

The **Shake-and-Bake** approach for MT consists of parsing the Source Language in any usual way, then looking up the words in the bilingual lexicon, and finally generating from the set of translations of these words, but allowing the Target Language grammar to instantiate the relative word ordering, taking advantage of the fact that the parse produces lexical and phrasal signs which are highly constrained (specifically in the semantics). The main algorithm presented for generation is a variation on the well-known CKY one used for parsing.

Acknowledgements

I would like to thank first and foremost my supervisors, Pete Whitelock and Elisabet Engdahl for their helpful advice, encouragement and inspiration throughout my research. During my final year, when Pete had left Edinburgh, and Elisabet was on maternity leave, Chris Mellish also helped me with many fruitful discussions. This final draft has also benefited from the corrections suggested by my examiners, Doug Arnold and Henry Thompson, to whom I am grateful for the feedback received during my *viva*.

Thanks are also due to my fellow students at the Department of Artificial Intelligence and at the Centre for Cognitive Science. I am reluctant to mention particular names for fear of leaving anyone out, but my warmest gratitude goes to Alan Black who tirelessly offered solutions to all my Unix gripes, Matt Crocker for his help with linguistics questions, Suresh Manandhar for his discussions about unification, logic and HPSG, Mike Reape, who with Pete Whitelock was involved in the development of Shake-and-Bake, and who with Mark Hepple wrote many LaTeXmacros used here.

John Foster, Anna Gavarró, Antonio Sanfilippo, and Max Volino were involved in several workshops in which we discussed the subtleties of Romance word order and clitic placement, and the ideas involving the Spanish grammar of Chapter 3 benefited from much discussions with the participants in the DYANA Workshop on Parametric Variation in Romance and Germanic held in Edinburgh in September 1989, and those of the Workshop on Categorial Grammar and Word Order held in Clermont Ferrand in May 1990.

Finally, I would like to thank the academic, administrative, computing, and research staff at the Department of Artificial Intelligence (and in particular the computing officers Michael Keightley and Craig Strachan who, with honorary computing officer Alan Black, never grew tired of compiling ever-growing versions of LaTeXwhich would not break under the size of my PATR-II boxes). Together with the students, they provided a warm and friendly research environment that I shall strongly miss.

This work was supported by a studentship from the Science and Engineering Research Council.

Contents

Chapter 1

Introduction

1.1 Machine translation

Machine translation (MT) has been a pursuit almost as old as computers themselves. Early systems were little more than on-line dictionaries and were a great disappointment, but it is hoped that as systems become linguistically more sophisticated, results become more interesting.

It is the general consensus amongst the MT community that Fully Automatic, High Quality MT (FAHQMT) is still very far away; some will even argue that it is impossible. The scope of this work is much less ambitious: we shall be examining a new approach to MT which makes the design and implementation of systems easier and more robust, and closer to becoming a usable tool for human translators.

One motivation behind the work is the belief that MT is a useful test-bed for linguistic theories and formalisms. There may not be a good theoretical definition of what exactly translation is, and how it differs from bilingual paraphrase. Ultimately, we may want to say that translation is whatever professional translators do, and they are the best judges as to the performance of a system.

One other reason why we believe that MT is interesting is that it forces the grammar writers to write grammars for at least two languages, and this might encourage the formalisms to move away from some of the anglocentrism into which many computational linguists may be tempted.

1.2 Current approaches to machine translation

Broadly speaking, there are two main ways of doing machine translation which are currently used. They are known as the *transfer* and *interlingual* approaches. Both consist of parsing the source language (SL) in order to obtain a more abstract representation of the text. The process is then reversed for the target language (TL), to obtain the target text from that abstract representation.

The two approaches differ in terms of what this abstract representation is. In an interlingual approach, it takes the form of an *interlingua* common for the languages being translated, which typically consists of some representation of the meaning of the text. If the two representations are different for the two languages (for instance, if they are some kind of syntactic structure tree), then a *transfer* component is needed that maps from the SL intermediate structure into the TL one.

The standard picture for these approaches is as shown in Figure 1.1.

In order to map from the SL text into the TL text, one may follow two paths: one goes via the interlingual representation, the other one via a transfer component.

1.2.1 Interlingual approaches

Interlingual approaches to MT were first proposed in the translational model of [Weaver 49], and later in systems such as CETA ([Hutchins 78]). These involve a parsing component that takes the Source Language text and maps it into an intermediate (interlingual) structure, and a generation component that maps this into the Target Language text. The advantage of using a common interlingua is that for translating between n languages, only n analysis and n generation components are required. The interlingua is some kind of logical representation of the meaning of the text, and the underlying idea is that translation is essentially an exercise in preservation of meaning between the languages. Discourse and pragmatic aspects are carried through if they are taken to be part of this meaning, which must therefore be understood in quite a broad sense.

It follows from this that the interlingua must be a logic powerful enough to represent

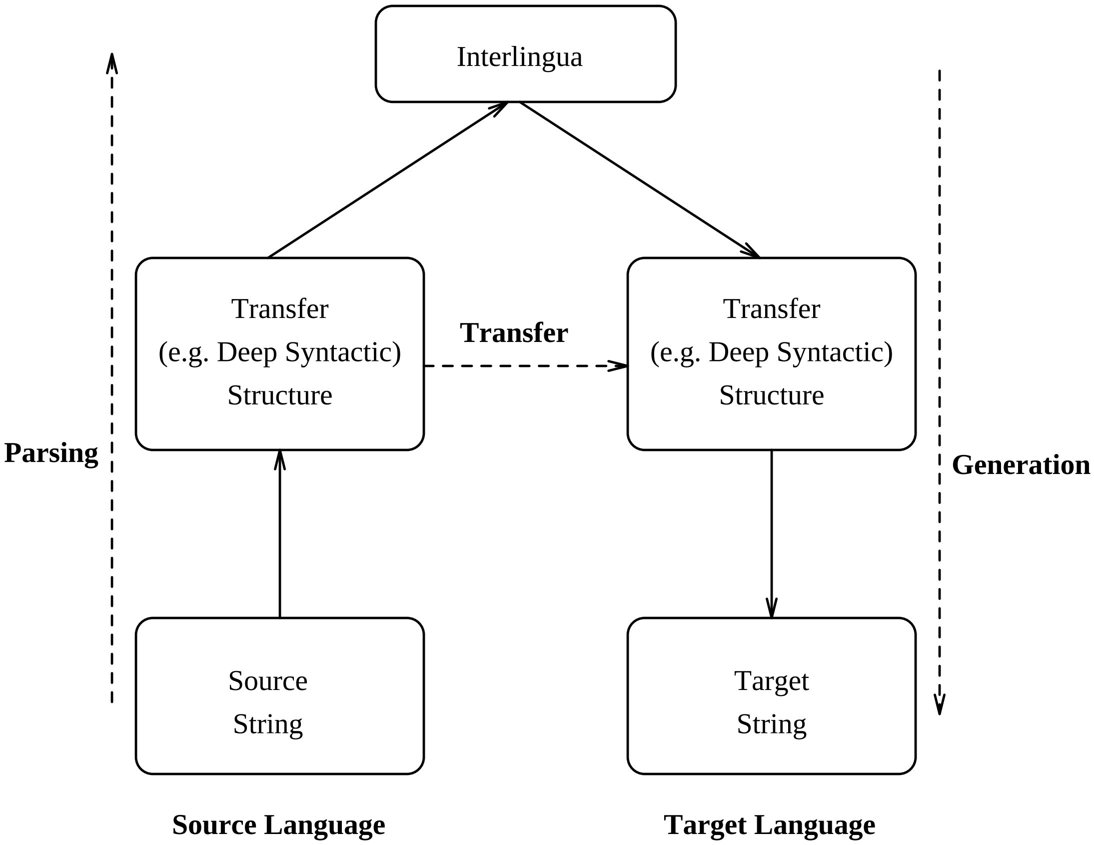


Figure 1.1: Standard Components of an MT System

all possible expressions in all the languages to be translated. This in its turn raises a problem of completeness: if the system is to be robust, it is essential to guarantee that any interlingual formula derived from any SL expression is amenable to generation into the TL. If the interlingua is powerful enough to represent all the meanings in all the languages involved, there will be several (probably infinitely many) formulae in that interlingua which are logically equivalent to the one produced by the analyser. It is then not guaranteed that this formula is one that comes under the coverage or the Target Language generator. What is then required is the power to draw logical inferences in the interlingua, and the complexity of this task may be computationally daunting, since subproblems of this (such as satisfiability, or non-tautology) are known to be NP-complete ([Garey and Johnson 1979]).

1.2.2 Transfer approaches

In a transfer approach, exemplified by systems such as those of GETA in Grenoble ([Boitet et al. 85]) or SUSY in Saarbrücken ([Maas 87]), there are essentially three components involved in the translation for each language pair. An analysis component maps the input string to some representation similar to deep syntactic structure, possibly annotated with discourse and pragmatic information. A transfer component then maps this to a similar structure for the Target Language, and a generator then uses that to produce the TL expression. In general, the analysis and generation modules may be monolingual in that the representations used by the transfer element are independent of the other languages in the translation pair. Consequently, in order to translate between n languages, n generation and analysis components are needed, together with $n(n-1)$ unidirectional transfer components. The transfer component is very much language-pair specific, and must be written bearing very closely in mind both monolingual components in order to ensure compatibility.

Depending on how much work is done by the analysis and generation components, the tasks carried out by the transfer element may vary, but in general this module is very idiosyncratic and will involve several hundred transfer rules. Writing these transfer rules is the most time-consuming aspect of the design of a transfer-based system, as it must be consistent with both monolingual grammars. The process is therefore error-prone, and the result is not very portable, since the consequences of making changes to the monolingual components may be far-reaching as far as the transfer rules are concerned.

1.3 Isomorphic grammars

A recent attempt to get around the problems of transfer and interlingual approaches was pioneered by Landsbergen and others working on the Rosetta system ([Appelo & Landsbergen 86], [Landsbergen 87a, 87b]). They advocate writing isomorphic grammars for the source and target languages, in which a tight correspondence is kept between pairs of grammar rules and between pairs of lexical entries.

In Montague grammars, syntactic rules are paired with semantic operations, and the basic expressions in the language mostly to semantic primitives. Rosetta uses this correspondence to create an isomorphism between the grammar rules of the two languages, which is used for the translation process.

The basic approach of Isomorphic Grammars was taken up by [CJ Rupp 86] and the author ([Beaven 87], see also [Beaven and Whitelock 88]), in the context of GPSG and Categorial Grammar approaches respectively.

1.3.1 The Rosetta system

The Rosetta system was the first attempt at a Machine Translation system based on isomorphic Montague grammars. Montague Grammars ([Dowty et al. 81]) define a correspondence between lexical entries and semantic primitives, and between syntactic rules and semantic operations.

In Rosetta, a translation relation is defined in terms of this correspondence: basic expressions in the two languages are defined to be equivalent to each other if they correspond to the same basic semantic primitives, and grammar rules are equivalent if they correspond to the same semantic operations. The translation relation is then defined recursively on that basis: two expressions stand in the translation relation if they can be derived from equivalent lexical entries by means of equivalent syntactic rules.

A consequence of this is that no transfer module is required. The negative aspect of the approach is that if a SL expression has several translations into the TL expression, then the difference in the meanings of these TL expressions has to be reflected in the SL. Given the tight correspondence between syntactic and semantic rules, this implies different syntactic parses for the SL expression, which might be counterintuitive, since from a monolingual speaker's point of view, it could well be the case that the SL expressions do not appear to be ambiguous. This is a specific instance of the "tuning" between the two monolingual grammars that has to take place. In other words, the grammars cannot be built independently of each other (a fact which the Rosetta researchers are aware of), since SL and TL expressions which are translations of each other must have isomorphic derivations. In fact, according to Landsbergen, writing isomorphic grammars is a way of being explicit about the tuning between the SL and TL grammars, which is always required for reliable MT. While this might be manageable when a small number of languages are concerned, the approach seems highly impractical for a system translating between many languages, as each time a new language is added, all the existing monolingual grammars have to be "retuned" to take the new language into account.

1.3.2 An application of isomorphic grammars to English and Spanish

The ideas of Isomorphic Grammars were also applied in the context of Unification Categorial Grammar (UCG) ([Zeevat et al. 87]) for a small experimental MT system between English and Spanish in [Beaven 87] and [Beaven and Whitelock 88]).

UCG is a grammar formalism in which, like many other recent ones, linguistic objects are represented as sets of attribute-value pairs, called *signs*. The values may be atomic, variables, or other sets of attribute-value pairs. The two standard ways of representing such objects are as directed acyclic graphs (DAGs), or as attribute-value matrices using the PATR-II notation of [Shieber 86].

UCG combines a categorial treatment of syntax with a semantics based on Kamp's Discourse Representation Theory [Kamp 81]. A typical monolingual sign may have, at its top level, features for PHONOLOGY (or more frequently orthography instead), SYNTAX, and SEMANTICS. The grammar rules used for combining such signs are the usual categorial ones, and include at least (backwards and forwards) function application.

In order to write isomorphic UCG grammars, monolingual lexical signs must be paired whenever they stand in the translation relation. This pairing can be thought of as a bilingual lexicon of signs with two attributes: ENGLISH and SPANISH. The values of these were the ordinary monolingual signs. This effectively implemented Landsbergen's notion of pairing lexical entries which are in the *possible translation* relation. The only grammar rule in UCG is functional application, so the correspondence between the grammar rules is straightforward. Since the signs are very rich, it is possible to state at the appropriate level (syntactic or semantic) the constraints required for the particular translation to hold.

From a procedural point of view, the TL syntactic tree is constructed as a "side effect" of parsing. When the SL halves of two bilingual daughter signs combine into the SL half of a mother sign by means of a grammar rule, the TL half of the mother is also constructed from the TL halves of the daughters as prescribed by the grammar.

Parsing and generation therefore work hand in hand, and the combination of a unification formalism and a categorial grammar available in UCG made it possible to build isomorphic derivation trees by means of non-standard constituents, and to state the necessary constraints for the translation relation to hold. This relied to a large extent on the partiality of the signs.

The system developed was tested with a small grammar covering the translation of some temporal expressions between the languages, where the translations of tenses depended on aspectual information of the verbs concerned. It was possible to capture these aspectual constraints concisely in the bilingual signs using the facilities provided by UCG, such as templates and lexical redundancy rules.

1.4 Lexically-driven machine translation

Although isomorphic grammars constitute a step in the right direction of portability and robustness, they still have problems which we feel are insurmountable.

The idea of developing the translation upwards from the lexicon is a good one, but attempting to do so by means of isomorphic rules is doomed to failure because it requires that the phrases built out of equivalent subphrases or lexical entries be translation equivalents of each other. In other words, if two sentences stand in the translation relation, they must have syntactic derivation trees with the same structure, though not necessarily the same linear ordering of the branches.

It is possible to design grammars (particularly on categorial-based approaches) which make constituents out of virtually any substring. However, if the two languages being translated have widely differing word ordering, it may be simply impossible to find any continuous substring of one sentence which translates as a continuous substring in the other. As an example, consider the well-known Dutch cross-dependency construction, as illustrated in (1.1):

(1.1)	(omdat)	Jan	Piet	de	kinderen	zag	helpen	zwemmen
	(because)	John	Pete	the	children	saw	help	swim
	‘ (because) John saw Pete help the children swim’							

A close inspection will convince the reader that the English continuous constituents do not correspond to continuous strings in Dutch. This means that in order to preserve the isomorphism of the grammars, we have to do away with the notion of having adjacent constituents, which has possibly disastrous consequences from a computational point of view.

Although some work has been done in the direction of creating formalisms that allow such constituents ([Reape 89, 90, forthcoming], [Manandhar forthcoming]), it is somewhat early to assess their success, and another approach is advocated here, namely to preserve the lexical foundation while doing away with the attempt to keep the grammars isomorphic.

1.4.1 Defining the translation relation from the lexicon upwards

In this work, we shall examine the consequences of carrying the ideas underlying many current unification-based formalisms, as represented in one such (namely UCG), over to the field of machine translation. The key point is that this will make it possible to have an MT system in which no meaningful elements in the translation relation are introduced in the form of transfer rules or operations with interlingual representations. In particular, assuming we have very rich lexical entries (which contain information about various dimensions of the language, such as orthography, syntax and semantics), all that is needed is a correspondence between the lexical entries, supplied by a bilingual lexicon.

The design of such a translation system will therefore involve three components: two monolingual lexicons for the languages concerned, and a bilingual lexicon putting these into correspondence. Very little else is required: the grammar rules are few and simple, as is usual in categorial grammars, and in order to make the lexicons more compact, there are lexical redundancy rules for the formation of closely-related lexical items (for instance, in order to account for dative transformations in English) and lexical templates, which are under-specified signs useful for building concise lexical entries. It is claimed that the two monolingual components may be written with great independence from each other, and the monolingual grammar writers just need to concern themselves with providing an adequate coverage of their own language.

1.4.2 Structure of the bilingual lexicon

The bilingual component of the system consists of a bilingual lexicon, which puts into correspondence pairs of monolingual lexical entries. In other words, each entry in the bilingual lexicon will contain a pair of pointers to monolingual entries in each of the languages translated, or to complex expressions made from these. These monolingual entries are very rich signs, and the bilingual entries may add constraints for their monolingual signs to be in the translation relation. For instance, if a word has more than one translation depending on how various semantic features become instantiated, the bilingual lexical entries may express these restrictions.

Ideally, the information in the bilingual lexicon should be very similar to what one expects to find in a conventional bilingual dictionary, namely correspondences between pairs of words or expressions, with perhaps some restrictions (on the meaning, usually supplied by the context) as to when a particular correspondence holds.

The bilingual lexicon writer needs to be aware of what the monolingual lexicons look like, but only in order to know what correspondences need to be built, and how the extra restrictions that the bilingual sign imposes on the monolingual entries will be encoded. As will be seen in the examples in this work, as long as some broad conventions are followed, this task becomes very straightforward. Most bilingual correspondences are very simple, and merely require some semantic indices in the monolingual signs to be unified. As long as these indices are made easily available in predictable places within the monolingual signs, the task of writing the corresponding lexical entries is very simple. When some semantic constraints need to be put on these indices, again it is a straightforward task. It is only on the comparatively rarer occasions when syntactic constraints have to be included that the monolingual signs need to be examined more closely, in order to determine how that syntactic information is encoded.

This results in a great modularity in the system. Any monolingual component may easily be changed, without affecting to any significant extent the bilingual lexicon, and certainly not the monolingual components for any other language. At the same time, the simplicity of the bilingual component makes it practicable to write multi-language systems, since all the hard work goes into the monolingual lexicons which may be re-used for many language pairs, and the language-pair-specific information is concisely kept in the bilingual lexicon.

1.5 Objectives

This work sets out to describe the implementation of such a lexically-based bidirectional Machine Translation system between English and Spanish, in order to investigate the feasibility of such an approach for a larger-scale application. Since in this context it only possible to develop a small toy system, we shall take a small number of phenomena which are problematic, and examine how they can be treated. These broadly consist of Spanish word order and clitic behaviour.

The first task involved is to write a monolingual UCG for Spanish. Because UCG was originally developed mainly with English in mind, a few important changes are needed to adapt it for Spanish, the main features of which are a greater word order freedom than that existing in English, and the possibility of subject pro-drop.

The first of these changes involves the treatment of noun phrases. Spanish NPs are treated as sentence modifiers, in a manner similar to that proposed in [Whitelock 88] for Japanese. This ties in with a Neo-Davidsonian treatment of the semantics ([Dowty 89]), and is motivated syntactically by the Spanish subject pro-drop. The second main change is a case-assignment mechanism built "on top" of UCG for reasons which will be discussed in Chapter 3.

A new algorithm for translation, developed with Pete Whitelock and Mike Reape, and known as **Shake-and-Bake**, is then presented. It can be outlined as follows: first of all the SL expression is parsed using the SL (monolingual) grammar. After the parse is complete (and hence some of the local ambiguities resolved, and the semantics instantiated), the lexical entries are looked up in the bilingual lexicon and replaced with their TL equivalents. Generation then takes place starting from the set of TL lexical entries.

Two well-known parsing algorithms (shift-reduce and CKY) have been adapted to do this kind of generation instead. Generation in this context can be seen as a variation of parsing, in which we let the syntactic constraints instantiate the word order rather than letting the word order drive the parsing process.

1.6 Outline

Chapter 2 presents a brief sketch of Spanish grammar, and in particular the aspects which will concern us in this work. It is intended to be an uncontroversial description of some basic facts such as word order, phrase structure and so on. The interesting behaviour of clitics, such as their relation with word ordering, and phenomena such as clitic doubling and clitic climbing, are also described. Finally topicalisation and dislocation are discussed.

Chapter 3 develops a UCG treatment for the data presented in Chapter 2. It starts with a brief introduction to UCG, and then proceeds to describe the modifications introduced to it in order to adapt it to the freer Spanish word order. The main one is a case assignment mechanism, and its interaction with the thematic roles which the semantics involves. Once the definitive version of the formalism is described, it is used for giving accounts of the phenomena described in Chapter 2, namely freedom of word order, clitic behaviour, dislocation and topicalisation.

Chapter 4 presents the UCG grammar that will be used for English. This is a very conventional grammar, and hence quite different from the Spanish one of Chapter 3. This is done deliberately in order to illustrate one key advantage of the Shake-and-Bake approach, namely that the grammars can be written quite independently of each other. The coverage is similar to the Spanish one: word order, clitic behaviour, topicalisation and dislocation, and clefts (since they often correspond to Spanish dislocated constructions).

Chapter 5 is an introduction to how to do MT with these grammars. It describes the bilingual lexicon, and the basic mechanisms for Shake-and-Bake: parsing, bilingual lookup, and generation or *baking*.

Chapter 6 discusses some implementation issues involved in Shake-and-Bake in greater detail. It discusses two approaches for parsing (namely shift-reduce and CKY), describes bilingual lookup in detail, and presents two algorithms for baking, one based on a Shift-Reduce parser, the other based on CKY. A section on semantics discusses the treatment of correspondence in the bilingual lexicon, and of semantic typing. Other question that arise in the implementation is what to do when two expressions standing in the translation relation have different numbers of words, and what the computational complexity of the approach is.

In Chapter 7, the treatment of morphology in the current framework is briefly discussed, and translations are presented for the phenomena covered in Chapters 2 and 3 for Spanish, and in Chapter 4 for English. We shall see sample sentences translated (in both directions) which cover the interesting aspects of word order, clitic behaviour, topicalisation, dislocation, clefts, and two final examples which cause some difficulties for many existing approaches (argument switching and head switching), but are remarkably straightforward in Shake-and-Bake.

Chapter 8 concludes with some remarks about some efficiency considerations and possible improvements.

Finally, the Appendices show some entries for the small system built, and demonstrates a sample run with it.

Chapter 2

A brief sketch of Spanish grammar

In this chapter I intend to present a relatively uncontroversial description of Spanish grammar, which will serve as an introduction to Chapter 3, where I shall give a computational account of the data presented here.

As the computational grammar formalisms of Chapter 3 have been mostly used for writing grammars of English I shall concentrate mainly on the differences between Spanish and English.

The main features of Spanish grammar that will be covered here will be word order, subject pro-drop, clitics, and dislocated NPs.

This chapter will follow standard descriptive textbooks such as [Bello 84], [Gili y Gaya 69], and [Marcos Marín 80]. Much of the information about word order has been taken from [Foster 90], who compiled data from some of the above sources and others, as well as from his own informants.

2.1 Essentials: word order, phrase structure, etc.

2.1.1 Subject pro-drop

Spanish verb inflections carry information about the tense, and also about the person and number of the subject, and the subject does not need to be explicitly present, as long as the discourse context and the form of the verb makes its referent clear. The conditions are similar to those that licence the presence of a subject pronoun in English.

For instance, the following sentence is well-formed. It is multiply ambiguous in isolation, but is perfectly acceptable as long as the context makes it clear what the subjects of the two verbs are.

(2.1)	No	sabía	qué	quería.
	Not	know-1or3sg-past	what	want-1or3sg-past
	'I/she/he did not know what I/she/he wanted.'			

The subjects of the two verbs need not be co-referential, and any possible ambiguity could be eliminated by having lexical NPs or pronouns as subjects.

Dummy subjects do not exist in Spanish:

(2.2)	a	Llueve.			
		Rain-3sg-pres			
		'It is raining.'			
	b	Es	necesario	que	vengas.
		Be-3sg-pres	necessary	that	come-2sg-pres
		'It is necessary that you come.'			

2.1.2 Order of constituents

The ordering of Spanish constituents appears to be fairly free. Although there is a canonical word order for main declarative sentences (SVO), according to [Gili y Gaya 69] every ordering is possible, as long as the verb is in first or second position. Of course, the different orders will convey emphasis or focus on different elements of the sentence. But the ordering is still essentially very free. The examples in [Gili y Gaya 69] (p.83) are:

(2.3) *Subject, Verb, and Direct Object:*

a	Mi	padre	compró	una	casa.
	My	father	buy-3sg-past	a	house

‘My father bought a house.’

b *Mi padre una casa compró.

c Compró mi padre una casa.

d Compró una casa mi padre.

e Una casa compró mi padre.

f *Una casa mi padre compró.

(2.4) *Subject, Verb, and Circumstantial Complement:*

a	Juan	vendrá	a	las	siete.
	John	come-3sg-fut	at	the	seven

‘ John will come at seven.’

b *Juan a las siete vendrá.

c Vendrá Juan a las siete.

d Vendrá a las siete Juan.

e A las siete vendrá Juan.

f *A las siete Juan vendrá.

(2.5) *Verb with two complements:*

a	Una	carta	traigo	para	ti.
	One	letter	bring-1sg-pres	for	you

‘I am bringing a letter for you’

b *Una carta para ti traigo.

c Traigo una carta para ti.

d Traigo para ti una carta.

e Para ti traigo una carta.

f *Para ti una carta traigo.

He comments (§69, p.83, my translation):

> ...it is easy to observe that, although all these combinations are possible and correct in modern Spanish, those marked with an asterisk are totally unused in conversation and rare in literary prose. Their usage gives the style a marked pedantic affectation. They are used somewhat more in poetry, particularly that of the classical period and the 19th Century. All have the

> verb at the end, which was precisely the location preferred by Latin writers *(Caesar Gallos vicit)* until the end of the Republic.

The restriction is not exactly that the verb may not be in final position, but may not be after the second position, as shown later (§73, p. 89), in the following examples:

(2.6) a El criado trajo una carta para mí.
The servant bring-3sg-past a letter for me
'The servant brought a letter for me.'
*[and the other 23 permutations, of which the 12 with the V in 3rd and 4th positions are marked with a *]*

Gili y Gaya comments (§73, p.88, again my translation):

> Keeping the sentence as a unit, it is clear that the twelve combinations marked with an asterisk are out of ordinary modern usage, although they may be found in poetry and in particularly affected style. In these twelve instances, the verb occupies the third or fourth place, and the tendency to divide in two [intonation units] is visibly favoured in them. Similarly to what was observed in the sentences formed by three elements, the verb may not, without affectation, move beyond the second place. We may repeat the same observation in a sentence with five elements ...

Many informants are less reluctant than Gili y Gaya to accept verb-third constructions, but those with the verb in fourth position seem to be generally out.

It is not clear that Gili y Gaya's examples involved an indirect object NP rather than some other complement PP. In fact, the distinction between a dative-marked NP and a PP is somewhat blurred. I shall take the impossibility of having it replaced by a clitic pronoun as evidence that *para mí* in (2.6) is indeed a PP. Nevertheless, the claim still holds, since the permutations of the following sentence, which clearly involves a IO NP, seem to give the same results.

(2.7) Carlos regaló un libro a María.
Charles give-3sg-past a book to Mary
'Charles gave a book to Mary'

2.2 Clitics

The most common direct and indirect object pronouns are unstressed and have to be immediately adjacent to the verb to which they attach. Such pronouns are called clitics. When they follow the verb, the clitic(s) and the verb are written as a single word. However, for the sake of clarity, in the following examples they will be separated from the verb with hyphens.

The paradigm for clitics is as follows:

	Singular		Plural	
	Masc.	Fem.	Masc	Fem.
First Person				
Obj.	me		nos	
Second Person				
Obj.	te		os	
Third Person				
Direct Obj.	lo	la	los	las
Indirect Obj.	le / se		les / se	

The form *se* replaces the indirect object *le* or *les* when it is followed by a direct object clitic, as in:

(2.8)	*Le	lo	di.
	Se	lo	di.
	CL-3-dat	CL-3sg-acc	give-1sg-past
	'I gave it to him/her.'		

2.2.1 Clitics and word order

The ordering of the clitics relative to the main verb is very straightforward. Except when clitic climbing is involved (see Section 2.4), the clitic must be immediately adjacent to the verb of which it is the complement, and nothing may intervene in between. In modern Spanish the clitic will precede the verb (proclitic) if it is in a finite form, otherwise it will follow it (enclitic), in which case it is written as one single word.

If a dative and an accusative clitic are both present, the dative always precedes the accusative, irrespective of whether they are proclitic or enclitic.

(2.9) a Da-me.
give-2sg-imperative-CL-1sg-obj
'Give to me.'

b Diciéndo-te-lo.
saying-CL-2sg-obj-CL-3sg-masc-DO
'Saying it to you.'

c Te lo dijo.
CL-2sg-obj 3sg-masc-DO say-3sg-past.
'He/she said it to you.'

d *Te lo siempre dijo.
CL-2sg-obj 3sg-masc-DO always say-3sg-past

e Siempre te lo dijo.
always CL-2sg-obj 3sg-masc-DO say-3sg-past

f Te lo dijo siempre.
always CL-2sg-obj 3sg-masc-DO say-3sg-past

g *Te lo no dijo.
CL-2sg-obj 3sg-masc-DO not say-3sg-past

h No te lo dijo.
not CL-2sg-obj 3sg-masc-DO say-3sg-past

The comments in the previous section about the word order within the sentence do not seem to apply if there are any clitics present: when it was said that the verb may not occur after the second position, any clitics that may precede the verb do not count, and neither do adverbs or adverbial phrases. This is explained and illustrated in Gili y Gaya (§74, p.89).

> In all the examples mentioned in this chapter, parts of speech with their own stress have been deliberately chosen. Unstressed pronouns, and in general, words and phrases which may easily be proclitic with relation to the main intensity stress of the group, allow the verb to occur in common language beyond the second place in the sentence. Examples:
>
> (2.10) a Nada me DIjo aquel día.
> nothing CL said-3sg that day
> 'He/she did not say anything to me that day.'
>
> b La casa a todos nos ha pareCIdo demasiado cara.
> the house to all CL has seemed too expensive
> 'The house seemed too expensive to all of us.'
>
> c El chico pruebas me ha DAdo de su capacidad.
> The boy proofs CL has given of his ability
> 'The boy has proved his ability to me.'

In these three sentences the main intensity stress is in the syllables written in upper case.

Note how the second example above (2.10 b) has two lexical NPs preceding the verb, thereby contradicting the claim in (2.6). The doubled clitic (*nos* co-occurring with the NP *a todos*) certainly plays some role here: the sentence without it is worse, but this could be attributed to the general preference for IO clitic doubling. The example however suggests that there is some grey area when the verb is in penultimate position, and this seems to be confirmed by several informants. In the third example, the head of the object NP also appears before the verb. One possible reason is that the expression *ha dado pruebas de* is somewhat stilted, which makes the verb in third position less strange.

It should also be pointed out that of the constructions where the object is fronted (OVS), those in which the object is indefinite are preferable to those in which it is not.

(2.11)	a	Una	casa	compró	mi	padre.
		A	house	bought	my	father.
	b	? La	casa	compró	mi	padre.
		The	house	bought	my	father.

The same holds if there is an indirect object present: in general, direct objects may only be fronted if they are indefinite:

(2.12)	a	Una	casa	regalaron	mis	padres	a	mi	hermana.
		A	house	gave	my	parents	to	my	sister.
	b	? La	casa	regalaron	mis	padres	a	mi	hermana.
		The	house	gave	my	parents	to	my	sister.

The only part of the data in (2.6) that seems certain is that of all these sentences consisting of the permutations of a ditransitive verb, its two objects and its subjects, the constructions in which the verb comes after its lexical subject and object(s) are excluded.

2.2.2 The particle "a" with personal direct object: *leísmo* and *laísmo*

When a direct object refers to a person, it must be marked by the particle *a*, which will be regarded as some kind of "animacy" marker rather than a preposition, following

[Suñer 88]. Thus:

(2.13) a Vi a Juan.
Saw-1sg to John.
'I saw John'

b *Vi Juan.

This straightforward fact has however a couple of important consequences. The *a*-marked DO looks like an indirect object (since *a* is also the standard IO marker). This affects the choice of clitic that may replace it: although the "purist" choice would be to use the DO clitic, as in example (2.14 b) below, the IO clitic is very widely used in all dialects (2.14 c). This occurrence is called *leísmo*.

(2.14) a Vi a Juan.

b Lo vi.

c Le vi.

A similar phenomenon, which arises from the lack of a masculine/feminine distinction in the dative third person clitic, has caused in some dialects *le(s)* to have become specialised as a masculine form, and made *la(s)* evolve as a feminine form. Thus instead of the standard usage as in (2.15 b) below, (2.15 c) is found. This is called *laísmo* and is somewhat stigmatised in educated Spanish. It is frequent in the spoken language of Madrid and Buenos Aires, amongst others.

(2.15) a Regalaron una bicicleta a María.
Gave-3pl a bicycle to Mary
'They gave Mary a bicycle'

b Le regalaron una bicicleta.

c ??La regalaron una bicicleta.

(2.16) a Y al preguntar-la me dijo: es que hueles a pellejo.
'And when I asked her, she said to me: it's just that you smell of rawhide.'
(Tango by Antonio Bartrina and Oswaldo Larrea [Malevaje 88])

The much rarer, but similar, *loísmo* involves the use of *lo(s)* as a IO masculine form:

(2.17) No lo he dado ninguna importancia.
Not CL-do-3sg have-1sg given any importance
'I have not given it any importance.'

2.2.3 Clitic doubling

In standard Spanish, the DO clitic and corresponding NP normally occur in complementary distribution:

(2.18) a María leyó el libro.
Mary read-3sg-past the book
'Mary read the book'

b María lo leyó.
Mary DO-CL read-3sg
'Mary read it.'

c *María lo leyó el libro.

The only way they can co-occur is if there is an intonation break between the object appearing in first position and the rest of the sentence. We shall refer to such cases as *dislocation*, and they will be covered in the next section.

The IO clitic, however, behaves differently: not only may it co-occur with the corresponding NP (as in (2.19 c) below), this is often preferred:

(2.19) a María dio el libro a Juan.
Mary gave-3sg the book to Juan.
'Mary gave the book to Juan.'

b María le dio el libro.
Mary IO-cl gave -3sg the book
'Mary gave him the book.'

c María le dio el libro a Juan.
Mary IO-cl gave -3sg the book to Juan

This will be referred to as (IO) clitic doubling. DO doubling, which exists in some Spanish dialects such as that of Buenos Aires, and under certain conditions on the NP (see [Suñer 88] for details and a thorough account), will not be covered here in any detail. It will suffice to mention that these restrictions on the NP involve its determinacy, specificity and so on.

[Jaeggli 86] offers an explanation of the data about co-occurrence of clitics and NPs in a Government and Binding framework. According to his analysis for Spanish, the DO clitic always absorbs case. Any DO NP present will therefore fail to receive a case, and the sentence is then be ruled out by the *case filter*. It follows that the occurrence of the clitic is complementary with that of the NP. Thus (2.18 a & b) above are allowed, but (2.18 c) is excluded: once case has been assigned to the clitic, it can no longer be assigned to the NP. The IO clitic, however, is only an *optional* case absorber, and hence may or may not be doubled ((2.19 a, b &c) above): if the verb does not assign case to the clitic, the NP in IO position is licensed, as it will receive case. The treatment can be extended to those dialects of Spanish (see [Suñer 88]) which allow DO doubling under certain conditions. It is just necessary to posit that under these conditions, the DO clitic is also an optional case absorber like the IO clitic.

In contrast, all Italian and French clitics always absorb case, and thus may never co-occur with the NP in a single intonational pattern:

(2.20)	a	Pierre	lui	a	donné	des	bonbons	(*à	Marie).
		Peter	IO-cl	has	given	DET	sweets	(*to	Mary)
		'Peter gave her sweets (*to Mary).'							
	b	Carlo	gli	ha	telefonato	(*a	Gianni)	ieri.	
		Charles	IO-cl	has	telephoned	(*to	John)	yesterday	
		'Charles telephoned him (*John) yesterday.'							

2.3 Topicalisation and dislocation

The facts about word order discussed in Section 2.1 above assume there is only one intonational phrase. There are two mechanisms that license word orders other than the canonical SVO, namely topicalisation and dislocation.

The exposition of these subject follows [Sanfilippo 90b&c] [1].

By topicalisation it is meant that an element of the sentence (the topicalised element, which will be written in CAPITALS) is stressed, and appears at the beginning of the

[1] I am grateful for Antonio Sanfilippo for his discussions of this subject with me. Any errors or misrepresentations remain, of course, my own.

sentence but without being separated from the rest by an intonational break. In GB terms, topicalisation is a form of extraction, and the usual rules about co-occurrence of clitics and their corresponding NPs applies.

From a discourse point of view, topicalisation roughly corresponds to the English *it*-cleft: it would be used for instance when the topicalised element needs to be corrected, or stressed for any other reason.

(2.21) a EL LIBRO compré esta tarde (no el disco).
The book bought-1sg this afternoon.
'It was the book that I bought this afternoon (not the record).'

b *EL LIBRO lo compré esta tarde.

c A CARLOS Luis conoció ayer (no a María).
To Charles Lewis met yesterday
'It was Charles that Lewis met yesterday (not Mary.)'

Dislocated elements (which will be written in **boldface**), on the other hand, are separated from the rest of the sentence by an intonational break. Dislocation differs from topicalisation and other extractions in that the dislocated object may co-occur with its corresponding clitic.

(2.22) **El libro,** lo compré esta tarde.
'The book, I bought it this afternoon'

Dislocated elements behave more like sentential adjuncts.

Dislocation may be like other extraction phenomena in obeying the **Complex NP Constraint** (2.23 a), but it differs from them in that it can relate a verb and one of its arguments across two *wh*-islands, as in (2.23 b), and it allows multiple displacements and crossed dependencies as in (2.23 c) and (2.23 d).

(2.23) a ? **Carlos,** María conoce al periodista que lo entrevistó.
Charles, Mary knows to the journalist who CL interviewed
'Charles, Mary knows the journalist who interviewed him'

b **Carlos,** ¿ quién sabe quién lo ha visto?
Charles, who knows who CL has seen
'Charles, who knows who has seen him?'

c **A Juan, Carlos,** se lo ha presentado María.
To John Charles CL CL has introduced Mary.
'Mary introduced Charles to John'

d **Carlos, a Juan,** se lo ha presentado María.

2.4 Clitic Climbing

We now turn to data involving clitic climbing, as illustrated in (2.24).

(2.24) a Quiere leerlo
Want-3sg read-DO-cl
'He/She wants to read it.'

b Lo quiere leer.
DO-cl want-3sg read

c Quiere poder leerlo.
Want-3sg can read-DO-cl
'He/She wants to be able to read it.'

d Quiere poderlo leer.

e Lo quiere poder leer.

When there is such a sequence of verbs in Spanish, the clitic which is thematically marked by the lowest (rightmost) verb (2.24 a) may "climb" up to a higher (leftmost) verb, as in (2.24 b). When there are more than two verbs as in (2.24 c), the clitic may climb to the middle position (2.24 d) or all the way up to (2.24 e). At any rate, the clitic will be immediately adjacent to the verb to which it attaches, preceding it if it is finite, and following it otherwise. In general the clitic in lower position is slightly preferred.

The clitic is capable of climbing over any adverbs or negation:

(2.25) a Quiero poder no verla.
Want-1sg can not see-CL-do-3sg
'I want to be able not to see her.'

b Quiero poderla no ver.

c La quiero poder no ver.

d Quiero no poder verla.
'I want to be unable to see her.'

e Quiero no poderla ver.

f La quiero no poder ver.

(2.26) a Quiero poder siempre verla.
Want-1sg can always see-CL-do-3sg
'I want to be able to see her always.'

b Quiero poderla siempre ver.

c La quiero poder siempre ver.

d Quiero siempre poder verla.
'I want to be always able to see her.'

e Quiero siempre poderla ver.

f La quiero siempre poder ver.

As a final observation, if the higher verb has an IO third person clitic (*le*) and the lower verb a DO clitic, should the DO clitic climb and arrive next to the IO clitic, the alternative form of the IO clitic *se* is required.

(2.27) a Le mandé comprar-lo.
CL-io-3sg ordered-1sg buy-CL-do-3sg
'I ordered him/her to buy it.'

b *Le lo mandé comprar.

c *Se lo mandé comprar.

2.5 Summary of data to be accounted for

2.5.1 Word order without clitics

In a sentence without clitics, all word orders except verb-final are allowed. There is a further restriction that if the direct object appears before the verb, it must be indefinite.

Thus we have the following orderings, where curly brackets indicate that any permutation of the elements within them is possible:

(2.28)

Simple Transitive Sentences	*Simple Ditransitive Sentences*
SVO	SV$\{$OI$\}$
V$\{$SO$\}$	V$\{$SOI$\}$
O[indef]VS	O[indef]V$\{$SI$\}$
	IV$\{$SO$\}$
	$\{$SO[indef]$\}$VI
	$\{$SI$\}$VO
	$\{$O[indef]I$\}$VS

2.5.2 Clitics

If any of the objects of the verb are clitics, they are not covered by the account given above. Thus, for example, if the DO is a clitic, the SOV ordering is fine.

In sentences with a single intonational contour, DO clitics are in complementary distribution with their corresponding NPs. In some spoken dialects (such as *Porteño* from Buenos Aires, or that of Madrid), NPs which are [+spec, +animate] may co-occur with the clitic, but these cases will not be considered here. IO Clitics may co-occur with their NPs, and this is in fact sometimes preferred.

Finally, in constructions with a sequence of verbs, the clitic that normally attaches to the "lower" verb may "climb" and attach to the "higher" verb.

Chapter 3

A computational grammar for Spanish

3.1 A brief introduction to UCG

This chapter sets out to outline the basic elements of a computational grammar for Spanish based on Unification Categorial Grammar (UCG) [Zeevat et al. 87], [Calder et al. 88] and Head-driven Phrase Structure Grammar (HPSG) [Pollard and Sag 87]. As the treatment will be based mostly on UCG, it seems appropriate to present the formalism at this stage.

UCG combines a categorial treatment of syntax with semantic insights from Discourse Representation Theory [Kamp 81]. Linguistic entities are represented as **signs**, which are bundles of feature-value pairs, and may be represented as PATR-style feature matrices [Shieber 86].

UCG signs consist of the following features:

- ORTHOGRAPHY.
- SYNTAX: this is based on Categorial Grammar. The value may be either an atomic category (S, N, and NP are used), or a complex category of the form A/B, where A is a category and B is a sign. In the usual Categorial Grammar notation, that means the sign in question is a functor that may combine with an argument sign non-distinct from B by means of function application, to yield a sign with syntactic

category A (the "slash" is order-free: the direction of the combination is specified by the ORDER feature). The unification of B with an existing sign allows the information to flow between argument and functor.

- ORDER: this feature regulates linear ordering of constituents. The value may be *fwd* or *back*[1], depending on whether, if that sign is an argument, its functor precedes it or follows it, respectively. Thus, a sign with syntactic category A/B combines to its right with a sign B with ORDER *fwd*, or to its left with with a sign B with ORDER *back* to produce a sign with category A. If the ORDER of B is uninstantiated, the combination may occur on either side. Either the functor or the argument sign may show the direction of the combination, and these directions must unify. The value of the ORDER feature may be referred to in a bilingual lexical entry, such as Spanish adjectives which have different meanings depending on their position. Compare, for instance, *un pobre hombre (an unfortunate man)* vs. *un hombre pobre (a destitute man)*, or *una buena parte (a large part)* vs. *una parte buena (a good part)*. Although most Spanish adjectives have their ORDER feature uninstantiated, these examples would use two entries with specified ORDER features and different semantics.

- SEMANTICS: the semantic representation language is derived from Discourse Representation Theory [Kamp 81], with a Davidsonian treatment of verb semantics [Davidson 67]. Essentially, a formula involves an index, (introducing existential quantification) together with a list of propositions that may involve that index as well as other indices.

A very simple sign may therefore look like the following one for the verb *canta* (to sing), in its third person singular form of the present indicative.

$$(3.1)\quad \begin{bmatrix} \textsc{ortho} & \text{canta} \\ \textsc{cat} & \text{s} \\ \textsc{sem} & \text{E} : \{\text{cantar(E),role(E,agt,X3)}\} \end{bmatrix}$$

[1]The values *pre* and *post* are normally used in other versions of UCG. I have renamed them *fwd* and *back* as these names seem more meaningful, being related to *forward* and *backward* Function Application (see later in this chapter)

Signs are combined by means of binary rules. UCG has forward and backward function application only. Unlike many other categorial grammars it does not have composition, since many of the effects of composition are achieved by unification and having NPs type-raised in the lexicon. In addition, there are lexical redundancy rules to build related but different signs, as well as templates that factor out the information common to several lexical entries of the same category.

3.2 Basics

The phenomena that will be covered by the computational grammar presented in this chapter were outlined in Chapter 2. In particular, subject-drop and clitic placement, doubling and climbing, which were presented there, will be accounted for. A mechanism for implementing case assignment and thematic role marking will be introduced to handle them. This will involve minor modifications to UCG.

One aspect of clitics that will be treated is their co-occurrence with the corresponding NPs. [Jaeggli 86] explains this in terms of the case absorption properties of the various clitics in different languages and dialects, within the theory of Government and Binding. A mechanism will be presented that allows UCG to describe this process, by providing a way to associate case to NPs, while handling thematic roles in a neo-Davidsonian treatment of semantics ([Dowty 89]). The treatment of the Spanish data will be extended to other Romance languages.

This approach bears many similarities to that of [Sanfilippo 90a,b], although the coverage and the implementation are somewhat different.

3.2.1 NPs as sentence modifiers

An elegant way of giving a categorial grammar treatment of languages in which certain NPs may be omitted (subject in the case of Spanish) consists of treating tensed verbs as sentences, and the NPs that may be dropped as sentence modifiers [Whitelock 88]. Hence the category assigned to a tensed (intransitive) verb is S, and that of a subject NP is S/S (and thus other NPs will also have category S/S). Having NPs as functors over the verb is in line with the UCG treatment, which, following Montague and others [Dowty et al. 81], uses type-raised NPs in order to give a reasonable treatment of quantified NPs.

It may seem at first that having NPs as S/Ss results in an unreasonably free word order, since either all NPs have to find the rest of the sentence which they modify (i.e. the "argument" S in a S/S) on the same side, which is too restrictive, or the order of the rest of the sentence is uninstantiated, which is completely unrestrictive. This is not the case: if a verb requires an obligatory object, it will have to subcategorize for it. In other words, it will have category S/(S/S). This, together with the use of the ORDER feature in the signs, may be used to put constraints on the linear ordering of the words. Examples of this will be shown below. Nevertheless, having NPs as sentence modifiers raises a few questions. First of all, if subjects are assigned the category S/S, what is to prevent an arbitrary number of them from combining with the sentence? Second, what is the difference between a subject NP and a "usual" sentence modifier? These two issues are addressed by the case assignment mechanism.

3.2.2 Case assignment and thematic roles

In order to implement the explanation given in [Jaeggli 86] and outlined in the previous chapter, some additions will be required to standard UCG, which essentially involve adding a case theory to the formalism, since Jaeggli's account relies on case assignment to explain why certain clitic-NP co-occurrences are licensed, while others are banned.

As case-assignment is not normally part of UCG, some justification for adding it seems in order. We shall start with a short review of UCG as originally developed for English.

Following the Montagovian tradition ([Dowty et al. 81]), UCG type-raises its NPs (assigning them category S/VP), making them functors over VPs, and treats transitive verbs as functors over their objects (thus assigning them category VP/(S/VP)). Ditransitive verbs are arguments over two such objects.

This is somewhat restrictive, in that there is no straightforward way of stating that an argument is optional: a sentence is not complete if it still subcategorizes for something. For instance, a tensed transitive verb will still subcategorize for its object NP, and so is not a complete sentence. The formalism could be modified to allow functors to mark some of their arguments as optional, and rules introduced to allow them to "discharge" such arguments.

The alternative proposed here, which uses case theory, involves using two mechanisms in conjunction. The subcategorization list contains what can be though of as "proper" (compulsory) arguments. In addition, a verb may assign cases to NPs. Any NP in that is subcategorized for will have a case specified. In addition, verbs can assign case to other NPs, which do not have to be present for the sentence to be well-formed. However, if they are present, they will absorb one of the cases that the verb may assign. This will allow us to account for optional subject NPs, and to prevent multiple "subject" NPs, since the verb may only assign nominative case to one of them, and all NPs must absorb case.

This in itself is not a great advantage over the alternative of having optional arguments in the subcategorization list. What is a more important issue is that building a case theory "on top" of UCG allows us to translate Jaeggli's account of the clitic doubling data, which is generally accepted as standard in the generative linguistics literature, into our formalism.

In the next section, we shall see in detail how this case theory is implemented in the form of extra features on the signs and unification, thus adding no formal complexity to the formalism.

3.3 The formalism

UCG represents linguistic entities as sets of feature-value pairs, or *signs*. Linguistic information associated with different signs may be linked by equality constraints, typically implemented in parsers by using unification. The features associated with a sign are ORTHOGRAPHY, SYNTAX (categorial), and SEMANTICS.

This framework is extended to include the following additions:

- Case assignment (for the "arguments" that may be present), which will ensure that no two NPs receive the same case. (Obligatory arguments are still subcategorized for: they are in the SLASH list.)

- The verb sign relates NPs with a given case to thematic roles.
- Further "syntactic" features (see below).

Signs will therefore have the following feature-value pairs:

- ORTHOGRAPHY: this might be occasionally annotated with phonological features.
- CAT: the "result" category.
- SLASH: the (ordered) list of "arguments". This is similar to HPSG's SUBCAT feature, and to the treatment of French grammar given in [Bès & Gardent 89]. The arguments appear in the order in which they must combine with the functor (so the first item in the list will be the first to combine). Of course, different possible orderings are often available in Spanish, but these involve changes in the semantic representation obtained, as well as in the intonation, which is not represented here but could be added at a later stage. How the alternative orderings are obtained will be discussed below, but for the time being it will suffice to say that they are obtained with lexical redundancy rules that achieve effects like "scrambling", by producing new lexical entries from existing ones, in which the order of the elements of this SLASH list is changed.
- ORDER: when the sign operates as an argument, this value states whether the functor precedes *(fwd)* or follows it *(back)*. This will be left uninstantiated much of the time.
- FEATS: this is a set of syntactic features, which include the boolean-valued FIN, which specifies whether a verb is in finite form, as well as others (not implemented in the present fragment, but which could be used for tense agreement and the like). These are essentially "head" features, as from the way in which the combination rules work together with the individual signs (see below), they get passed from the head constituent to the result.

- LEX: which specifies whether an item is a lexical entry or not. This is important for clitic placement, as clitics must be immediately adjacent to the verb to which they attach. In general, this feature is uninstantiated in lexical signs, and instantiated to the value $-$ for the result of the combination of two signs. Clitics specify that they combine with a (verb) sign whose LEX feature is set to $+$, and so they may only combine with a lexical sign. All this is explained in detail in Subsection 3.4.3. HPSG also uses such a feature for the linear ordering of complements.

- VFINAL: this feature (whose values are either $+$ or $-$) is used for ruling out ungrammatical verb-final constructions, as described in Chapter 2. It will be left out of the introductory examples at the beginning of this Chapter, until issues of linear ordering are discussed in Section 3.4.

- CASES: this represents the set of cases that the verb may assign to the possible NPs present. It is in fact a set of syntactic agreement features AGR (case, person, number, gender), together with a semantics index INDEX. It is used to link explicitly these NPs with the correct semantic index, as well as for preventing the presence of two NPs with the same case, such as two subjects (the need arises because NPs are treated as sentence modifiers).

- SEM: this has the form (Index : Propositions), where, as in UCG, each expression has a DRT-like index associated with it, which will be used to refer to the semantics of the whole expression, and a set of propositions which will usually involve this index. This may be seen as shorthand for the more usual notation where a formula has the form *[Index] Pred(Args)* (complex formulae being just instances of this pattern, where *Pred* is a logical connective and *Args* are formulae), and is a variant of the PATR-II notation of [Shieber 86], in which an index is associated to any discourse referent. Thus the semantic representation for *John walks* will be something like: $(E{:}\left\{walk(E),role(E,agent,X3),name(X3,john)\right\})$. This may be glossed as: there is an event E, which is a walking event, and in which the agent role is filled by a third person *X3* (a typed variable standing for third person entities), where the name of *X3* is *john*. In UCG, indices are typed according to a subsumption lattice. This is a partial ordering of types, where the unification of two types is their greatest lower bound. These types are built from a lattice of descriptions, such as the one shown in Figure 3.1. The nodes stand for properties that a type may have, and are tied together by two sorts of connectives. In the description of Figure 3.1, for instance, any entity may be at most one of *human* and *non-human*, and every *non-temporal* entity must be one of these two. In addition to that, any entity may be at most one of *count* and *mass*, and every *non-temporal* entity must be one of these two. A full account of such connectives and description lattices is given in [Mellish 88], together with an algorithm for generating Prolog terms to represent the types described by such a lattice[2]. In order to make the notation more transparent, certain letters are reserved for variables of certain types. Thus, as above *X3, Y3,* and *Z3* will be variable names ranging over third person entities, while *F3* and *M3* will range respectively over female and male third person entities, and *E* and *S* will range over events and states. Normally it will be clear from the context what these variables are meant to range over.

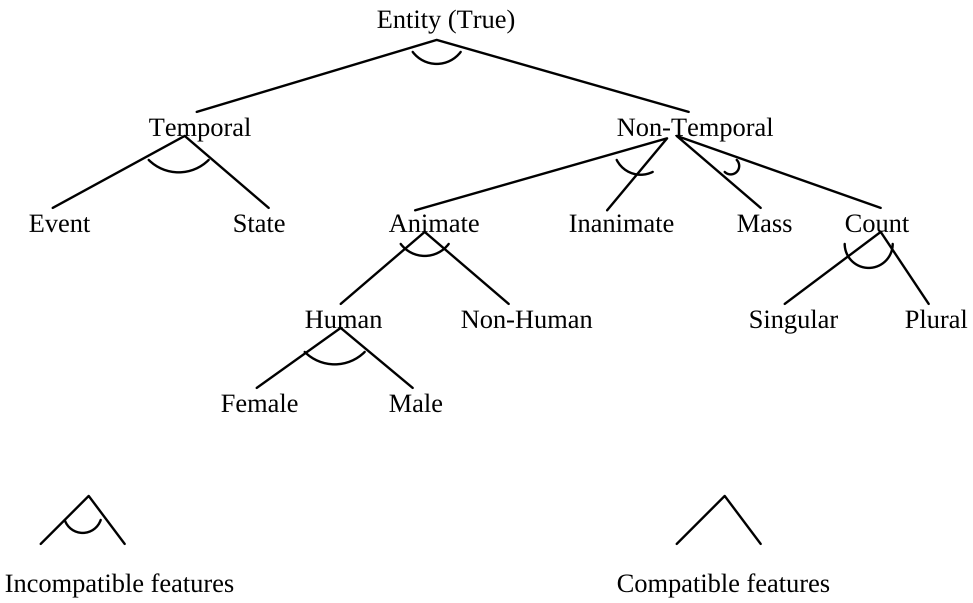


Figure 3.1: Subsumption network used

Case assignment is implemented by means of the CASES feature which links the syntactic agreement feature to the indices in the semantics. Role assignment is carried out using

[2] I am grateful to Chris Mellish for allowing me to use his code.

neo-Davidsonian semantics ([Dowty 89]), and is best illustrated by an example. The key notion is that the semantic representation of a verb consists of an index and a set of propositions involving that index. Amongst these are some that specify thematic roles, though the fillers of these thematic roles are left as variables to be instantiated in future unifications (that is, when the verb combines with its "arguments"), if at all.

There is an important restriction on the assignment of thematic roles that corresponds to the intuitive notion that an event may not have, for instance, two different agents.

(3.2) **The Principle of Thematic Uniqueness (cf. [Dowty 89])**

$$\forall E, R, I_1, I_2, (\mathrm{role}(E, R, I_1) \wedge \mathrm{role}(E, R, I_2)) \Rightarrow I_1 = I_2$$

(The semantics of an expression with index E may not assign the same thematic role R to two different expressions with indices I1 and I2.)

This Principle of Thematic Uniqueness may be realised by interpreting it as a set unification operation which shortens the representations of sets by unifying as many elements as possible. Thus for instance, the unification of $\{\mathrm{a(x),b(y),c(z)}\}\cup\mathrm{T}$ and $\{\mathrm{b(Y),C(z)}\}\cup\mathrm{S}$ yields Y=y, C=c, S=$\{\mathrm{a(x)}\}\cup\mathrm{T}$. The order in which elements appear in a set is irrelevant. More realistic examples will be given with the signs below. The broader problems of set values for unification-based grammars are discussed in [Rounds 89], but the entries presented here will be much simpler.

It should be noted that set unification as explained here does not always give a unique answer. For instance, the unification of $\{\mathrm{A,B}\}$ and $\{\mathrm{x}\}\cup\mathrm{S}$ may yield A=x or B=x. This is perfectly desirable: in Spanish, with semi-free word order and the absence of morphological case marking, it may not be possible to tell whether a given NP is the subject or the object of the verb, and hence which case it takes. A similar situation arises if the verb does not specify a unique correspondence between the NPs and thematic roles (compare *The trucks are loaded in the factory* with *The trucks are loaded in the ship*).

The basic rule for combining two signs in UCG is the standard Categorial Grammar one of FUNCTION APPLICATION ([Steedman 85]). This combines a "functor" sign with an "argument" sign unifying with the first item of the functor's SLASH list. The result is a sign that is just like the functor, but with its argument removed from the SLASH feature. In our current notation, it can be written as shown in (3.3) and (3.4). In the following examples, a list will be represented between double vertical bars, as in $\left\| \mathrm{a,\ b} \mid \mathrm{C} \right\|$, which stands for the list having *a* and *b* as its first two terms, and *C* as its tail.

Function Application (Forward)

(3.3)
$$\begin{bmatrix} \textsc{ortho} & \mathrm{O1} \\ \textsc{cat} & \mathrm{R1} \\ \textsc{slash} & \left\| \begin{bmatrix} \textsc{ortho} & \mathrm{O2} \\ \textsc{cat} & \mathrm{R2} \\ \textsc{slash} & \mathrm{B} \\ \textsc{order} & \mathrm{fwd} \\ \textsc{feats} & \mathrm{F2} \\ \textsc{cases} & \mathrm{C2} \\ \textsc{sem} & \mathrm{S2} \end{bmatrix} \mid \mathrm{A} \right\| \\ \textsc{order} & \mathrm{O1} \\ \textsc{feats} & \mathrm{F1} \\ \textsc{cases} & \mathrm{C1} \\ \textsc{sem} & \mathrm{S1} \end{bmatrix} \begin{bmatrix} \textsc{ortho} & \mathrm{O2} \\ \textsc{cat} & \mathrm{R2} \\ \textsc{slash} & \mathrm{B} \\ \textsc{order} & \mathrm{fwd} \\ \textsc{feats} & \mathrm{F2} \\ \textsc{vfinal} & \mathrm{V2} \\ \textsc{cases} & \mathrm{C2} \\ \textsc{sem} & \mathrm{S2} \end{bmatrix}$$
$$\longrightarrow \begin{bmatrix} \textsc{ortho} & \mathrm{O1\ O2} \\ \textsc{cat} & \mathrm{R1} \\ \textsc{slash} & \mathrm{A} \\ \textsc{order} & \mathrm{O1} \\ \textsc{feats} & \mathrm{F1} \\ \textsc{lex} & - \\ \textsc{vfinal} & \mathrm{V2} \\ \textsc{cases} & \mathrm{C1} \\ \textsc{sem} & \mathrm{S1} \end{bmatrix}$$

Backward application is almost identical, except that the functor's argument has its ORDER feature specified as *back*:

Function Application (Backward)

$$\begin{bmatrix} \textsc{ortho} & \mathrm{O2} \\ \textsc{cat} & \mathrm{R2} \\ \textsc{slash} & \mathrm{B} \\ \textsc{order} & \mathrm{back} \\ \textsc{feats} & \mathrm{F2} \\ \textsc{cases} & \mathrm{C2} \\ \textsc{sem} & \mathrm{S2} \end{bmatrix} \begin{bmatrix} \textsc{ortho} & \mathrm{O1} \\ \textsc{cat} & \mathrm{R1} \\ \textsc{slash} & \left\| \begin{bmatrix} \textsc{ortho} & \mathrm{O2} \\ \textsc{cat} & \mathrm{R2} \\ \textsc{slash} & \mathrm{B} \\ \textsc{order} & \mathrm{back} \\ \textsc{feats} & \mathrm{F2} \\ \textsc{cases} & \mathrm{C2} \\ \textsc{sem} & \mathrm{S2} \end{bmatrix} \mid \mathrm{A} \right\| \\ \textsc{order} & \mathrm{O1} \\ \textsc{feats} & \mathrm{F1} \\ \textsc{vfinal} & \mathrm{V1} \\ \textsc{cases} & \mathrm{C1} \\ \textsc{sem} & \mathrm{S1} \end{bmatrix} \longrightarrow \begin{bmatrix} \textsc{ortho} & \mathrm{O2\ O1} \\ \textsc{cat} & \mathrm{R1} \\ \textsc{slash} & \mathrm{A} \\ \textsc{order} & \mathrm{O1} \\ \textsc{feats} & \mathrm{F1} \\ \textsc{lex} & - \\ \textsc{vfinal} & \mathrm{V1} \\ \textsc{cases} & \mathrm{C1} \\ \textsc{sem} & \mathrm{S1} \end{bmatrix} \tag{3.4}$$

The above rules be paraphrased by saying that a "functor" sign X, whose SLASH list is $\|\mathrm{Y1} \mid \mathrm{A}\|$, may combine with a functor sign Y2 unifying with Y1, to give a resulting sign which is broadly like X, but with SLASH list A, and all the other modifications brought about by the unification of Y1 with Y2. The only exception to this rule that the result of a function application inherits its features from the functor sign X is the VFINAL feature, which is inherited from the right daughter (the argument under forward application, but the functor under backward application).

In order to make the notation clearer, let us start with a very simple example: the combination of a subject NP with an intransitive verb. Subject NPs, being optional, are sentence modifiers, as shown in (3.5), observing the usual Prolog convention that variables start with an upper case letter or an underline (_), and constants start with a lower case letter or are enclosed within single quotes:

$$(3.5)\quad \left[\begin{array}{ll}
\textsc{ortho} & \text{'María'} \\
\textsc{cat} & \text{s} \\
\textsc{slash} & \left\|\left[\begin{array}{ll}
\textsc{cat} & \text{s} \\
\textsc{slash} & \text{_C1} \\
\textsc{order} & \text{fwd} \\
\textsc{feat} & \text{Feats1} \\
\textsc{vfinal} & \text{-} \\
\textsc{cases} & \left\{\left[\begin{array}{ll}
\textsc{agr} & \left[\begin{array}{ll}
\textsc{pers} & 3 \\
\textsc{num} & \text{sg} \\
\textsc{gen} & \text{fem} \\
\textsc{case} & \text{_Case1}
\end{array}\right] \\
\textsc{index} & \text{F3}
\end{array}\right]\right\} \cup \text{Cases1} \\
\textsc{sem} & \text{I1 : Sem1}
\end{array}\right]\right\| \\
\textsc{cases} & \text{Cases1} \\
\textsc{feat} & \text{Feats1} \\
\textsc{sem} & \text{I1 : }(\{\text{role(I1,_R1,F3), name(F3,maria)}\} \cup \text{Sem1})
\end{array}\right]$$

It should be noted again that the CASES and part of the SEM features are (possibly underspecified) sets, as suggested by the notation, and $\cup$ stands for set union (as in the semantics these are sets of propositions, this gets interpreted as conjunction of descriptions). The expression $\{\text{a,b}\}\cup\text{C}$ denotes a set with elements a and b, and possibly others, represented by the variable C.

Syntactically, sign (3.5) is a sentence modifier: the sign is looking for something which unifies with the only element of SLASH (a sentence with semantics *I1:Sem1*) to yield a sentence where the semantics have been further instantiated with the propositions stating that there is a formula with index *F3*, whose name is *María*, and which fills a given role *_R1*. That first proposition is likely to be unified with one of the form *role(I,agent,X)* taken from the semantics of the verb in the way the principle of thematic uniqueness is implemented.

The case assigned to an NP is represented by the AGR:CASE value of the set difference between SLASH:CASES and CASES. In example (3.5), this set difference is:

$$\left[\begin{array}{ll}
\textsc{agr} & \left[\begin{array}{ll}
\textsc{pers} & 3 \\
\textsc{num} & \text{sg} \\
\textsc{gen} & \text{fem} \\
\textsc{case} & \text{_Case1}
\end{array}\right] \\
\textsc{index} & \text{F3}
\end{array}\right]$$

The value of the case is therefore the variable *_Case1*.

An intransitive verb in finite form, such as the following *(to sing)* is a sentence:

$$(3.6)\quad \begin{bmatrix} \textsc{ortho} & \text{canta} \\ \textsc{cat} & \text{s} \\ \textsc{slash} & \| \, \| \\ \textsc{feats} & \begin{bmatrix} \textsc{fin} & + \end{bmatrix} \\ \textsc{cases} & \left\{ \begin{bmatrix} \textsc{agr} & \begin{bmatrix} \textsc{pers} & 3 \\ \textsc{num} & \text{sg} \\ \textsc{case} & \text{nom} \end{bmatrix} \\ \textsc{index} & \text{X3} \end{bmatrix} \right\} \\ \textsc{sem} & \text{E} : \{\text{cantar(E),role(E,agt,X3)}\} \end{bmatrix}$$

The CASES feature of this sign indicates that the verb may assign a nominative case. However, the fact that the SLASH feature is empty means that it does not require any arguments: we have a complete sentence here, which may be modified by a nominative NP such as (3.5) by means of the following unifications:

(3.7)

$$\begin{aligned} &\_\text{C1}=\|\,\| \\ &\_\text{Case1}=\text{nom} \\ &\text{F3}=\text{X3} \\ &\text{Cases1}=\{\} \\ &\text{I1}=\text{E} \\ &\text{Sem1}=\{\text{cantar(E),role(E,agt,F3)}\} \\ &\_\text{R1} = \text{agt (by the principle of thematic uniqueness)} \\ &\text{Feats1}=\begin{bmatrix} \textsc{fin} & + \end{bmatrix} \end{aligned}$$

The resulting sign is:

$$(3.8)\quad \begin{bmatrix} \textsc{ortho} & \text{'María canta'} \\ \textsc{cat} & \text{s} \\ \textsc{slash} & \| \, \| \\ \textsc{cases} & \{\} \\ \textsc{feats} & \begin{bmatrix} \textsc{fin} & + \end{bmatrix} \\ \textsc{lex} & - \\ \textsc{sem} & \text{E} : \{\text{cantar(E), role(E,agt,F3), name(F3,maria)}\} \end{bmatrix}$$

Let us now turn towards verbs that do require (object) arguments, starting with a simple transitive verb.

As was said above, all subject NPs are sentence modifiers. As there is no way of distinguishing them from other NPs, all NPs look like (3.5). The following means *the book* (we shall leave out, in order to simplify the exposition, how this is derived).

$$
(3.9)\quad \left[\begin{array}{ll}
\text{ORTHO} & \text{'el libro'} \\
\text{CAT} & \text{s} \\
\text{SLASH} & \left\|\left[\begin{array}{ll}
\text{CAT} & \text{s} \\
\text{SLASH} & \text{_C2} \\
\text{ORDER} & \text{fwd} \\
\text{FEATS} & \text{Feats2} \\
\text{VFINAL} & \text{-} \\
\text{CASES} & \left\{\left[\begin{array}{ll}
\text{AGR} & \left[\begin{array}{ll}
\text{PERS} & 3 \\
\text{NUM} & \text{sg} \\
\text{GENR} & \text{masc} \\
\text{CASE} & \text{_Case2}
\end{array}\right] \\
\text{INDEX} & \text{B}
\end{array}\right]\right\} \cup \text{Cases2} \\
\text{SEM} & \text{I2 : Sem2}
\end{array}\right]\right\| \\
\text{FEATS} & \text{Feats2} \\
\text{CASES} & \text{Cases2} \\
\text{SEM} & \text{I2 : } (\{\text{role(I2,_R2,B),libro(B),definite(B)}\} \cup \text{Sem2})
\end{array}\right]
$$

(The semantics is only approximate for the time being: reference to the definiteness of the NP is rather crude).

A transitive verb must then subcategorize for such a NP as (3.9), so it must be, in traditional categorial grammar notation, a S/(S/S), i.e. a sentence looking for a sentence modifier. The following one means *read* (in the past tense, third person singular). Following the standard PATR notation, tags such as $\boxed{1}$ will be used to indicate reentrancy in this example and others below.

$$(3.10)\quad \left[\begin{array}{ll}
\textsc{ortho} & \text{leyó} \\
\textsc{cat} & \text{s} \\
\textsc{slash} & \left\| \left[\begin{array}{ll}
\textsc{cat} & \text{s} \\
\textsc{slash} & \left\| \left[\begin{array}{ll}
\textsc{cat} & \text{s} \\
\textsc{slash} & \| \, \| \\
\textsc{feats} & \boxed{1} \\
\textsc{cases} & \left\{ \left[\begin{array}{ll} \textsc{agr} & \left[\begin{array}{ll} \textsc{case} & \text{acc} \end{array}\right] \\ \textsc{index} & \text{Y} \end{array}\right] \right\} \\
\textsc{sem} & \text{E} : \left\{ \begin{array}{l} \text{leer(E)}, \\ \text{role(E,agt,X3)}, \\ \text{role(E,pat,Y)} \end{array} \right\}
\end{array}\right] \right\| \\
\textsc{order} & \text{fwd} \\
\textsc{cases} & \text{Cases} \\
\textsc{sem} & \text{Sem3}
\end{array}\right] \right\| \\
\textsc{feats} & \boxed{1} \left[\begin{array}{ll} \textsc{fin} & + \end{array}\right] \\
\textsc{cases} & \left\{ \left[\begin{array}{ll} \textsc{agr} & \left[\begin{array}{ll} \textsc{pers} & 3 \\ \textsc{num} & \text{sg} \\ \textsc{case} & \text{nom} \end{array}\right] \\ \textsc{index} & \text{X3} \end{array}\right] \right\} \cup \text{Cases} \\
\textsc{sem} & \text{Sem3}
\end{array}\right]$$

This sign tells us that we have a sentence looking for a sentence modifier (represented by the SLASH). This sentence modifier (an NP following what has been said above) will receive the *accusative* case, represented by SLASH:SLASH:CASES: (this mechanism will, incidentally, also prevent any adverbs from filling that slot.) The NP will also enrich the semantics of the resulting sentence by instantiating the index that fills the *patient* role. The full steps involved in the combination are given below. A second lexical entry realises the intransitive version of this verb.

Combination of the verb *leyó* (3.10) with its object *el libro* (3.9) is by means of the unification of (3.9) with the SLASH feature of (3.10), giving the following unifications:

(3.11)
$$\begin{aligned}
&_C2 = \|\ \| \\
&_Case2 = \text{acc} \\
&B = Y \\
&Cases2 = \{\} \\
&I2 = E \\
&_R2 = \text{pat} \\
&Sem2 = \{\text{leer(E),role(E,agt,X3),role(E,pat,B)}\} \\
&Feats2 = \begin{bmatrix}\textsc{fin} & +\end{bmatrix} \\
&Cases = \{\} \\
&Sem3 = E : \left\{\begin{array}{l}\text{role(E,pat,B), libro(B), definite(B),} \\ \text{leer(E), role(E,agt,X3)}\end{array}\right\}
\end{aligned}$$

The resulting sign will then be:

(3.12)
$$\begin{bmatrix}
\textsc{ortho} & \text{'leyó el libro'} \\
\textsc{cat} & \textsc{s} \\
\textsc{slash} & \|\ \| \\
\textsc{feats} & \begin{bmatrix}\textsc{fin} & +\end{bmatrix} \\
\textsc{lex} & - \\
\textsc{cases} & \left\{\begin{bmatrix}\textsc{agr} & \begin{bmatrix}\textsc{pers} & 3 \\ \textsc{num} & \text{sg} \\ \textsc{case} & \text{nom}\end{bmatrix} \\ \textsc{index} & X3\end{bmatrix}\right\} \\
\textsc{sem} & E : \{\text{role(E,pat,B), libro(B), definite(B), leer(E), role(E,agt,X3)}\}
\end{bmatrix}$$

This may be combined with the subject NP (3.5) which acts as a syntactic functor over the S (3.12), causing (3.12) to unify with the SLASH feature of (3.5). This gives the same unifications (3.7) as between the subject NP (3.5) and the intransitive verb (3.6) above.

(3.13)

$$
\begin{array}{l}
\_C1=\langle\!\langle\rangle\!\rangle \\
\_Case1=nom \\
X3=F3 \\
Cases1=\{\} \\
I1=E \\
Sem1=\{role(E,pat,B),libro(B),definite(B),leer(E),role(E,agt,F3)\} \\
\_R1=agt \\
Feats1=\left[\textsc{fin}\ \ +\right]
\end{array}
$$

The resulting sentence is then:

$$
(3.14)\left[\begin{array}{ll}
\textsc{ortho} & \text{'María leyó el libro'} \\
\textsc{cat} & \text{s} \\
\textsc{slash} & \langle\!\langle\rangle\!\rangle \\
\textsc{cases} & \{\} \\
\textsc{feats} & \left[\textsc{fin}\ \ +\right] \\
\textsc{lex} & - \\
\textsc{sem} & \text{E} : \left\{\begin{array}{l}\text{role(E,pat,B), libro(B), definite(B),} \\ \text{leer(E), role(E,agt,F3), name(F3,maria)}\end{array}\right\}
\end{array}\right]
$$

3.4 Extending the grammar

This section will present a basic grammar for Spanish that will cover a small fragment of the language, including phenomena discussed in Chapter 2.

3.4.1 Freedom of word order: lexical redundancy rules

As was seen in the previous chapter, the transitive verb in final position is ruled out when the NPs are lexical (as opposed to clitics), but the ordering OVS is allowed (albeit with some effects on the semantics, which will be ignored here, but could be built-in since the semantic representation is made explicit in the “input” to the rule). Clearly, the transitive verb entry in the previous section (3.10) requires its object to follow it and will prevent the ordering OVS. The way such ordering is allowed is with a new lexical entry for the verb, built from the “canonical” one (3.10) by means of a **lexical redundancy**

rule (or **lexical rule** for short). Lexical rules ([Pollard and Sag 87]) can be seen as a way of reducing the complexity of the lexicon by "factoring out" the information common to several entries. For instance, they allow passive verb forms to be built from their corresponding active ones. They are a way of making the lexicon more compact and that distinguishes them from *unary rules*, which are more properly part of the grammar, and are not used here. The consequences of lexical redundancy rules for the complexity of the translation process will be addressed in Section 6.4.

Lexical rules will be of the form *Sign1* $\longrightarrow$ *Sign2*, and the meaning of this will be that whenever we have a lexical sign unifying with *Sign1*, we may build another lexical sign in the form of *Sign2*. A lexical rule can be devised to build a new entry for a transitive verb, such that it requires its object to precede it. The resulting sign will have its VFINAL feature set to +, to make sure that no ungrammatical combinations may result from its subject preceding it.

VO – OV

$$(3.15)\quad \begin{bmatrix} \textsc{ortho} & \mathrm{O1} \\ \textsc{cat} & \mathrm{s} \\ \textsc{slash} & \left\| \begin{bmatrix} \textsc{cat} & \mathrm{s} \\ \textsc{slash} & \left\| \begin{bmatrix} \textsc{cat} & \mathrm{s} \\ \textsc{slash} & \|\,\| \\ \textsc{feats} & \boxed{1} \\ \textsc{cases} & \mathrm{C1} \\ \textsc{sem} & \mathrm{Sem1} \end{bmatrix} \right\| \\ \textsc{order} & \mathrm{fwd} \\ \textsc{cases} & \mathrm{C2} \\ \textsc{sem} & \mathrm{Sem2} \end{bmatrix} \right\| \\ \textsc{feats} & \boxed{1}\begin{bmatrix} \textsc{fin} & + \end{bmatrix} \\ \textsc{cases} & \mathrm{C3} \\ \textsc{sem} & \mathrm{Sem3} \end{bmatrix}$$

$$\longrightarrow \quad \begin{bmatrix} \textsc{ortho} & \mathrm{O1} \\ \textsc{cat} & \mathrm{s} \\ \textsc{slash} & \left\| \begin{bmatrix} \textsc{cat} & \mathrm{s} \\ \textsc{slash} & \left\| \begin{bmatrix} \textsc{cat} & \mathrm{s} \\ \textsc{slash} & \|\,\| \\ \textsc{feats} & \boxed{1} \\ \textsc{cases} & \mathrm{C1} \\ \textsc{sem} & \mathrm{Sem1} \end{bmatrix} \right\| \\ \textsc{order} & \mathrm{back} \\ \textsc{cases} & \mathrm{C2} \\ \textsc{sem} & \mathrm{Sem2} \end{bmatrix} \right\| \\ \textsc{feats} & \boxed{1}\begin{bmatrix} \textsc{fin} & + \end{bmatrix} \\ \textsc{vfinal} & + \\ \textsc{cases} & \mathrm{C3} \\ \textsc{sem} & \mathrm{Sem3} \end{bmatrix}$$

If the phonology and the semantics were covered more completely, the output of this lexical rule would account for the effects of having the object before the verb. As it stands, the rule produces, from the transitive verb (3.10), the following one:

$$
(3.16)\quad
\begin{bmatrix}
\textsc{ortho} & \text{leyó} \\
\textsc{cat} & \text{s} \\
\textsc{slash} & \left\Vert \begin{bmatrix}
\textsc{cat} & \text{s} \\
\textsc{slash} & \left\Vert \begin{bmatrix}
\textsc{cat} & \text{s} \\
\textsc{slash} & \Vert\,\Vert \\
\textsc{feats} & \boxed{1} \\
\textsc{cases} & \left\{ \begin{bmatrix} \textsc{agr} & \begin{bmatrix} \textsc{case} & \text{acc} \end{bmatrix} \\ \textsc{index} & \text{Y} \end{bmatrix} \right\} \\
\textsc{sem} & \text{E} : \left\{ \begin{array}{l} \text{leer(E)}, \\ \text{role(E,agt,X3)}, \\ \text{role(E,pat,Y)} \end{array} \right\}
\end{bmatrix} \right\Vert \\
\textsc{order} & \text{back} \\
\textsc{cases} & \text{Cases} \\
\textsc{sem} & \text{Sem3}
\end{bmatrix} \right\Vert \\
\textsc{feats} & \boxed{1} \begin{bmatrix} \textsc{fin} & + \end{bmatrix} \\
\textsc{vfinal} & + \\
\textsc{cases} & \left\{ \begin{bmatrix} \textsc{agr} & \begin{bmatrix} \textsc{pers} & 3 \\ \textsc{num} & \text{sg} \\ \textsc{case} & \text{nom} \end{bmatrix} \\ \textsc{index} & \text{X3} \end{bmatrix} \right\} \cup \text{Cases} \\
\textsc{sem} & \text{Sem3}
\end{bmatrix}
$$

It is almost identical to (3.10), except that it searches for its object to its left, and it has the VFINAL feature set to +, to prevent any ungrammatical verb-final constructions. NPs such as (3.5) and (3.9) require the S for which they subcategorize not to be verb-final. These NPs look for their S to their right, and cannot therefore combine with an OV construction. The required NPs are built from the existing ones by a lexical rule which produces an NP searching for the S to its left, which allows the OVS ordering. The rule is:

Subj – Inv

$$
\begin{bmatrix}
\textsc{ortho} & \mathrm{O2} \\
\textsc{cat} & \mathrm{s} \\
\textsc{slash} & \left\| \begin{bmatrix}
\textsc{ortho} & \mathrm{O1} \\
\textsc{cat} & \mathrm{s} \\
\textsc{slash} & \mathrm{S1} \\
\textsc{order} & \mathrm{fwd} \\
\textsc{feat} & \mathrm{F1} \\
\textsc{vfinal} & - \\
\textsc{cases} & \mathrm{C1} \\
\textsc{sem} & \mathrm{Sem1}
\end{bmatrix} \right\| \\
\textsc{cases} & \mathrm{C2} \\
\textsc{feat} & \mathrm{F2} \\
\textsc{sem} & \mathrm{S2}
\end{bmatrix}
\tag{3.17}
$$

$$
\longrightarrow
\begin{bmatrix}
\textsc{ortho} & \mathrm{O2} \\
\textsc{cat} & \mathrm{s} \\
\textsc{slash} & \left\| \begin{bmatrix}
\textsc{ortho} & \mathrm{O1} \\
\textsc{cat} & \mathrm{s} \\
\textsc{slash} & \mathrm{S1} \\
\textsc{order} & \mathrm{back} \\
\textsc{feat} & \mathrm{F1} \\
\textsc{cases} & \mathrm{C1} \\
\textsc{sem} & \mathrm{Sem1}
\end{bmatrix} \right\| \\
\textsc{cases} & \mathrm{C2} \\
\textsc{feat} & \mathrm{F2} \\
\textsc{sem} & \mathrm{S2}
\end{bmatrix}
$$

This rule produces, from the standard NP (3.5), the alternative version, which looks for the rest of the sentence to its left, and no longer requires it not to be verb-final.

$$
(3.18)\left[\begin{array}{ll}
\textsc{ortho} & \text{'María'} \\
\textsc{cat} & \text{s} \\
\textsc{slash} & \left\|\left[\begin{array}{ll}
\textsc{cat} & \text{s} \\
\textsc{slash} & \text{\_C1} \\
\textsc{order} & \text{back} \\
\textsc{feat} & \text{Feats1} \\
\textsc{cases} & \left\{\left[\begin{array}{ll}
\textsc{agr} & \left[\begin{array}{ll}
\textsc{pers} & 3 \\
\textsc{num} & \text{sg} \\
\textsc{gen} & \text{fem} \\
\textsc{case} & \text{\_Case1}
\end{array}\right] \\
\textsc{index} & \text{F3}
\end{array}\right]\right\} \cup \text{Cases1} \\
\textsc{sem} & \text{I1 : Sem1}
\end{array}\right]\right\| \\
\textsc{cases} & \text{Cases1} \\
\textsc{feat} & \text{Feats1} \\
\textsc{sem} & \text{I1} : (\{\text{role(I1,\_R1,F3), name(F3,maria)}\} \cup \text{Sem1})
\end{array}\right]
$$

From the above verb and the NP (3.9), the following "VP" can be constructed:

$$
(3.19)\left[\begin{array}{ll}
\textsc{ortho} & \text{'el libro leyó'} \\
\textsc{cat} & \text{s} \\
\textsc{slash} & \|\,\| \\
\textsc{feats} & \left[\begin{array}{ll}\textsc{fin} & +\end{array}\right] \\
\textsc{lex} & - \\
\textsc{vfinal} & + \\
\textsc{cases} & \left\{\left[\begin{array}{ll}
\textsc{agr} & \left[\begin{array}{ll}
\textsc{pers} & 3 \\
\textsc{num} & \text{sg} \\
\textsc{case} & \text{nom}
\end{array}\right] \\
\textsc{index} & \text{X3}
\end{array}\right]\right\} \\
\textsc{sem} & \text{E} : \{\text{role(E,pat,B), libro(B), definite(B), leer(E), role(E,agt,X3)}\}
\end{array}\right]
$$

It is just like 3.12, but with its verb and complement swapped around. This may combine with (3.18) as its subject (but crucially not with (3.5) because of the clashing values of the VFINAL feature, to give the sentence:

$$(3.20)\begin{bmatrix} \textsc{ortho} & \text{'el libro leyó María'} \\ \textsc{cat} & \textsc{s} \\ \textsc{slash} & \|\| \\ \textsc{cases} & \{\} \\ \textsc{feats} & \begin{bmatrix}\textsc{fin} & + \end{bmatrix} \\ \textsc{lex} & - \\ \textsc{sem} & \text{E} : \left\{ \begin{array}{l} \text{role(E,pat,B), libro(B), definite(B),} \\ \text{leer(E), role(E,agt,F3), name(F3,maria)} \end{array} \right\} \end{bmatrix}$$

A similar lexical rule can be specified to change the order of the two objects of a ditransitive verb, and to "move" either (or both) to the front, while the VFINAL feature still prevents verb-final constructions.

As an example, consider the following ditransitive verb (*gave*, in the past tense, third person singular form):

$$
(3.21)\quad
\left[\begin{array}{ll}
\textsc{ortho} & \text{dio} \\
\textsc{cat} & \text{s} \\
\textsc{slash} & \left\|
\begin{array}{l}
\left[\begin{array}{ll}
\textsc{cat} & \text{s} \\
\textsc{slash} & \left\| \left[\begin{array}{ll}
\textsc{cat} & \text{s} \\
\textsc{slash} & \|\,\| \\
\textsc{feats} & \boxed{1} \\
\textsc{cases} & \left\{ \left[\begin{array}{ll} \textsc{agr} & \left[\textsc{case}\ \ \text{acc}\right] \\ \textsc{index} & \text{Y} \end{array}\right] \right\} \\
\textsc{sem} & \text{E} : \left\{\begin{array}{l} \text{dar(E)}, \\ \text{role(E,agt,X3)}, \\ \text{role(E,pat,Y)}, \\ \text{role(E,goal,Z)} \end{array}\right\}
\end{array}\right] \right\| \\
\textsc{order} & \text{fwd} \\
\textsc{cases} & \text{Cases1} \\
\textsc{sem} & \text{Sem1}
\end{array}\right] \\
\left[\begin{array}{ll}
\textsc{cat} & \text{s} \\
\textsc{slash} & \left\| \left[\begin{array}{ll}
\textsc{cat} & \text{s} \\
\textsc{slash} & \|\,\| \\
\textsc{feats} & \boxed{1} \\
\textsc{cases} & \left\{ \left[\begin{array}{ll} \textsc{agr} & \left[\textsc{case}\ \ \text{dat}\right] \\ \textsc{index} & \text{Z} \end{array}\right] \right\} \\
\textsc{sem} & \text{Sem1}
\end{array}\right] \right\| \\
\textsc{order} & \text{fwd} \\
\textsc{cases} & \text{Cases2} \\
\textsc{sem} & \text{Sem2}
\end{array}\right]
\end{array}
\right\| \\
\textsc{feats} & \boxed{1} \left[\textsc{fin}\ \ +\right] \\
\textsc{cases} & \left\{ \left[\begin{array}{ll} \textsc{agr} & \left[\begin{array}{ll} \textsc{pers} & 3 \\ \textsc{num} & \text{sg} \\ \textsc{case} & \text{nom} \end{array}\right] \\ \textsc{index} & \text{X3} \end{array}\right] \right\} \cup \text{Cases1} \cup \text{Cases2} \\
\textsc{sem} & \text{Sem2}
\end{array}\right]
$$

In a similar manner to the simple transitive verb (3.10) of the previous section, the case mechanism ensures that the accusative NP gets assigned the role of patient, the dative one the role of goal, and the nominative one the role of agent.

Let us concentrate for the time being on the essential part of the combination mechanism: the SLASH and SEM features. The order of the objects in the SLASH list requires the direct object to combine before the indirect object. In other words, this entry specifies the relative ordering VOI. As the verb combines with its objects, the semantics become more and more instantiated.

Let us start by combining it with a direct object NP, such as (3.9), reprinted here for clarity:

$$(3.22)\begin{bmatrix} \textsc{ortho} & \text{'el libro'} \\ \textsc{cat} & \text{s} \\ \textsc{slash} & \left\| \begin{bmatrix} \textsc{cat} & \text{s} \\ \textsc{slash} & \text{\_C2} \\ \textsc{order} & \text{fwd} \\ \textsc{feats} & \text{Feats2} \\ \textsc{vfinal} & \text{-} \\ \textsc{cases} & \left\{ \begin{bmatrix} \textsc{agr} & \begin{bmatrix} \textsc{pers} & 3 \\ \textsc{num} & \text{sg} \\ \textsc{gen} & \text{masc} \\ \textsc{case} & \text{\_Case2} \end{bmatrix} \\ \textsc{index} & \text{B} \end{bmatrix} \right\} \cup \text{Cases2} \\ \textsc{sem} & \text{I2 : Sem2} \end{bmatrix} \right\| \\ \textsc{feats} & \text{Feats2} \\ \textsc{cases} & \text{Cases2} \\ \textsc{sem} & \text{I2 : } (\{\text{role(I2,\_R2,B),libro(B),definite(B)}\} \cup \text{Sem2}) \end{bmatrix}$$

The result is:

$$
\begin{bmatrix}
\text{ORTHO} & \text{'dio el libro'} \\
\text{CAT} & \text{s} \\
\text{SLASH} & \left\| \begin{bmatrix}
\text{CAT} & \text{s} \\
\text{SLASH} & \left\| \begin{bmatrix}
\text{CAT} & \text{s} \\
\text{SLASH} & \|\,\| \\
\text{FEATS} & \boxed{1} \\
\text{CASES} & \left\{ \begin{bmatrix} \text{AGR} & \begin{bmatrix} \text{CASE} & \text{dat} \end{bmatrix} \\ \text{INDEX} & \text{Z} \end{bmatrix} \right\} \\
\text{SEM} & \text{E} : \left\{ \begin{array}{l} \text{dar(E),role(E,agt,X3),} \\ \text{role(E,pat,B),libro(B),} \\ \text{definite(B),role(E,goal,Z)} \end{array} \right\}
\end{bmatrix} \right\| \\
\text{ORDER} & \text{fwd} \\
\text{CASES} & \text{Cases2} \\
\text{SEM} & \text{Sem2}
\end{bmatrix} \right\| \\
\text{FEATS} & \boxed{1} \begin{bmatrix} \text{FIN} & + \end{bmatrix} \\
\text{LEX} & - \\
\text{CASES} & \left\{ \begin{bmatrix} \text{AGR} & \begin{bmatrix} \text{PERS} & 3 \\ \text{NUM} & \text{sg} \\ \text{CASE} & \text{nom} \end{bmatrix} \\ \text{INDEX} & \text{X3} \end{bmatrix} \right\} \cup \text{Cases2} \\
\text{SEM} & \text{Sem2}
\end{bmatrix}
\tag{3.23}
$$

This can then combine with a dative NP such as:

$$
\begin{bmatrix}
\text{ORTHO} & \text{'a María'} \\
\text{CAT} & \text{s} \\
\text{SLASH} & \left\| \begin{bmatrix}
\text{CAT} & \text{s} \\
\text{SLASH} & \text{_C6} \\
\text{ORDER} & \text{fwd} \\
\text{FEATS} & \text{Feats6} \\
\text{VFINAL} & \text{-} \\
\text{CASES} & \left\{ \begin{bmatrix} \text{AGR} & \begin{bmatrix} \text{PERS} & 3 \\ \text{NUM} & \text{sg} \\ \text{GEN} & \text{fem} \\ \text{CASE} & \text{dat} \end{bmatrix} \\ \text{INDEX} & \text{F3} \end{bmatrix} \right\} \cup \text{Cases6} \\
\text{SEM} & \text{I6 : Sem6}
\end{bmatrix} \right\| \\
\text{FEATS} & \text{Feats6} \\
\text{CASES} & \text{Cases6} \\
\text{SEM} & \text{I6} : (\{\text{role(I6,_R6,F3), name(F3, maria)}\} \cup \text{Sem6})
\end{bmatrix}
\tag{3.24}
$$

The result is:

$$(3.25)\begin{bmatrix} \textsc{ortho} & \text{'dio el libro a María'} \\ \textsc{cat} & \text{s} \\ \textsc{slash} & \| \, \| \\ \textsc{feats} & \begin{bmatrix} \textsc{fin} & + \end{bmatrix} \\ \textsc{lex} & - \\ \textsc{cases} & \left\{ \begin{bmatrix} \textsc{agr} & \begin{bmatrix} \textsc{pers} & 3 \\ \textsc{num} & \text{sg} \\ \textsc{case} & \text{nom} \end{bmatrix} \\ \textsc{index} & \text{X3} \end{bmatrix} \right\} \\ \textsc{sem} & \text{E} : \left\{ \begin{array}{l} \text{role(E,goal,F3),name(F3, maria),} \\ \text{role(E,pat,B),libro(B),} \\ \text{definite(B), dar(E), role(E,agt,X3)} \end{array} \right\} \end{bmatrix}$$

It is clear from the above entry for the ditransitive verb (3.21), that the two objects may only appear in the order specified. An alternative lexical entry for *dio* is built by the following lexical rule:

VOI – VIO

$$\left[\begin{array}{ll}
\textsc{ortho} & \text{O1} \\
\textsc{cat} & \text{s} \\
\textsc{slash} & \left\| \left[\begin{array}{ll}
\textsc{cat} & \text{s} \\
\textsc{slash} & \left\| \left[\begin{array}{ll}
\textsc{cat} & \text{s} \\
\textsc{slash} & \| \| \\
\textsc{feats} & \boxed{1} \\
\textsc{cases} & \text{C1} \\
\textsc{sem} & \text{Sem1}
\end{array}\right] \right\| \\
\textsc{order} & \text{fwd} \\
\textsc{cases} & \text{C2} \\
\textsc{sem} & \text{Sem2}
\end{array}\right] \left[\begin{array}{ll}
\textsc{cat} & \text{s} \\
\textsc{slash} & \left\| \left[\begin{array}{ll}
\textsc{cat} & \text{s} \\
\textsc{slash} & \| \| \\
\textsc{feats} & \boxed{2} \\
\textsc{cases} & \text{C3} \\
\textsc{sem} & \text{Sem3}
\end{array}\right] \right\| \\
\textsc{order} & \text{fwd} \\
\textsc{cases} & \text{C3} \\
\textsc{sem} & \text{Sem3}
\end{array}\right] \right\| \\
\textsc{feats} & \left[\begin{array}{ll} \textsc{fin} & + \end{array}\right] \\
\textsc{cases} & \text{C3} \\
\textsc{sem} & \text{Sem3}
\end{array}\right]$$

(3.26)$\longrightarrow$

$$\left[\begin{array}{ll}
\textsc{ortho} & \text{O1} \\
\textsc{cat} & \text{s} \\
\textsc{slash} & \left\| \left[\begin{array}{ll}
\textsc{cat} & \text{s} \\
\textsc{slash} & \left\| \left[\begin{array}{ll}
\textsc{cat} & \text{s} \\
\textsc{slash} & \| \| \\
\textsc{feats} & \boxed{2} \\
\textsc{cases} & \text{C3} \\
\textsc{sem} & \text{Sem3}
\end{array}\right] \right\| \\
\textsc{order} & \text{fwd} \\
\textsc{cases} & \text{C3} \\
\textsc{sem} & \text{Sem3}
\end{array}\right] \left[\begin{array}{ll}
\textsc{cat} & \text{s} \\
\textsc{slash} & \left\| \left[\begin{array}{ll}
\textsc{cat} & \text{s} \\
\textsc{slash} & \| \| \\
\textsc{feats} & \boxed{1} \\
\textsc{cases} & \text{C1} \\
\textsc{sem} & \text{Sem1}
\end{array}\right] \right\| \\
\textsc{order} & \text{fwd} \\
\textsc{cases} & \text{C2} \\
\textsc{sem} & \text{Sem2}
\end{array}\right] \right\| \\
\textsc{feats} & \boxed{1} \left[\begin{array}{ll} \textsc{fin} & + \end{array}\right] \\
\textsc{vfinal} & + \\
\textsc{cases} & \text{C3} \\
\textsc{sem} & \text{Sem3}
\end{array}\right]$$

This produces a version of the ditransitive verb *dio* just like (3.21), but with the elements of the SLASH list in the other order (again, we shall ignore any discourse effects that this may bring about). This can then first combine with its indirect object NP (3.24). The combination is achieved by the following unifications:

(3.27)

$$\begin{array}{l}
_C6 = \|\,\| \\
Z = F3 \\
Cases6 = \{\} \\
Sem1 = I6 : Sem6 \\
Feat6 = \left[\textsc{fin}\ \ +\right] \\
Cases2 = \{\} \\
Sem2 = I6 : (\{role(I6,_R6,F3),name(F3,maria)\} \cup Sem6)
\end{array}$$

The resulting sign is:

$$(3.28)\left[\begin{array}{ll}
\textsc{ortho} & \text{'dio a María'}\\
\textsc{cat} & s\\
\textsc{slash} & \left\|\left[\begin{array}{ll}
\textsc{cat} & s\\
\textsc{slash} & \left\|\left[\begin{array}{ll}
\textsc{cat} & s\\
\textsc{slash} & \|\,\|\\
\textsc{feats} & \boxed{1}\\
\textsc{cases} & \left\{\left[\begin{array}{ll}\textsc{agr} & \left[\textsc{case}\ \ acc\right]\\ \textsc{index} & Y\end{array}\right]\right\}\\
\textsc{sem} & E : \left\{\begin{array}{l} dar(E),role(E,agt,X3),\\ role(E,pat,Y),role(E,goal,F3)\end{array}\right\}
\end{array}\right]\right\|\\
\textsc{cases} & Cases1\\
\textsc{sem} & I6 : Sem6
\end{array}\right]\right\|\\
\textsc{feats} & \boxed{1}\left[\textsc{fin}\ \ +\right]\\
\textsc{lex} & -\\
\textsc{cases} & \left\{\left[\begin{array}{ll}\textsc{agr} & \left[\begin{array}{ll}\textsc{pers} & 3\\ \textsc{num} & sg\\ \textsc{case} & nom\end{array}\right]\\ \textsc{index} & X3\end{array}\right]\right\} \cup Cases1\\
\textsc{sem} & I6 : \{role(I6,_R6,F3),name(F3,maria)\} \cup Sem6
\end{array}\right]$$

This sign can now combine with a direct object NP such as (3.22) by the following unifications:

(3.29)
$$
\begin{aligned}
&\_C2 = \|\,\|\\
&\_Case2 = \mathrm{acc}\\
&B = Y\\
&\mathrm{Cases2} = \{\}\\
&\mathrm{I2} = E\\
&\mathrm{Sem2} = \{\mathrm{dar(E),role(E,agt,X3),role(E,pat,B),role(E,goal,F3)}\}\\
&\mathrm{Feat2} = \begin{bmatrix}\textsc{fin} & +\end{bmatrix}\\
&\mathrm{Cases1} = \{\}\\
&\mathrm{I8} = E\\
&\mathrm{Sem6} = \{\mathrm{role(E,\_R2,B),libro(B),definite(B)}\} \cup \mathrm{Sem2}
\end{aligned}
$$

The resulting sign is as (3.25):

(3.30)
$$
\begin{bmatrix}
\textsc{ortho} & \text{'dio a María el libro'}\\
\textsc{cat} & \textsc{s}\\
\textsc{slash} & \|\,\|\\
\textsc{feats} & \begin{bmatrix}\textsc{fin} & +\end{bmatrix}\\
\textsc{lex} & -\\
\textsc{cases} & \left\{\begin{bmatrix}\textsc{agr} & \begin{bmatrix}\textsc{pers} & 3\\ \textsc{num} & \mathrm{sg}\\ \textsc{case} & \mathrm{nom}\end{bmatrix}\\ \textsc{index} & X3\end{bmatrix}\right\}\\
\textsc{sem} & E : \left\{\begin{array}{l}\mathrm{role(E,goal,F3),name(F3,\ maria),}\\ \mathrm{role(E,pat,B),libro(B),}\\ \mathrm{definite(B),\ dar(E),\ role(E,agt,X3)}\end{array}\right\}
\end{bmatrix}
$$

The advantage of this lexical rule approach is that such rules would be able to alter the level of "non-canonicity" of the word order as the sentence is parsed, and this could be put into some explicit correspondence with the discourse effects that are achieved with these word orderings, if such effects were to be implemented.

For instance, according to the data in [Foster 90], the ordering V[preterite]SO is questionable when V and O are given and S is new information, but perfectly acceptable when S and V are given but O new:

(3.31) Preceding Question: *Who did the soldiers arrest?*
Arrestaron los soldados a María.
Arrested the soldiers to Mary
'The soldiers arrested Mary.'
Preceding Question: *Who arrested Mary?*
? Arrestaron los soldados a María.

Similarly, the ordering O[def]VS is ungrammatical with V and O given and S new, but acceptable with S and V given and O new:

(3.32) Preceding Question: *Who hid the rifle?*
*El fusil escondió Juan.
The rifle hid John
'John hid the rifle.'
Preceding Question: *Which one did John hide?*
El fusil escondió Juan.

If the information structure in the semantics represented such distinctions as given *vs.* new, these lexical rules could relate the ORDER features of the signs to these requirements on the semantics.

3.4.2 Function composition

The previous subsection showed how signs can be built from existing ones, by using lexical redundancy rules that reorder the SLASH list of their left hand side. The signs are put together essentially by function application.

However, some orderings are still not generated using this alone. For instance, the ditransitive verb must combine with its objects before it combines with its subject (since the subject, not being in the SLASH list, is not amenable to treatment by lexical rules that apply to the verb). This rules out grammatical examples where the subject is between the verb and the objects, such as:

(3.33) a. Dio María el libro a Juan.
gave-3sg Mary the book to John
'Mary gave the book to John'

b. Le dio María el libro a Juan.
IO-cl gave-3sg Mary the book to John

c. A Juan le dio María el libro.

For this purpose, and also for independent reasons such as getting the facts about "non-constituent" coordination right, it is necessary to have, as in [Dowty 85], [Moortgat 88] and [Steedman 85], a rule that does function composition, the standard formulation of which is as follows:

(3.34)

$$\frac{X/Y \quad Y/Z}{X/Z}\text{Forward Composition (FC)}$$

Such a rule can be summarised by saying that a sign with syntax X/Y1 combines with one with syntax Y2/Z, (where Y1 unifies with Y2) to give a result X/Z, together with the modifications that may come as a result of unifying Y1 with Y2.

Since the elements which are subcategorized for appear in our notation under the SLASH list, we have a rule which (in its forward variant) can be glossed as follows using the Steedman notation:

(3.35)

$$\frac{R1/(R2/B) \quad ((R2/B)/C \ldots)}{R1/C \ldots}\text{Forward Composition (FC)}$$

Since the SLASH feature is a list, if B above is empty, the rule is identical to Steedman's (3.34). In order to describe the rule more precisely, we need to introduce an operation to append (or concatenate) two lists, written as +. Thus if A and B are two lists (such as [a, b, c] and [d, e] respectively), A + B will denote the result of appending lists A and B (namely [a, b, c, d, e]).

There are two versions of composition: backwards and forwards. They are as follows:

Function Composition (Forward)

$$
(3.36)\quad
\left[\begin{array}{ll}
\textsc{ortho} & \mathrm{O1} \\
\textsc{cat} & \mathrm{R1} \\
\textsc{slash} & \left\|\left[\begin{array}{ll}
\textsc{ortho} & \mathrm{O2} \\
\textsc{cat} & \mathrm{R2} \\
\textsc{slash} & \mathrm{B} \\
\textsc{order} & \mathrm{fwd} \\
\textsc{feats} & \mathrm{F2} \\
\textsc{cases} & \mathrm{C2} \\
\textsc{sem} & \mathrm{S2}
\end{array}\right]\right\| \\
\textsc{order} & \mathrm{O1} \\
\textsc{feats} & \mathrm{F1} \\
\textsc{cases} & \mathrm{C1} \\
\textsc{sem} & \mathrm{S1}
\end{array}\right]
\left[\begin{array}{ll}
\textsc{ortho} & \mathrm{O2} \\
\textsc{cat} & \mathrm{R2} \\
\textsc{slash} & \mathrm{C} + \mathrm{B} \\
\textsc{order} & \mathrm{fwd} \\
\textsc{feats} & \mathrm{F2} \\
\textsc{vfinal} & \mathrm{V2} \\
\textsc{cases} & \mathrm{C2} \\
\textsc{sem} & \mathrm{S2}
\end{array}\right]
$$

$$
\longrightarrow
\left[\begin{array}{ll}
\textsc{ortho} & \mathrm{O1\ O2} \\
\textsc{cat} & \mathrm{R1} \\
\textsc{slash} & \mathrm{C} \\
\textsc{order} & \mathrm{O1} \\
\textsc{feats} & \mathrm{F1} \\
\textsc{lex} & - \\
\textsc{vfinal} & \mathrm{V2} \\
\textsc{cases} & \mathrm{C1} \\
\textsc{sem} & \mathrm{S1}
\end{array}\right]
$$

Function Composition (Backward)

$$
(3.37)\quad
\begin{bmatrix}
\textsc{ortho} & \mathrm{O2} \\
\textsc{cat} & \mathrm{R2} \\
\textsc{slash} & \mathrm{C} + \mathrm{B} \\
\textsc{order} & \mathrm{back} \\
\textsc{feats} & \mathrm{F2} \\
\textsc{cases} & \mathrm{C2} \\
\textsc{sem} & \mathrm{S2}
\end{bmatrix}
\begin{bmatrix}
\textsc{ortho} & \mathrm{O1} \\
\textsc{cat} & \mathrm{R1} \\
\textsc{slash} & \left\Vert \begin{bmatrix}
\textsc{ortho} & \mathrm{O2} \\
\textsc{cat} & \mathrm{R2} \\
\textsc{slash} & \mathrm{B} \\
\textsc{order} & \mathrm{back} \\
\textsc{feats} & \mathrm{F2} \\
\textsc{cases} & \mathrm{C2} \\
\textsc{sem} & \mathrm{S2}
\end{bmatrix} \right\Vert \\
\textsc{order} & \mathrm{O1} \\
\textsc{feats} & \mathrm{F1} \\
\textsc{vfinal} & \mathrm{V1} \\
\textsc{cases} & \mathrm{C1} \\
\textsc{sem} & \mathrm{S1}
\end{bmatrix}
$$

$$
\longrightarrow
\begin{bmatrix}
\textsc{ortho} & \mathrm{O2\ O1} \\
\textsc{cat} & \mathrm{R1} \\
\textsc{slash} & \mathrm{C} \\
\textsc{order} & \mathrm{O1} \\
\textsc{feats} & \mathrm{F1} \\
\textsc{lex} & - \\
\textsc{vfinal} & \mathrm{V1} \\
\textsc{cases} & \mathrm{C1} \\
\textsc{sem} & \mathrm{S1}
\end{bmatrix}
$$

Function composition allows the subject to combine with a verb before the verb has combined with all the objects that it subcategorizes for, as in examples (3.33). A further use of this rule will be examined in the subsection on clitic climbing (3.4.7). It is well-known that the introduction of composition tends to produce spurious parses. As has been pointed out above, these parses are necessary in order to get the facts about flexible coordination right. The number of spurious parses can be very much reduced if phonological (and in particular intonational), rather than just orthographical information is represented in the signs, something that is not done here.

The code for these rules appears in the appendices.

3.4.3 Clitics

Clitic placement: the LEX feature

As was said earlier, the main restriction about the placement of clitics is that they must be immediately adjacent to the verb to which they attach. They will be to the left or the right depending on whether the verb is finite or not, and this will be represented with the ORDER feature.

The verb may *a priori* have either a lexical NP or a clitic as its complement. As there is no way of expressing logical implications (like "if my complement is a clitic, it must be to my right"), two entries are needed either for the verb or for the clitic. If we choose to have two entries for the verb, they will subcategorize for lexical or clitic complements, putting constraints on the side on which they combine. Alternatively, we may have two entries for the clitics, with different ORDER features, and impose different constraints on the finiteness of the verbs with which they combine (finiteness being a feature of the verb, appearing together with tense information and other such). Obviously, the latter solution is simpler, as it is easier to have multiple entries for the clitics than for every verb.

As NPs are sentence (verb) modifiers, some mechanism is also required for preventing NPs (as well as other constituents) from getting between the verb and the clitic.

This mechanism is provided by the LEX feature, which is just a flag stating whether a sign is a lexical entry or not. The entry for the clitic (which is syntactically a S/S) specifies that the argument S has its LEX feature unifiable with + (i.e. uninstantiated actually, given the organisation of the entries). To make this more explicit, this is what the signs look like (looking just at the CAT, SLASH, ORDER, FEAT:FIN and LEX features). This example assumes a finite verb, and the version of the clitic that combines with the finite verb.

Clitic

$$\begin{bmatrix} \mathrm{CAT} & \mathrm{s} \\ \mathrm{SLASH} & \left\| \begin{bmatrix} \mathrm{CAT} & \mathrm{s} \\ \mathrm{FEATS{:}FIN} & + \\ \mathrm{LEX} & + \end{bmatrix} \right\| \\ \mathrm{ORDER} & \text{back} \end{bmatrix}$$

Finite Transitive Verb

$$\begin{bmatrix} \mathrm{CAT} & \mathrm{s} \\ \mathrm{SLASH} & \left\| \begin{bmatrix} \mathrm{CAT} & \mathrm{s} \\ \mathrm{SLASH} & \left\| \begin{bmatrix} \mathrm{CAT} & \mathrm{s} \\ \mathrm{FEATS{:}FIN} & \boxed{1}\ + \\ \mathrm{LEX} & \boxed{2} \end{bmatrix} \right\| \end{bmatrix} \right\| \\ \mathrm{FEATS{:}FIN} & \boxed{1} \\ \mathrm{LEX} & \boxed{2} \end{bmatrix}$$

The other entry for the clitic has SLASH:FEAT:FIN set to $-$, and ORDER to *fwd.*

Now, recall that when two signs are combined, the feature LEX of the result is set to $-$. So if the verb had combined with anything else first, the resulting sign is unable to combine with a clitic. This, together with the ORDER feature, is what is required to give the correct clitic placement.

Basics: the DO clitic

With the essentials of the mechanism set up, we now examine the treatment of clitics. The basic entry for a (case absorbing) DO clitic is very similar to a NP such as (3.9), the only difference being that its case is now specified as *accusative*, and the semantics are less specific. The examples presented in this section will be the ones that combine with finite verbs.

$$
(3.38)\left[\begin{array}{ll}
\textsc{ortho} & \text{lo} \\
\textsc{cat} & \text{s} \\
\textsc{slash} & \left\|\left[\begin{array}{ll}
\textsc{cat} & \text{s} \\
\textsc{slash} & \_\text{C3} \\
\textsc{feats} & \boxed{1}\left[\textsc{fin}\ +\right] \\
\textsc{lex} & + \\
\textsc{cases} & \left\{\left[\begin{array}{ll}
\textsc{agr} & \left[\begin{array}{ll}
\textsc{pers} & 3 \\
\textsc{num} & \text{sg} \\
\textsc{gen} & \text{masc} \\
\textsc{case} & \text{acc}
\end{array}\right] \\
\textsc{index} & \text{Y3}
\end{array}\right]\right\} \cup \text{Cases3} \\
\textsc{sem} & \text{I3 : Sem3}
\end{array}\right]\right\| \\
\textsc{order} & \text{back} \\
\textsc{feats} & \boxed{1} \\
\textsc{cases} & \text{Cases3} \\
\textsc{sem} & \text{I3 : }(\{\text{role(I3,\_R3,Y3)}\} \cup \text{Sem3})
\end{array}\right]
$$

The combination with (3.10) will therefore proceed in the same way as that with the lexical NP (3.9), by means of backwards application.

$$
(3.39)\left[\begin{array}{ll}
\textsc{ortho} & \text{'lo leyó'} \\
\textsc{cat} & \text{s} \\
\textsc{slash} & \|\,\| \\
\textsc{lex} & - \\
\textsc{feats} & \left[\textsc{fin}\ -\right] \\
\textsc{cases} & \left\{\left[\begin{array}{ll}
\textsc{agr} & \left[\begin{array}{ll}
\textsc{pers} & 3 \\
\textsc{num} & \text{sg} \\
\textsc{case} & \text{nom}
\end{array}\right] \\
\textsc{index} & \text{X3}
\end{array}\right]\right\} \\
\textsc{sem} & \text{E : }\{\text{role(E,pat,Y3), leer(E), role(E,agt,X3)}\}
\end{array}\right]
$$

3.4.4 Clitic doubling

In contrast to the DO clitics, the Spanish IO clitic is an optional case absorber, and so there are two versions of it. The case-absorbing version parallels the entry for the DO clitic.

$$
(3.40)\left[\begin{array}{ll}
\textsc{ortho} & \text{le} \\
\textsc{cat} & \text{s} \\
\textsc{slash} & \left\| \left[\begin{array}{ll}
\textsc{cat} & \text{s} \\
\textsc{slash} & \text{\_C4} \\
\textsc{feats} & \boxed{1} \left[\textsc{fin} \; +\right] \\
\textsc{lex} & + \\
\textsc{cases} & \left\{ \left[\begin{array}{ll}
\textsc{agr} & \left[\begin{array}{ll} \textsc{pers} & 3 \\ \textsc{num} & \text{sg} \\ \textsc{case} & \text{dat} \end{array}\right] \\
\textsc{index} & \text{Z3}
\end{array}\right] \right\} \cup \text{Cases4} \\
\textsc{sem} & \text{I4} : \text{Sem4}
\end{array}\right] \right\| \\
\textsc{order} & \text{back} \\
\textsc{feats} & \boxed{1} \\
\textsc{cases} & \text{Cases4} \\
\textsc{sem} & \text{I4} : (\{\text{role(I4,\_R4,Y3)}\} \cup \text{Sem4})
\end{array}\right]
$$

There is in addition a version which does not absorb case, and hence will have shared values for SLASH:CASES and CASES. Of course, it still gets a role in the semantics:

$$
(3.41)\left[\begin{array}{ll}
\textsc{ortho} & \text{le} \\
\textsc{cat} & \text{s} \\
\textsc{slash} & \left\| \left[\begin{array}{ll}
\textsc{cat} & \text{s} \\
\textsc{slash} & \text{\_C5} \\
\textsc{feats} & \boxed{1} \left[\textsc{fin} \; +\right] \\
\textsc{lex} & + \\
\textsc{cases} & \boxed{2} \left(\left\{ \left[\begin{array}{ll}
\textsc{agr} & \left[\begin{array}{ll} \textsc{pers} & 3 \\ \textsc{num} & \text{sg} \\ \textsc{case} & \text{dat} \end{array}\right] \\
\textsc{index} & \text{Z3}
\end{array}\right] \right\} \cup \text{Cases5} \right) \\
\textsc{lex} & + \\
\textsc{sem} & \text{I5} : \text{Sem5}
\end{array}\right] \right\| \\
\textsc{order} & \text{back} \\
\textsc{feats} & \boxed{1} \\
\textsc{cases} & \boxed{2} \\
\textsc{sem} & \text{I5} : (\{\text{role(I5,\_R5,Y3)}\} \cup \text{Sem5})
\end{array}\right]
$$

To see this clitic in action, recall what the entry for a ditransitive verb (subcategorizing for its two objects) is like (3.21), reprinted below again. This following version takes its indirect object first.

(3.42)
$$\begin{bmatrix}
\textsc{ortho} & \text{dio} \\
\textsc{cat} & \text{s} \\
\textsc{slash} & \left\| \begin{matrix}
\begin{bmatrix}
\textsc{cat} & \text{s} \\
\textsc{slash} & \left\| \begin{bmatrix}
\textsc{cat} & \text{s} \\
\textsc{slash} & \| \, \| \\
\textsc{feats} & \boxed{1} \\
\textsc{cases} & \left\{ \begin{bmatrix} \textsc{agr} & \begin{bmatrix} \textsc{case} & \text{dat} \end{bmatrix} \\ \textsc{index} & \text{Z} \end{bmatrix} \right\} \\
\textsc{sem} & \text{Sem1}
\end{bmatrix} \right\| \\
\textsc{cases} & \text{Cases2} \\
\textsc{sem} & \text{Sem2}
\end{bmatrix} \\
\begin{bmatrix}
\textsc{cat} & \text{s} \\
\textsc{slash} & \left\| \begin{bmatrix}
\textsc{cat} & \text{s} \\
\textsc{slash} & \| \, \| \\
\textsc{feats} & \boxed{1} \\
\textsc{cases} & \left\{ \begin{bmatrix} \textsc{agr} & \begin{bmatrix} \textsc{case} & \text{acc} \end{bmatrix} \\ \textsc{index} & \text{Y} \end{bmatrix} \right\} \\
\textsc{sem} & \text{E} : \left\{ \begin{matrix} \text{dar(E),role(E,agt,X3),} \\ \text{role(E,pat,Y),role(E,goal,Z)} \end{matrix} \right\}
\end{bmatrix} \right\| \\
\textsc{cases} & \text{Cases1} \\
\textsc{sem} & \text{Sem1}
\end{bmatrix}
\end{matrix} \right\| \\
\textsc{feats} & \boxed{1} \begin{bmatrix} \textsc{fin} & + \end{bmatrix} \\
\textsc{cases} & \left\{ \begin{bmatrix} \textsc{agr} & \begin{bmatrix} \textsc{pers} & 3 \\ \textsc{num} & \text{sg} \\ \textsc{case} & \text{nom} \end{bmatrix} \\ \textsc{index} & \text{X3} \end{bmatrix} \right\} \cup \text{Cases1} \cup \text{Cases2} \\
\textsc{sem} & \text{Sem2}
\end{bmatrix}$$

This may then combine with the non-case absorbing IO clitic (3.41) and still be able to assign dative case, though it will no longer subcategorize for a dative NP:

$$
(3.43)\quad
\begin{bmatrix}
\textsc{ortho} & \text{'le dio'} \\
\textsc{cat} & \text{s} \\
\textsc{slash} & \left\|\begin{bmatrix}
\textsc{cat} & \text{s} \\
\textsc{slash} & \left\|\begin{bmatrix}
\textsc{cat} & \text{s} \\
\textsc{slash} & \|\| \\
\textsc{feats} & \boxed{1} \\
\textsc{cases} & \left\{\begin{bmatrix} \textsc{agr} & \begin{bmatrix}\textsc{case} & \text{acc}\end{bmatrix} \\ \textsc{index} & \text{Y} \end{bmatrix}\right\} \\
\textsc{sem} & \text{E} : \left\{\begin{array}{l} \text{dar(E),role(E,agt,X3),} \\ \text{role(E,pat,Y),role(E,goal,Z3)} \end{array}\right\}
\end{bmatrix}\right\| \\
\textsc{cases} & \text{Cases1} \\
\textsc{sem} & \text{Sem1}
\end{bmatrix}\right\| \\
\textsc{feats} & \boxed{1}\begin{bmatrix}\textsc{fin} & +\end{bmatrix} \\
\textsc{lex} & - \\
\textsc{cases} & \left\{\begin{bmatrix} \textsc{agr} & \begin{bmatrix} \textsc{pers} & 3 \\ \textsc{num} & \text{sg} \\ \textsc{case} & \text{nom} \end{bmatrix} \\ \textsc{index} & \text{X3} \end{bmatrix} \begin{bmatrix} \textsc{agr} & \begin{bmatrix} \textsc{pers} & 3 \\ \textsc{num} & \text{sg} \\ \textsc{case} & \text{dat} \end{bmatrix} \\ \textsc{index} & \text{Z3} \end{bmatrix}\right\} \cup \text{Cases1} \\
\textsc{sem} & \text{Sem2}
\end{bmatrix}
$$

At this stage, it can then combine, just as above, with a direct object NP such as (3.9), giving:

$$
(3.44)\quad
\begin{bmatrix}
\textsc{ortho} & \text{'le dio el libro'} \\
\textsc{cat} & \text{s} \\
\textsc{slash} & \|\| \\
\textsc{feats} & \begin{bmatrix}\textsc{fin} & +\end{bmatrix} \\
\textsc{lex} & - \\
\textsc{cases} & \left\{\begin{bmatrix} \textsc{agr} & \begin{bmatrix} \textsc{pers} & 3 \\ \textsc{num} & \text{sg} \\ \textsc{case} & \text{nom} \end{bmatrix} \\ \textsc{index} & \text{X3} \end{bmatrix} \begin{bmatrix} \textsc{agr} & \begin{bmatrix} \textsc{pers} & 3 \\ \textsc{num} & \text{sg} \\ \textsc{case} & \text{dat} \end{bmatrix} \\ \textsc{index} & \text{Y3} \end{bmatrix}\right\} \\
\textsc{sem} & \text{E} : (\left\{\begin{array}{l} \text{role(E,goal,Y3), role(E,pat,B),} \\ \text{libro(B), definite(B), dar(E), role(E,agt,X3)} \end{array}\right\}
\end{bmatrix}
$$

The resulting phrase no longer subcategorizes for an IO (and hence is a complete sentence), but can still assign a dative case to any NP that acts as a modifier, such as (3.24), reprinted here:

$$(3.45)\quad \begin{bmatrix} \textsc{ortho} & \text{'a María'} \\ \textsc{cat} & \text{s} \\ \textsc{slash} & \left\| \begin{bmatrix} \textsc{cat} & \text{s} \\ \textsc{slash} & \_\text{C6} \\ \textsc{feats} & \text{Feats6} \\ \textsc{cases} & \left\{ \begin{bmatrix} \textsc{agr} & \begin{bmatrix} \textsc{pers} & 3 \\ \textsc{num} & \text{sg} \\ \textsc{gen} & \text{fem} \\ \textsc{case} & \text{dat} \end{bmatrix} \\ \textsc{index} & \text{F3} \end{bmatrix} \right\} \cup \text{Cases6} \\ \textsc{sem} & \text{I6} : \text{Sem6} \end{bmatrix} \right\| \\ \textsc{feats} & \text{Feats6} \\ \textsc{cases} & \text{Cases6} \\ \textsc{sem} & \text{I6} : (\{\text{role(I6,\_R6,F3), name(F3, maria)}\} \cup \text{Sem6}) \end{bmatrix}$$

The result is then:

$$(3.46)\quad \begin{bmatrix} \textsc{ortho} & \text{'le dio el libro a María'} \\ \textsc{cat} & \text{s} \\ \textsc{slash} & \|\,\| \\ \textsc{feats} & \begin{bmatrix} \textsc{fin} & + \end{bmatrix} \\ \textsc{lex} & - \\ \textsc{cases} & \left\{ \begin{bmatrix} \textsc{agr} & \begin{bmatrix} \textsc{pers} & 3 \\ \textsc{num} & \text{sg} \\ \textsc{case} & \text{nom} \end{bmatrix} \\ \textsc{index} & \text{X3} \end{bmatrix} \right\} \\ \textsc{sem} & \text{E} : \left\{ \begin{array}{l} \text{role(E,goal,F3),name(F3, maria),} \\ \text{role(E,pat,B),libro(B),} \\ \text{definite(B),} \\ \text{dar(E), role(E,agt,X3)} \end{array} \right\} \end{bmatrix}$$

In this sense, the IO NP is treated in a very similar manner to a subject NP: it is optional, but if it is present, it absorbs case. This analogy can be taken further, and it could be said that the Spanish verb inflection, like the IO clitic, is an optional case absorber. We shall come back to this later.

3.4.5 A note on the "double clitic" *se lo*

It was mentioned in the last chapter that when the third person IO clitic appears next to a DO clitic, the form that it takes is *se* rather than *le*. Furthermore, the relative ordering of these two clitics is always *se lo*, whether they are proclitic or enclitic:

(3.47) a Se lo dio.
CL-IO-3sg CL-DO-3sg gave-3sg
'He/she gave it to him/her.'

b *Le lo dio.

c Dándo-se-lo.
'Giving it to him/her.'

The construction in (3.47 b) is ruled out already by the constraint that clitics may only combine with lexical entries (i.e. those that have their LEX feature set to −). *Lo dio* (cf. 3.39 & 3.43) is not such a sign, and hence *le* may not combine with it.

As far as (3.47 a & c) are concerned, nothing can intervene between *se* and *lo*, or between the clitics and the verb. It has often been claimed (e.g. [Hewson 81]) that the combination of these two clitics constitutes a lexical unit, often referred to as the "double clitic".

Such an approach is taken here, and that involves having lexical entries for the double clitics. These double clitics are sentence modifiers, as usual, but require the sentence that they modify to be essentially a ditransitive verb capable of assigning an accusative and a dative case. In a manner similar to the optional case absorption of the dative clitic, the double clitic always absorbs accusative case, but only optionally absorbs the dative one.

For example, the following is the entry for a double clitic that may combine with the finite ditransitive verb *dio* shown above (3.21), absorbing both cases.

$$
(3.48)\quad
\begin{bmatrix}
\textsc{ortho} & \text{se-lo} \\
\textsc{cat} & \text{s} \\
\textsc{slash} & \left\| \begin{bmatrix}
\textsc{cat} & \text{s} \\
\textsc{slash} & \left\| \begin{array}{l}
\begin{bmatrix}
\textsc{cat} & \text{s} \\
\textsc{slash} & \left\| \begin{bmatrix}
\textsc{cat} & \text{s} \\
\textsc{cases} & \left\{ \begin{bmatrix} \textsc{agr} & \begin{bmatrix} \textsc{case} & \text{acc} \end{bmatrix} \\ \textsc{index} & \text{Y3} \end{bmatrix} \right\} \\
\textsc{sem} & \text{E:Sem1}
\end{bmatrix} \right\| \\
\textsc{sem} & \text{E:}(\{\text{role(E,R1,Y3)}\} \cup \text{Sem1})
\end{bmatrix} \\
\begin{bmatrix}
\textsc{cat} & \text{s} \\
\textsc{slash} & \left\| \begin{bmatrix}
\textsc{cat} & \text{s} \\
\textsc{cases} & \left\{ \begin{bmatrix} \textsc{agr} & \begin{bmatrix} \textsc{case} & \text{dat} \end{bmatrix} \\ \textsc{index} & \text{Z3} \end{bmatrix} \right\} \\
\textsc{sem} & \text{E:Sem2}
\end{bmatrix} \right\| \\
\textsc{sem} & \text{E:}(\{\text{role(E,R2,Z3)}\} \cup \text{Sem2})
\end{bmatrix}
\end{array} \right\| \\
\textsc{order} & \text{back} \\
\textsc{feats} & \boxed{1} \begin{bmatrix} \textsc{fin} & + \end{bmatrix} \\
\textsc{lex} & + \\
\textsc{cases} & \text{Cases} \\
\textsc{sem} & \text{Sem3}
\end{bmatrix} \right\| \\
\textsc{feats} & \boxed{1} \\
\textsc{cases} & \text{Cases} \\
\textsc{sem} & \text{Sem3}
\end{bmatrix}
$$

The result of the combination (by forwards application) is simply:

$$
(3.49)\quad
\begin{bmatrix}
\textsc{ortho} & \text{'se lo dio'} \\
\textsc{cat} & \text{s} \\
\textsc{slash} & \|\,\| \\
\textsc{feats} & \begin{bmatrix} \textsc{fin} & + \end{bmatrix} \\
\textsc{lex} & - \\
\textsc{cases} & \left\{ \begin{bmatrix} \textsc{agr} & \begin{bmatrix} \textsc{pers} & 3 \\ \textsc{num} & \text{sg} \\ \textsc{case} & \text{nom} \end{bmatrix} \\ \textsc{index} & \text{X3} \end{bmatrix} \right\} \\
\textsc{sem} & \text{E} : \left\{ \begin{array}{l} \text{role(E,goal,Z3),role(E,pat,Y3),} \\ \text{dar(E), role(E,agt,X3)} \end{array} \right\}
\end{bmatrix}
$$

3.4.6 Dislocation

Dislocated NPs (that is, those which are separated from the rest of the sentence by an intonational break) may, unlike non-dislocated ones, co-occur with the corresponding clitic. Let us follow the explanation in [Jaeggli 86] by saying that they do not absorb case (they will be licensed by discourse conditions which are omitted here, and just hinted at in the following semantics):

$$
(3.50)\begin{bmatrix}
\text{ORTHO} & \textbf{'el libro,'} \\
\text{CAT} & \mathrm{s} \\
\text{SLASH} & \left\| \begin{bmatrix} \text{CAT} & \mathrm{s} \\ \text{SLASH} & \|\,\| \\ \text{ORDER} & \mathrm{fwd} \\ \text{FEATS} & \mathrm{Feats7} \\ \text{CASES} & \mathrm{Cases7} \\ \text{SEM} & \mathrm{I7} : \mathrm{Sem7} \end{bmatrix} \right\| \\
\text{FEATS} & \mathrm{Feats7} \\
\text{CASES} & \mathrm{Cases7} \\
\text{SEM} & \mathrm{I7} : (\{\mathrm{role(I7,\_R7,B),\ libro(B),\ definite(B),\ focus(B)}\} \cup \mathrm{Sem7})
\end{bmatrix}
$$

It seems likely that dislocation is characterised by the comma, representing an intonational break. Consequently, the comma should be treated as a functor from ordinary case absorbing NPs into non-case absorbing ones such as (3.50) above. Furthermore, if phonology is represented (as it is crudely done here following the conventions of Section 2.3), some suitable constraints should be attached to it. The following entry derives the above dislocated NP (3.50) from the non-dislocated one (3.9).

$$
(3.51)\quad
\left[\begin{array}{ll}
\textsc{ortho} & \textbf{boldface ,} \\
\textsc{cat} & \mathrm{s} \\
\textsc{slash} & \left\| \begin{array}{l}
\left[\begin{array}{ll}
\textsc{cat} & \mathrm{s} \\
\textsc{slash} & \left\| \left[\begin{array}{ll}
\textsc{cat} & \mathrm{s} \\
\textsc{slash} & \|\,\| \\
\textsc{feats} & \mathrm{Feats7} \\
\textsc{cases} & \left\{ \left[\textsc{index}\;\; \mathrm{X}\right] \right\} \cup \mathrm{Cases7} \\
\textsc{sem} & \mathrm{I7{:}Sem7}
\end{array}\right] \right\| \\
\textsc{order} & \mathrm{back} \\
\textsc{feats} & \mathrm{Feats7} \\
\textsc{cases} & \mathrm{Cases7} \\
\textsc{sem} & \mathrm{I7} : \mathrm{Sem8}
\end{array}\right] \\
\left[\begin{array}{ll}
\textsc{cat} & \mathrm{s} \\
\textsc{slash} & \|\,\| \\
\textsc{order} & \mathrm{fwd} \\
\textsc{feats} & \mathrm{Feats7} \\
\textsc{cases} & \mathrm{Cases7} \\
\textsc{sem} & \mathrm{I7} : \mathrm{Sem7}
\end{array}\right]
\end{array} \right\| \\
\textsc{feats} & \mathrm{Feats7} \\
\textsc{cases} & \mathrm{Cases7} \\
\textsc{sem} & \mathrm{I7} : (\{\mathrm{focus(X)}\} \cup \mathrm{Sem8})
\end{array}\right]
$$

The dislocated NP (3.50) may therefore combine with a complete sentence such as *lo leyó* (3.39), and impose some constraints on the semantics to yield:

$$
(3.52)\quad
\left[\begin{array}{ll}
\textsc{ortho} & \text{'}\textbf{el libro,}\text{ lo leyó'} \\
\textsc{cat} & \mathrm{s} \\
\textsc{slash} & \|\,\| \\
\textsc{feats} & \left[\textsc{fin}\;\; +\right] \\
\textsc{lex} & \text{—} \\
\textsc{cases} & \left\{ \left[\begin{array}{ll}
\textsc{agr} & \left[\begin{array}{ll}
\textsc{pers} & 3 \\
\textsc{num} & \mathrm{sg} \\
\textsc{case} & \mathrm{nom}
\end{array}\right] \\
\textsc{index} & \mathrm{X3}
\end{array}\right] \right\} \\
\textsc{sem} & \mathrm{E} : \{\mathrm{role(E,pat,B),\ libro(B),\ definite(B),\ focus(B),\ leer(E),\ role(E,agt,X3)}\}
\end{array}\right]
$$

The exact nature of the semantic constraints will not be discussed here. The kind of requirements that one would expect to find are, for instance, that the dislocated element should have previously appeared in the discourse, and that it should contribute something towards the semantics of the resulting sentence. This rules out the ungrammatical multiple applications of expressions like (3.52) to the sentence (3.39).

One other issue that has not been touched on is how to ensure that the dislocated element agrees in gender and number with the clitic. Gender and number should be treated as a syntactic feature of the dislocated phrase, and there should be some part of the sign for (3.39) to indicate that the object should be masculine singular. This requires a complex sign to keep track of the syntactic features of its constituent signs.

This treatment of dislocated phrases is very similar to the one offered in [Sanfilippo 90b]. Sanfilippo's **thematic-domain** performs a very similar job to the CASE feature here, the main difference being that the former is more part of the semantics, while the latter is more syntax-oriented. Having an explicit case-assignment mechanism distinct from the subcategorization allows a treatment of "real" clitic-doubling (i.e. in sentences with a single intonation pattern), which Sanfilippo does not cover as it does not occur in Italian.

3.4.7 Clitic Climbing

We now turn to data involving clitic climbing, as illustrated in (3.53).

(3.53)

a

Quiere	leer-lo.
Want-3sg	read-CL-do

'He/She wants to read it.'

b

Lo	quiere	leer.
CL-do	want-3sg	read

c

Quiere	poder	leer-lo.
Want-3sg	can	read-CL-do

'He/She wants to be able to read it.'

d Quiere poder-lo leer.

e Lo quiere poder leer.

f

Queriéndo-lo	leer.
Wanting-CL-do	to read

'Wanting to read it.'

g

Queriéndo-lo	y	pudiéndo-lo	leer.
Wanting-CL-do	and	being-able-CL-do	to read

'Wanting and being able to read it.'

h

Lo	quiero	y	lo	puedo	leer.
CL-DO	want-1sg	and	CL-DO	be-able-1sg	to read

'I want and am able to read it.'

A possible explanation of what goes on in examples (3.53 a & b) (and also examples (3.53 c to e)) above is that *quiere* subcategorizes for an infinitive VP, and "inherits" any element that its argument subcategorizes for. Therefore, if the lower verb *leer* has not combined with a clitic, *quiere leer* will still be looking for an object, which may be a clitic. This "inheritance" of the SLASH is exactly what function composition gives us. The order in which the clitic comes is determined by the fact that *quiere leer* behaves like a tensed verb, and hence any clitic must precede it.

However, composing the two verbs into such a "cluster" is not quite satisfactory. Firstly, we would be at a loss to explain example (3.53 f), without having recourse to discontinuous constituents. Secondly, examples (3.53 g & h) show further evidence that the higher verb forms a constituent with the clitic. Finally, this does not explain the fact that contrastive stress can fall on either of the two verbs (the higher or the lower one).

What is therefore required is the possibility of combining the matrix verb with the clitic that has climbed to a position adjacent to it. However, the matrix verb does not subcategorize for the clitic, nor is it capable of assigning case to it. We shall see shortly how this combination may be achieved, but for the time being let us see how these sentences with the clitic attaching to the lower verb are formed.

Let the entry for *leer* be:

$$
(3.54)\quad
\begin{bmatrix}
\textsc{ortho} & \text{leer} \\
\textsc{cat} & \text{vp\_inf} \\
\textsc{slash} & \left\| \begin{bmatrix}
\textsc{cat} & \text{s} \\
\textsc{slash} & \left\| \begin{bmatrix}
\textsc{cat} & \text{s} \\
\textsc{slash} & \| \, \| \\
\textsc{feats} & \boxed{1} \\
\textsc{lex} & + \\
\textsc{cases} & \left\{ \begin{bmatrix} \textsc{agr} & \begin{bmatrix} \textsc{case} & \text{acc} \end{bmatrix} \\ \textsc{index} & \text{Y} \end{bmatrix} \right\} \\
\textsc{sem} & \text{E} : \{\text{leer(E),role(E,agt,X),role(E,pat,Y)}\}
\end{bmatrix} \right\| \\
\textsc{order} & \text{fwd} \\
\textsc{cases} & \text{Cases1} \\
\textsc{sem} & \text{Sem3}
\end{bmatrix} \right\| \\
\textsc{feats} & \boxed{1} \begin{bmatrix} \textsc{fin} & - \end{bmatrix} \\
\textsc{cases} & \left\{ \begin{bmatrix} \textsc{agr} & \begin{bmatrix} \textsc{case} & \text{nom} \end{bmatrix} \\ \textsc{index} & \text{X3} \end{bmatrix} \right\} \cup \text{Cases1} \\
\textsc{sem} & \text{Sem3}
\end{bmatrix}
$$

This is basically the same entry as the tensed verb *leyó* (3.10), but its category is no longer S (it would not constitute a sentence with a DO NP, and could not combine with a subject NP).

However it combines trivially with the DO clitic *lo*, in the following version, which is a variant of (3.38) that is used for verbs in non-finite form:

$$(3.55)\quad \begin{bmatrix} \textsc{ortho} & \text{lo} \\ \textsc{cat} & \text{s} \\ \textsc{slash} & \left\| \begin{bmatrix} \textsc{cat} & \text{s} \\ \textsc{slash} & \text{\_C3} \\ \textsc{feats} & \boxed{1} \begin{bmatrix} \textsc{fin} & - \end{bmatrix} \\ \textsc{lex} & + \\ \textsc{cases} & \left\{ \begin{bmatrix} \textsc{agr} & \begin{bmatrix} \textsc{pers} & 3 \\ \textsc{num} & \text{sg} \\ \textsc{gen} & \text{masc} \\ \textsc{case} & \text{acc} \end{bmatrix} \\ \textsc{index} & \text{Y3} \end{bmatrix} \right\} \cup \text{Cases3} \\ \textsc{sem} & \text{I3 : Sem3} \end{bmatrix} \right\| \\ \textsc{order} & \text{fwd} \\ \textsc{feats} & \boxed{1} \\ \textsc{cases} & \text{Cases3} \\ \textsc{sem} & \text{I3 : } (\{\text{role(I3,\_R3,Y3)}\} \cup \text{Sem3}) \end{bmatrix}$$

The result of the function application is:

$$(3.56)\quad \begin{bmatrix} \textsc{ortho} & \text{leer-lo} \\ \textsc{cat} & \text{vp\_inf} \\ \textsc{slash} & \|\,\| \\ \textsc{feats} & \begin{bmatrix} \textsc{fin} & - \end{bmatrix} \\ \textsc{cases} & \left\{ \begin{bmatrix} \textsc{agr} & \begin{bmatrix} \textsc{case} & \text{nom} \end{bmatrix} \\ \textsc{index} & \text{X} \end{bmatrix} \right\} \\ \textsc{sem} & \text{E : } \{\text{leer(E),role(E,agt,X),role(E,pat,Y3)}\} \end{bmatrix}$$

One entry for *quiere* (subcategorizing for an infinitival) is then:

$$
(3.57)\left[\begin{array}{ll}
\textsc{ortho} & \text{quiere} \\
\textsc{cat} & \text{s} \\
\textsc{slash} & \left\| \left[\begin{array}{ll}
\textsc{cat} & \text{vp\_inf} \\
\textsc{cases} & \left\{ \left[\begin{array}{ll}
\textsc{agr} & \left[\begin{array}{ll} \textsc{pers} & 3 \\ \textsc{num} & \text{sg} \\ \textsc{case} & \text{nom} \end{array}\right] \\
\textsc{index} & \text{X3}
\end{array}\right] \right\} \cup \text{Cases2} \\
\textsc{sem} & \text{I : Sem}
\end{array}\right] \right\| \\
\textsc{feats} & \left[\textsc{fin} \ \ +\right] \\
\textsc{cases} & \left\{ \left[\begin{array}{ll}
\textsc{agr} & \left[\begin{array}{ll} \textsc{pers} & 3 \\ \textsc{num} & \text{sg} \\ \textsc{case} & \text{nom} \end{array}\right] \\
\textsc{index} & \text{X3}
\end{array}\right] \right\} \cup \text{Cases2} \\
\textsc{sem} & \text{S : }(\{\text{querer(S), role(S,agt,X3), role(S,prop,I), role(I,\_R1,X3)}\} \cup \text{Sem})
\end{array}\right]
$$

As can be seen by the repetition of *X3* in the semantics and in the values of CASES, this is a subject control verb which may easily combine with the sign (3.56) above, to yield:

$$
(3.58)\left[\begin{array}{ll}
\textsc{ortho} & \text{'quiere leer-lo'} \\
\textsc{cat} & \text{s} \\
\textsc{slash} & \| \| \\
\textsc{feats} & \left[\textsc{fin} \ \ +\right] \\
\textsc{lex} & \text{---} \\
\textsc{cases} & \left\{ \left[\begin{array}{ll}
\textsc{agr} & \left[\begin{array}{ll} \textsc{pers} & 3 \\ \textsc{num} & \text{sg} \\ \textsc{case} & \text{nom} \end{array}\right] \\
\textsc{index} & \text{X3}
\end{array}\right] \right\} \\
\textsc{sem} & \text{S : } \left\{ \begin{array}{l} \text{querer(S), role(S,agt,X3), role(S,prop,E),} \\ \text{role(E,agt,X3), role(E,pat,Y3), leer(E)} \end{array} \right\}
\end{array}\right]
$$

In addition to this very straightforward derivation, another one is needed in order to account for clitic climbing. As was pointed out above, this requires the matrix verb *quiere* (3.57) to combine with the DO clitic first.

Recall that the entry for the DO clitic which combines with a finite verb is as follows:

$$
(3.59)\quad \left[\begin{array}{ll}
\text{ORTHO} & \text{lo}\\
\text{CAT} & \text{s}\\
\text{SLASH} & \left\|\left[\begin{array}{ll}
\text{CAT} & \text{s}\\
\text{SLASH} & \_\text{C3}\\
\text{FEATS} & \boxed{1}\left[\text{FIN}\ +\right]\\
\text{LEX} & +\\
\text{CASES} & \left\{\left[\begin{array}{ll}\text{AGR} & \left[\begin{array}{ll}\text{PERS} & 3\\ \text{NUM} & \text{sg}\\ \text{GEN} & \text{masc}\\ \text{CASE} & \text{acc}\end{array}\right]\\ \text{INDEX} & \text{Y3}\end{array}\right]\right\} \cup \text{Cases3}\\
\text{SEM} & \text{I3} : \text{Sem3}
\end{array}\right]\right\|\\
\text{ORDER} & \text{back}\\
\text{FEATS} & \boxed{1}\\
\text{CASES} & \text{Cases3}\\
\text{SEM} & \text{I3} : (\{\text{role(I3,\_R3,Y3)}\} \cup \text{Sem3})
\end{array}\right]
$$

The value of the LEX feature, in conjunction with the function combination rule, ensures that no element may intervene between the verb and the clitic. If the verb combined with anything else, that value would be set by the rule to $-$, which would prevent combination with the clitic.

It might be tempting to think that the clitic (which is a S/S) may combine with the higher verb (a S/vp_inf) by function composition (to yield a S/vp_inf). This however gives the wrong semantics. The reason is that the clitic instantiates a role in the semantics whose index is that of the verb with which it combines. But in the case of clitic climbing, the clitic should be playing a role in the semantics of the lower verb, for which the higher verb subcategorizes, and the index corresponding to the semantics of this lower verb is different from that of the higher verb.

What is therefore required is a rule other than composition to put the clitic and the higher verb together. There seem to be two ways in which to do this. To simplify the discussion, let the clitic have category A, the higher verb category B/C and the lower one C/A (where the slashes are non-directional). Clearly the two verbs can combine by functional composition to give a B/A, which may combine with the clitic by function application to give a B. As was pointed out above, this derivation is not the required one, since it produces a verb cluster: what is required is the combination of the A and the B/C somehow to produce a $B/(C/A)$.

There are two unary rules in wide use throughout the Categorial Grammar literature that make this possible, namely **Type-Raising** and **Division**. Very briefly, Type-Raising changes the category A into $D/(D/A)$ for some D, and Division takes a B/C into a $(B/D)/(C/D)$. As with other unary rules, the indiscriminate use of Type-Raising and Division leads to great inefficiencies in parsing, and if we use them, we should look for ways in which they can be restricted.

Type-raising the clitic (3.59) yields

$$\begin{bmatrix} \textsc{ortho} & \text{lo} \\ \textsc{cat} & \text{Cat} \\ \textsc{slash} & \left\| \begin{bmatrix} \textsc{cat} & \text{Cat} \\ \textsc{slash} & \left\| \begin{bmatrix} \textsc{ortho} & \text{lo} \\ \textsc{cat} & \text{s} \\ \textsc{slash} & \left\| \begin{bmatrix} \textsc{cat} & \text{s} \\ \textsc{slash} & \text{\_C3} \\ \textsc{feats} & \boxed{1} \begin{bmatrix} \textsc{fin} & + \end{bmatrix} \\ \textsc{lex} & + \\ \textsc{cases} & \left\{ \begin{matrix} \begin{bmatrix} \textsc{agr} & \begin{bmatrix} \textsc{pers} & 3 \\ \textsc{num} & \text{sg} \\ \textsc{gen} & \text{masc} \\ \textsc{case} & \text{acc} \end{bmatrix} \\ \textsc{index} & \text{Y3} \end{bmatrix} \\ \cup \text{ Cases3} \end{matrix} \right\} \\ \textsc{sem} & \text{I3 : Sem3} \end{bmatrix} \right\| \\ \textsc{order} & \text{back} \\ \textsc{feats} & \boxed{1} \\ \textsc{cases} & \text{Cases3} \\ \textsc{sem} & \text{I3 : (\{role(I3,\_R3,Y3)\}} \cup \text{Sem3)} \end{bmatrix} \right\| \\ \textsc{sem} & \text{I4:Sem4} \end{bmatrix} \right\| \\ \textsc{sem} & \text{I4:Sem4} \end{bmatrix} \tag{3.60}$$

This type-raised clitic combines with the higher verb *quiere* (3.57) by backwards composition to give:

$$
(3.61)\quad
\left[\begin{array}{ll}
\text{ORTHO} & \text{lo quiere}\\
\text{CAT} & \text{s}\\
\text{SLASH} & \left\| \left[\begin{array}{ll}
\text{CAT} & \text{vp\_inf}\\
\text{SLASH} & \left\| \left[\begin{array}{ll}
\text{ORTHO} & \text{lo}\\
\text{CAT} & \text{s}\\
\text{SLASH} & \left\| \left[\begin{array}{ll}
\text{CAT} & \text{s}\\
\text{CASES} & \left\{ \left[\begin{array}{ll}
\text{AGR} & \left[\begin{array}{ll}\text{PERS} & 3\\ \text{NUM} & \text{sg}\\ \text{GEN} & \text{masc}\\ \text{CASE} & \text{acc}\end{array}\right]\\
\text{INDEX} & \text{Y3}
\end{array}\right] \right\}\\
\text{SEM} & \text{I3:Sem3}
\end{array}\right] \right\|\\
\text{SEM} & \text{I3:}\{\text{role(I3,\_R3,Y3)}\} \cup \text{Sem3}
\end{array}\right] \right\|\\
\text{CASES} & \boxed{1}\ (\left\{ \left[\begin{array}{ll}
\text{AGR} & \left[\begin{array}{ll}\text{PERS} & 3\\ \text{NUM} & \text{sg}\\ \text{CASE} & \text{nom}\end{array}\right]\\
\text{INDEX} & \text{X3}
\end{array}\right] \right\} \cup \text{\_Cases2})\\
\text{SEM} & \text{I4 : Sem4}
\end{array}\right] \right\|\\
\text{FEATS} & \left[\text{FIN} \ +\right]\\
\text{CASES} & \boxed{1} \cup \text{Cases3}\\
\text{SEM} & \text{S : }(\left\{\begin{array}{l}\text{querer(S),role(S,agt,X3),role(S,prop,I),}\\ \text{role(I,\_R1,X3),role(I,\_R3,Y3)}\end{array}\right\} \cup \text{Sem4})
\end{array}\right]
$$

The alternative is to use the division rule with the higher verb *quiere* (3.57) to give:

$$
(3.62)\quad
\left[\begin{array}{ll}
\text{ORTHO} & \text{quiere}\\
\text{CAT} & \text{s}\\
\text{SLASH} & \left\| \boxed{1} \left[\begin{array}{ll}
\text{CAT} & \text{s}\\
\text{SLASH} & \left\| \left[\text{CAT} \ \ \text{s}\right] \right\|
\end{array}\right]
\left[\begin{array}{ll}
\text{CAT} & \text{vp\_inf}\\
\text{SLASH} & \boxed{1}\\
\text{CASES} & \text{Cases2}\\
\text{SEM} & \text{I : Sem}
\end{array}\right] \right\|\\
\text{FEATS} & \left[\text{FIN} \ +\right]\\
\text{CASES} & \boxed{1} \left\{ \left[\begin{array}{ll}
\text{AGR} & \left[\begin{array}{ll}\text{PERS} & 3\\ \text{NUM} & \text{sg}\\ \text{CASE} & \text{nom}\end{array}\right]\\
\text{INDEX} & \text{X3}
\end{array}\right] \right\} \cup \text{Cases}\\
\text{SEM} & \text{S : }(\{\text{querer(S), role(S,agt,X3), role(S,prop,I), role(I,\_R1,X3)}\} \cup \text{Sem})
\end{array}\right]
$$

This may now combine with the (non type-raised) clitic *lo* (3.38) to yield the same result as *lo quiere* (3.61) above.

It should be noted that this entry for *quiere* forces clitic climbing. If, instead of being derived by the use of a unary rule, it was built by a lexical redundancy rule based on the lexical properties of the verb, this would help to account for the fact that some verbs allow clitic climbing, while others do not, as the following example illustrates:

(3.63) a Lo quiere leer.
'He/She wants to read it.'

b *Lo odia leer.
'He/She hates to read it.'

We shall therefore avoid the use of any unary rules at all (both Type-raising and Division), and introduce a further lexical redundancy rule that builds, from a verb such as *quiere*, marked with a feature CLIMB telling us that it allows clitic climbing, a new entry which makes the clitic to "climb over" the verb. Thus there will be two entries for such a verb, and the consequences of this in terms of the complexity of translation will be addressed in Section 6.4.

Clitic – Climb

(3.64)

$$
\left[\begin{array}{ll}
\text{ORTHO} & \mathrm{O1}\\
\text{CAT} & \mathrm{Cat}\\
\text{SLASH} & \left\|\left[\begin{array}{ll}\text{CAT} & \mathrm{vp\_inf}\\ \text{SLASH} & \boxed{1}\\ \text{CASES} & \mathrm{Cases2}\\ \text{SEM} & \mathrm{Sem2}\end{array}\right]\right\|\\
\text{FEATS} & \left[\begin{array}{ll}\text{FIN} & +\\ \text{CLIMB} & +\end{array}\right]\\
\text{LEX} & \mathrm{L1}\\
\text{CASES} & \mathrm{Cases1}\\
\text{SEM} & \mathrm{S{:}Sem1}
\end{array}\right]
$$

$$
\longrightarrow
\left[\begin{array}{ll}
\text{ORTHO} & \mathrm{O1}\\
\text{CAT} & \mathrm{Cat}\\
\text{SLASH} & \left\|\ \boxed{1}\ \left[\begin{array}{ll}\text{CAT} & \text{s}\\ \text{SLASH} & \left\|\left[\text{CAT}\ \ \text{s}\right]\right\|\end{array}\right]\ \left[\begin{array}{ll}\text{CAT} & \mathrm{vp\_inf}\\ \text{SLASH} & \boxed{1}\\ \text{CASES} & \mathrm{Cases2}\\ \text{SEM} & \mathrm{Sem2}\end{array}\right]\right\|\\
\text{FEATS} & \left[\text{FIN}\ \ +\right]\\
\text{LEX} & \mathrm{L1}\\
\text{CASES} & \mathrm{Cases1}\\
\text{SEM} & \mathrm{S{:}Sem1}
\end{array}\right]
$$

This produces the above entry for *quiere* (3.62), which combines first with the clitic (by backwards application) to give (3.61), and then (by forwards application) with a *vp_inf* such as *leer* (3.54) to give the desired result.

$$
(3.65)\quad \left[\begin{array}{ll}
\textsc{ortho} & \text{'lo quiere leer'} \\
\textsc{cat} & \text{s} \\
\textsc{slash} & \|\,\| \\
\textsc{feats} & \left[\begin{array}{ll} \textsc{fin} & + \end{array}\right] \\
\textsc{lex} & - \\
\textsc{cases} & \left\{\left[\begin{array}{ll}
\textsc{agr} & \left[\begin{array}{ll} \textsc{pers} & 3 \\ \textsc{num} & \text{sg} \\ \textsc{case} & \text{nom} \end{array}\right] \\
\textsc{index} & \text{X3}
\end{array}\right]\right\} \\
\textsc{sem} & \text{S} : \left\{\begin{array}{l} \text{querer(S), role(S,agt,X3), role(S,prop,E),} \\ \text{role(E,agt,X3), role(E,pat,Y3), leer(E)} \end{array}\right\}
\end{array}\right]
$$

It would be similarly easy to build signs like (3.57) which have object control rather than subject control, and this is the kind of generalisation over the lexicon that UCG's lexical templates are very well suited for.

The proposed solution for clitic climbing (3.64) can be applied to more than one verb to account for the clitic climbing over several verbs, as in *lo quiere poder leer* (3.53 e).

The lexical rule (3.64) may take the following entry for *poder*:

$$
(3.66)\quad \left[\begin{array}{ll}
\textsc{ortho} & \text{poder} \\
\textsc{cat} & \text{vp\_inf} \\
\textsc{slash} & \left\|\left[\begin{array}{ll}
\textsc{cat} & \text{vp\_inf} \\
\textsc{cases} & \left\{\left[\begin{array}{ll}
\textsc{agr} & \left[\begin{array}{ll} \textsc{pers} & 3 \\ \textsc{num} & \text{sg} \\ \textsc{case} & \text{nom} \end{array}\right] \\
\textsc{index} & \text{X3}
\end{array}\right]\right\} \cup \text{Cases2} \\
\textsc{sem} & \text{I : Sem}
\end{array}\right]\right\| \\
\textsc{feats} & \left[\begin{array}{ll} \textsc{fin} & - \end{array}\right] \\
\textsc{cases} & \left\{\left[\begin{array}{ll}
\textsc{agr} & \left[\begin{array}{ll} \textsc{pers} & 3 \\ \textsc{num} & \text{sg} \\ \textsc{case} & \text{nom} \end{array}\right] \\
\textsc{index} & \text{X3}
\end{array}\right]\right\} \cup \text{Cases2} \\
\textsc{sem} & \text{S} : (\{\text{poder(S), role(S,agt,X3), role(S,prop,I), role(I,\_R1,X3)}\} \cup \text{Sem})
\end{array}\right]
$$

Just as when it applies to *quiere* to give (3.62) from (3.57), it will yield:

$$
(3.67)\quad \begin{bmatrix}
\text{ORTHO} & \text{poder}\\
\text{CAT} & \text{vp\_inf}\\
\text{SLASH} & \left\| \boxed{1}\begin{bmatrix}\text{CAT} & \text{s}\\ \text{SLASH} & \left\|\begin{bmatrix}\text{CAT} & \text{s}\end{bmatrix}\right\|\\ \text{SEM} & \text{I1:Sem1}\end{bmatrix} \begin{bmatrix}\text{CAT} & \text{vp\_inf}\\ \text{SLASH} & \boxed{1}\\ \text{CASES} & \text{Cases2}\\ \text{SEM} & \text{I2 : Sem2}\end{bmatrix}\right\|\\
\text{FEATS} & \begin{bmatrix}\text{FIN} & -\end{bmatrix}\\
\text{CASES} & \boxed{1}\left\{\begin{bmatrix}\text{AGR} & \begin{bmatrix}\text{PERS} & 3\\ \text{NUM} & \text{sg}\\ \text{CASE} & \text{nom}\end{bmatrix}\\ \text{INDEX} & \text{X3}\end{bmatrix}\right\} \cup \text{Cases}\\
\text{SEM} & \text{S : (}\{\text{poder(S), role(S,agt,X3), role(S,prop,I), role(I,\_R1,X3)}\} \cup \text{Sem2)}
\end{bmatrix}
$$

Function composition then combines *lo quiere* (3.61) with (3.67) to yield:

$$
(3.68)\quad \begin{bmatrix}
\text{ORTHO} & \text{lo quiere poder}\\
\text{CAT} & \text{s}\\
\text{SLASH} & \left\| \begin{bmatrix}
\text{CAT} & \text{vp\_inf}\\
\text{SLASH} & \left\| \begin{bmatrix}
\text{CAT} & \text{s}\\
\text{SLASH} & \left\| \begin{bmatrix}
\text{CAT} & \text{s}\\
\text{CASES} & \left\{\begin{bmatrix}\text{AGR} & \begin{bmatrix}\text{PERS} & 3\\ \text{NUM} & \text{sg}\\ \text{GEN} & \text{masc}\\ \text{CASE} & \text{acc}\end{bmatrix}\\ \text{INDEX} & \text{Y3}\end{bmatrix}\right\}\\
\text{SEM} & \text{I3:Sem3}
\end{bmatrix}\right\|\\
\text{SEM} & \text{I3:}\{\text{role(I3,\_R3,Y3)}\} \cup \text{Sem3}
\end{bmatrix}\right\|\\
\text{CASES} & \boxed{1}\ (\left\{\begin{bmatrix}\text{AGR} & \begin{bmatrix}\text{PERS} & 3\\ \text{NUM} & \text{sg}\\ \text{CASE} & \text{nom}\end{bmatrix}\\ \text{INDEX} & \text{X3}\end{bmatrix}\right\} \cup \text{\_Cases2)}\\
\text{SEM} & \text{I4 : Sem4}
\end{bmatrix}\right\|\\
\text{FEATS} & \begin{bmatrix}\text{FIN} & +\end{bmatrix}\\
\text{CASES} & \boxed{1} \cup \text{Cases3}\\
\text{SEM} & \text{S : (}\left\{\begin{array}{l}\text{querer(S),role(S,agt,X3),role(S,prop,T),}\\ \text{poder(T), role(T,agt,X3), role(T,prop,I),}\\ \text{role(I,\_R1,X3),role(I,\_R3,Y3)}\end{array}\right\} \cup \text{Sem4)}
\end{bmatrix}
$$

This sign looks very much like that for *lo quiere* (3.61), and it combines with *leer* (3.54) by forward application to give the final sentence.

$$(3.69)\quad \begin{bmatrix} \text{ORTHO} & \text{'lo quiere poder leer'} \\ \text{CAT} & \text{s} \\ \text{SLASH} & \| \| \\ \text{FEATS} & \begin{bmatrix}\text{FIN} & +\end{bmatrix} \\ \text{LEX} & - \\ \text{CASES} & \left\{ \begin{bmatrix} \text{AGR} & \begin{bmatrix} \text{PERS} & 3 \\ \text{NUM} & \text{sg} \\ \text{CASE} & \text{nom} \end{bmatrix} \\ \text{INDEX} & \text{X3} \end{bmatrix} \right\} \\ \text{SEM} & \text{S} : (\left\{ \begin{array}{l} \text{querer(S),role(S,agt,X3),role(S,prop,T),} \\ \text{poder(T), role(T,agt,X3), role(T,prop,E),} \\ \text{role(E,agt,X3),role(E,pat,Y3), leer(E)} \end{array} \right\} \end{bmatrix}$$

Finally, it should be noted that this treatment allows for the combination of verbs and their arguments in such a chain to proceed in many different orders. The readings however all yield the same semantics here, and this is a case of the "spurious" parses typical of categorial grammars.

3.5 Extensions

3.5.1 Other romance languages and other dialects of Spanish

It should be clear to the reader by now how to generalise this to other romance languages and dialects. As was shown in example (2.20), French and Italian clitics may not be doubled. The entries for such clitics are identical to the Spanish direct object clitic (3.38).

The Spanish DO clitic may also be doubled in some dialects. In particular, [Suñer 88] gives a thorough account of when it may be doubled in the *Porteño* dialect of Buenos Aires. There are restrictions on the specificity and animacy of the doubled NPs.

An entry for a DO clitic that may be doubled follows the same pattern as for the IO clitic (3.41) in that it does not absorb case. The extra restrictions are then added to the non-case absorbing clitic sign to bring it into line with Suñer's account. Again, this is a straightforward task for the unification-based mechanism of UCG.

3.5.2 Subject pro-drop

As was mentioned earlier, there is an analogy between the possibility of having a clitic and its corresponding NP co-occurring, and subject pro-drop. Given the present treatment, both subject and object NPs are sentence modifiers. The clitic may or may not absorb case, and this will respectively forbid or license the corresponding NP.

This suggests a similar treatment for the verb inflection. The inflection could be subcategorized for by the stem, but may or may not absorb the nominative case that the stem can assign. If it does, then the subject will not occur.

This possibility requires much further exploration, but if it were to be generalised to other languages, might provide an account of the impossibility of subject pro-drop in French, and the need for dummy subjects, by saying that French inflections never absorb case.

3.6 Conclusions

This approach to a computational grammar of Spanish allows the grammar writer to think in such terms as case assignment and thematic role assignment. A mechanism is presented that allows one to state this information explicitly in the lexical entry for the verb.

The flexibility and the generalisations over the lexicon that UCG provides are preserved, as lexical entries that are similar (such as those for verbs that allow clitic climbing mentioned in Subsection 3.4.7) may be produced by lexical redundancy rules.

Chapter 4

An English grammar

In this chapter I present a computational grammar for English which will be a very conventional Categorial approach, based on that presented in [Steedman 87], rather than the conventional UCG from the Acord grammar described in [Zeevat et al. 87] and [Calder et al. 88], which is not quite so suitable for handling phenomena such as non-constituent coordination and topicalisation.

An alternative approach would have been to write a grammar for English as similar as possible to the Spanish grammar of the previous chapter, so that the bilingual correspondences arising from these would be more transparent. Indeed it could be argued that having all NPs as sentence modifiers in English as well as in Spanish seems like a reasonable approach given the neo-Davidsonian treatment of the semantics, since no explicit distinction is made in the semantics of the complete sentence between the subject and the objects of the verb. Recall that this approach to Spanish was motivated by the possibility of subject pro-drop in that language. This does not arise in English, where finite verbs subcategorize for their subjects. It follows from this that an English grammar along the lines of the Spanish one, though in principle feasible, would be somewhat awkward to write and understand.

There is however a more important reason for choosing to write a more conventional grammar for English, and that is to demonstrate how, under the current approach to MT, the monolingual components of the system can be written with great independence of each other, while still keeping the bilingual correspondences transparent.

4.1 Outline of the grammar

What follows is therefore an uncontroversial treatment of English grammar under a categorial framework. The only thing that is kept from the Spanish grammar is the case-assignment mechanism. Arguments for having this have been widely discussed in the generative grammar literature [Chomsky 86], [Vergnaud 82], but retain much of their validity in the context of categorial grammars. The main use of a case assignment mechanism in Generative Grammar is to work in conjunction with the Case Filter, which states that every phonetically realised NP must receive (abstract) case. In categorial grammar there is a subcategorization mechanism, which to a large extent fulfills an analogous role (namely ensuring that all and only the required NPs are present).

4.2 Sample entries

From a syntactic point of view, we have a very conventional categorial grammar with the following basic categories: *n, np, s*. VPs and intransitive verbs will then receive the category s/np, transitive verbs will be s/np/np, determiners np/n, adjectives n/n, and so on. Note that, as in the previous chapter, these slashes are non-directional, since in UCG the directionality is handled by the ORDER feature of the argument sign.

The most obvious difference between the two grammars is the treatment of NPs, which are here considered to be basic categories, like the following:

$$(4.1)\quad \begin{bmatrix} \textsc{ortho} & \text{'Mary'} \\ \textsc{cat} & \text{np} \\ \textsc{slash} & \|\,\| \\ \textsc{feats} & \begin{bmatrix} \textsc{person} & 3 \\ \textsc{number} & \text{sg} \\ \textsc{gender} & \text{fem} \end{bmatrix} \\ \textsc{case} & \text{C} \\ \textsc{sem} & \text{F3} : \{\text{name(F3,mary)}\} \end{bmatrix}$$

The semantics of this are also quite different from the Spanish one, and this results from the fact that the above sign will never act as a functor over the sentence. Notice also that another consequence is that the CASE feature is much simpler, since it no longer needs to be treated as a difference list.

We can now look at an intransitive verb that parallels the corresponding entry in the Spanish grammar. It will subcategorize for a NP such as the one above.

$$(4.2)\quad \begin{bmatrix} \textsc{ortho} & \text{sings} \\ \textsc{cat} & \text{s} \\ \textsc{slash} & \left\| \begin{bmatrix} \textsc{cat} & \text{np} \\ \textsc{cat} & \text{subj} \\ \textsc{order} & \text{back} \\ \textsc{feats} & \begin{bmatrix} \textsc{person} & 3 \\ \textsc{number} & \text{sg} \end{bmatrix} \\ \textsc{sem} & \text{X3:Sem} \end{bmatrix} \right\| \\ \textsc{feats} & \begin{bmatrix} \textsc{fin} & + \end{bmatrix} \\ \textsc{sem} & \text{E} : \{\text{singing(E), role(E,agt,X3)}\} \cup \text{Sem} \end{bmatrix}$$

This is no longer a complete sentence (as it was in Spanish), because it subcategorizes for a subject NP to its left.

The result of the combination of these two is just as for the Spanish examples of the last chapter (though the derivation is slightly different, as the functor and the argument are now reversed).

$$(4.3)\quad \begin{bmatrix} \textsc{ortho} & \text{'Mary sings'} \\ \textsc{cat} & \text{s} \\ \textsc{slash} & \|\,\| \\ \textsc{feats} & \begin{bmatrix} \textsc{fin} & + \end{bmatrix} \\ \textsc{lex} & - \\ \textsc{sem} & \text{E} : \{\text{singing(E), role(E,agt,F3), name(F3,mary)}\} \end{bmatrix}$$

Determiners are treated as functors over nouns:

$$(4.4)\quad \begin{bmatrix} \textsc{ortho} & \text{the} \\ \textsc{cat} & \text{np} \\ \textsc{slash} & \left\| \begin{bmatrix} \textsc{cat} & \text{noun} \\ \textsc{slash} & \| \, \| \\ \textsc{order} & \text{fwd} \\ \textsc{feats} & \text{F} \\ \textsc{case} & \text{C} \\ \textsc{sem} & \text{X:Sem} \end{bmatrix} \right\| \\ \textsc{feats} & \text{F} \\ \textsc{case} & \text{C} \\ \textsc{sem} & \text{X} : \{\text{definite(X)}\} \cup \text{Sem} \end{bmatrix}$$

This may combine by forward function application with a common noun such as the following:

$$(4.5)\quad \begin{bmatrix} \textsc{ortho} & \text{book} \\ \textsc{cat} & \text{noun} \\ \textsc{slash} & \| \, \| \\ \textsc{case} & \text{C} \\ \textsc{feats} & \begin{bmatrix} \textsc{person} & 3 \\ \textsc{number} & \text{sg} \end{bmatrix} \\ \textsc{sem} & \text{B} : \{\text{book(B)}\} \end{bmatrix}$$

The result is the following NP:

$$(4.6)\quad \begin{bmatrix} \textsc{ortho} & \text{'the book'} \\ \textsc{cat} & \text{np} \\ \textsc{case} & \text{C} \\ \textsc{feats} & \begin{bmatrix} \textsc{person} & 3 \\ \textsc{number} & \text{sg} \end{bmatrix} \\ \textsc{sem} & \text{B} : \{\text{book(B), definite(B)}\} \end{bmatrix}$$

4.3 Order of constituents

Similarly, a transitive verb such as *read* subcategorizes for its object and its subject, as follows:

$$
(4.7)\quad \left[\begin{array}{ll}
\textsc{ortho} & \text{read} \\
\textsc{cat} & \text{s} \\
\textsc{slash} & \left\| \begin{array}{l}
\left[\begin{array}{ll} \textsc{cat} & \text{np} \\ \textsc{order} & \text{fwd} \\ \textsc{case} & \text{obj} \\ \textsc{sem} & \text{Y : Sem1} \end{array}\right] \\
\left[\begin{array}{ll} \textsc{cat} & \text{np} \\ \textsc{order} & \text{back} \\ \textsc{case} & \text{subj} \\ \textsc{sem} & \text{X3 : Sem2} \end{array}\right]
\end{array}\right\| \\
\textsc{feats} & \left[\textsc{fin}\ \ +\right] \\
\textsc{sem} & \text{E} : \left\{\begin{array}{l} \text{reading(E),role(E,agt,X3)} \\ \text{role(E,pat,Y)} \end{array}\right\} \cup \text{Sem1} \cup \text{Set2}
\end{array}\right]
$$

The order of English constituents is in many ways simpler than in Spanish. The mechanisms introduced in the last chapter to handle free word order (such as the lexical redundancy rules that "scramble" the elements of the SUBCAT list) were built "on top" of UCG. All the standard UCG is still there, and therefore a verb such as the ditransitive *give* just subcategorizes for its arguments in the usual way, by taking its objects in the correct order. In order to handle dative alternation, two different lexical entries are needed, but these can be derived using lexical redundancy rules as usual. The entry that combines with two object NPs is as follows:

$$
(4.8)\quad \left[\begin{array}{ll}
\textsc{ortho} & \text{gave} \\
\textsc{cat} & \text{s} \\
\textsc{slash} & \left\| \begin{array}{l}
\left[\begin{array}{ll} \textsc{cat} & \text{np} \\ \textsc{order} & \text{fwd} \\ \textsc{case} & \text{obj} \\ \textsc{sem} & \text{Y : Sem1} \end{array}\right] \\
\left[\begin{array}{ll} \textsc{cat} & \text{np} \\ \textsc{order} & \text{fwd} \\ \textsc{case} & \text{obj} \\ \textsc{sem} & \text{Z : Sem2} \end{array}\right] \\
\left[\begin{array}{ll} \textsc{cat} & \text{np} \\ \textsc{order} & \text{back} \\ \textsc{case} & \text{subj} \\ \textsc{sem} & \text{X : Sem3} \end{array}\right]
\end{array}\right\| \\
\textsc{feats} & \left[\textsc{fin}\ \ +\right] \\
\textsc{sem} & \text{E} : \left\{\begin{array}{l} \text{giving(E),role(E,agt,X3)} \\ \text{role(E,pat,Y),role(E,goal,Z)} \end{array}\right\} \cup \text{Sem1} \cup \text{Sem2} \cup \text{Sem3}
\end{array}\right]
$$

This verb combines with the two objects that it subcategorizes for (the dative first and then the accusative), in exactly the same way as its Spanish counterpart, and finally combines with its subject argument, instead of being modified by it, which is what happens in Spanish.

The VP will therefore be formed in a straightforward way by combining first with (4.1) and then with the direct object *the book* (4.6). The result of the two combinations is then:

$$(4.9)\ \left[\begin{array}{ll} \textsc{ortho} & \text{'gave Mary the book'} \\ \textsc{cat} & \text{s} \\ \textsc{slash} & \left\| \left[\begin{array}{ll} \textsc{cat} & \text{np} \\ \textsc{case} & \text{Subj} \\ \textsc{sem} & \text{X : Sem3} \end{array}\right] \right\| \\ \textsc{feats} & \left[\textsc{fin}\ +\right] \\ \textsc{sem} & \text{E} : \left\{\begin{array}{l} \text{giving(E),role(E,agt,X3),} \\ \text{role(E,pat,M),name(M,mary),} \\ \text{role(E,goal,B),book(B),definite(B)} \end{array}\right\} \cup \text{Sem3} \end{array}\right]$$

The VP now subcategorizes for its subject, and may combine with *John*, which is a NP similar to 4.1:

$$(4.10)\ \left[\begin{array}{ll} \textsc{ortho} & \text{'John gave Mary the book'} \\ \textsc{cat} & \text{s} \\ \textsc{slash} & \|\ \| \\ \textsc{feats} & \left[\textsc{fin}\ +\right] \\ \textsc{sem} & \text{E} : \left\{\begin{array}{l} \text{give(E),role(E,agt,M),name(M,john),} \\ \text{role(E,pat,B),book(B),definite(B),} \\ \text{role(E,goal,F), name(F,mary),} \end{array}\right\} \end{array}\right]$$

4.4 Dummy subjects:

English expletive subjects have a straightforward though perhaps *ad hoc* treatment: verbs that require them simply subcategorize for them (assigning case), but these subjects play no role in the semantics.

The simplest example of this are the "weather" verbs, such as *rains*, which just subcategorizes for its dummy subject *it*, as shown below:

$$(4.11)\quad \begin{bmatrix} \textsc{ortho} & \text{rains} \\ \textsc{cat} & \text{s} \\ \textsc{slash} & \left\| \begin{bmatrix} \textsc{ortho} & \text{it} \\ \textsc{cat} & \text{np} \\ \textsc{order} & \text{back} \\ \textsc{case} & \text{subj} \end{bmatrix} \right\| \\ \textsc{feats} & \begin{bmatrix} \textsc{fin} & + \end{bmatrix} \\ \textsc{sem} & \text{S} : \{\text{raining(S)}\} \end{bmatrix}$$

A slightly more complicated entry is that for *seems*, subcategorizing both for its dummy subject *it* and its *that-* complement:

$$(4.12)\quad \begin{bmatrix} \textsc{ortho} & \text{seems} \\ \textsc{cat} & \text{s} \\ \textsc{slash} & \left\| \begin{matrix} \begin{bmatrix} \textsc{ortho} & \text{O} \\ \textsc{cat} & \text{s} \\ \textsc{feats} & \begin{bmatrix} \textsc{fin} & + \\ \textsc{compl} & \text{that} \end{bmatrix} \\ \textsc{order} & \text{fwd} \\ \textsc{sem} & \text{E:Sem} \end{bmatrix} \\ \begin{bmatrix} \textsc{ortho} & \text{it} \\ \textsc{cat} & \text{np} \\ \textsc{order} & \text{back} \\ \textsc{case} & \text{subj} \end{bmatrix} \end{matrix} \right\| \\ \textsc{feats} & \begin{bmatrix} \textsc{fin} & + \end{bmatrix} \\ \textsc{sem} & \text{S} : \{\text{seeming(S),role(S,prop,E)}\} \cup \text{Sem} \end{bmatrix}$$

4.5 Clitics

English clitics are very straightforward compared with the Spanish ones: in terns of the explanation given in Chapter 2 for the distribution of Spanish clitics, they always absorb case, and therefore are in strict complementary distribution with the lexical NPs (assuming one intonational phrase). They always follow the verb to which they attach.

$$(4.13)\quad \begin{bmatrix} \textsc{ortho} & \text{her} \\ \textsc{cat} & \text{np} \\ \textsc{order} & \text{fwd} \\ \textsc{case} & \text{obj} \\ \textsc{sem} & \text{F3} : \{\} \end{bmatrix}$$

The only question that arises is what blocks clitic climbing in English:

(4.14) a I tried to see her.
  b *I her tried to see.
  c *I tried her to see.
  d *Trying her to see.

Example (4.14 b) is stopped by linear ordering conditions (the English clitic may not precede its verb): this is controlled with the ORDER feature of 4.13.

The more important question is how to prevent examples (4.14 c & d). Recall that the way in which clitic climbing in Spanish was achieved was by combining the matrix verb and the clitic together, using a new entry for the verb, built from a lexical rule for clitic climbing. Clearly then, the way to stop such examples as (4.14 c & d) in English is not to have this lexical rule.

The following entries are broadly similar to their Spanish counterparts:

$$(4.15)\begin{bmatrix} \text{ORTHO} & \text{'to see'} \\ \text{CAT} & \text{s} \\ \text{SLASH} & \left\| \begin{matrix} \begin{bmatrix} \text{CAT} & \text{np} \\ \text{ORDER} & \text{fwd} \\ \text{CASE} & \text{obj} \\ \text{SEM} & \text{Y : Sem1} \end{bmatrix} \\ \begin{bmatrix} \text{CAT} & \text{np} \\ \text{ORDER} & \text{back} \\ \text{CASE} & \text{subj} \\ \text{SEM} & \text{X : Sem2} \end{bmatrix} \end{matrix} \right\| \\ \text{FEATS} & \begin{bmatrix} \text{FIN} & - \end{bmatrix} \\ \text{SEM} & \text{E} : \left\{ \begin{matrix} \text{seeing(E),role(E,agt,X),} \\ \text{role(E,pat,Y)} \end{matrix} \right\} \cup \text{Sem1} \cup \text{Sem2} \end{bmatrix}$$

This is basically the same as for Spanish, as is:

$$
(4.16)\begin{bmatrix}
\textsc{ortho} & \text{tried} \\
\textsc{cat} & \text{s} \\
\textsc{slash} & \left\| \begin{bmatrix}
\textsc{cat} & \text{s} \\
\textsc{slash} & \left\| \boxed{1} \begin{bmatrix} \textsc{cat} & \text{np} \\ \textsc{order} & \text{back} \\ \textsc{case} & \text{subj} \\ \textsc{sem} & \text{X : Sem2} \end{bmatrix} \right\| \\
\textsc{sem} & \text{I : Sem1}
\end{bmatrix} \boxed{1} \right\| \\
\textsc{feats} & \begin{bmatrix} \textsc{fin} & + \end{bmatrix} \\
\textsc{sem} & \text{S : }(\{\text{try(S), past(S), role(S,agt,X), role(S,prop,I)}\} \cup \text{Sem1} \cup \text{Sem2})
\end{bmatrix}
$$

The two signs may combine with the clitic *her* in two different ways: the two verbs may combine first by function composition, and the result of that may combine with the clitic by function application. Alternatively, the lower verb may combine with the clitic first (function application), and the result of that may be the argument of the matrix verb (function application again). This is a case of spurious ambiguity so common in many categorial grammars. We shall only follow the second of these derivations. The lower verb *to see* (4.15) first combines with the clitic, giving:

$$
(4.17)\begin{bmatrix}
\textsc{ortho} & \text{'to see her'} \\
\textsc{cat} & \text{s} \\
\textsc{slash} & \left\| \begin{bmatrix} \textsc{cat} & \text{np} \\ \textsc{order} & \text{back} \\ \textsc{case} & \text{subj} \\ \textsc{sem} & \text{X : Sem2} \end{bmatrix} \right\| \\
\textsc{feats} & \begin{bmatrix} \textsc{fin} & - \end{bmatrix} \\
\textsc{sem} & \text{E : }(\{\text{see(E),role(E,agt,X),role(E,pat,F)}\} \cup \text{Sem2})
\end{bmatrix}
$$

This combines with the upper verb *tried* (4.16), yielding:

$$
(4.18)\begin{bmatrix}
\textsc{ortho} & \text{'tried to see her'} \\
\textsc{cat} & \text{s} \\
\textsc{slash} & \left\| \begin{bmatrix} \textsc{cat} & \text{np} \\ \textsc{order} & \text{back} \\ \textsc{case} & \text{subj} \\ \textsc{sem} & \text{X : Sem2} \end{bmatrix} \right\| \\
\textsc{feats} & \begin{bmatrix} \textsc{fin} & + \end{bmatrix} \\
\textsc{sem} & \text{S : }\left(\left\{ \begin{array}{l} \text{try(S), past(S), role(S,agt,X),} \\ \text{role(S,prop,I), role(I,agt,X), role(I,pat,F)} \end{array} \right\} \cup \text{Sem2}\right)
\end{bmatrix}
$$

4.6 Type-raising

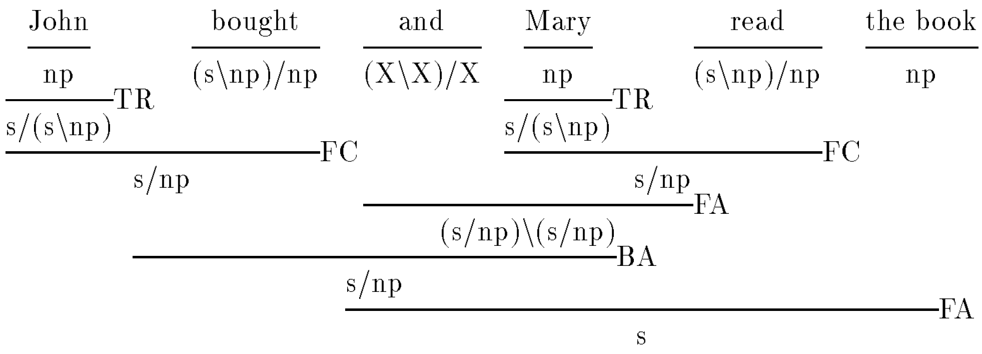
As the grammar stands so far, it is impossible to do so-called non-constituent coordination. Assuming that the conjunction *and* allows us to build a constituent of category X from two constituents of category X (we shall not go into further details about coordination here), in order to parse a sentence like (4.19) we require the phrases *John bought* and *Mary read* to be constituents.

(4.19) John bought and Mary read the book.

With the grammar as described so far, this cannot be achieved, since the transitive verb must combine first with its object before combining with the subject.

The solution to this is well known in the Categorial Grammar literature ([Steedman 85, 87]): using Steedman's notation, we require the subject to be type-raised from NP to S/(S\NP). It may then combine (by function composition) with the transitive verb (with category (S\NP)/NP), in order to give a constituent with category S/NP.

The resulting derivation of (4.19) is as follows, where FA and BA stand for forward and backward function application, FC and BC for forward and backward composition, and TR for type-raising.



Instead of having type-raising rules like Steedman, standard UCG has NPs type-raised in the lexicon. [Steedman 85] states that type-raising can be seen as a syntactic unary rule, or as a lexical redundancy rule, and [Steedman 90] draws a parallel between subject type-raising and case assignment. This can be implemented here by building subject type-raised NPs from simple ones such as (4.1) by means of the following lexical redundancy rule, which applies to lexical NPs as well as to determiners (which are NPs subcategorizing for a noun). The operator + stands, as before, for list concatenation.

NP – SubjTRNP

(4.20)

$$
\begin{array}{l}
\left[\begin{array}{ll}
\textsc{ortho} & \mathrm{O} \\
\textsc{cat} & \mathrm{np} \\
\textsc{slash} & \mathrm{S} \\
\textsc{feats} & \mathrm{F} \\
\textsc{case} & \mathrm{C} \\
\textsc{sem} & \mathrm{Sem1}
\end{array}\right] \\
\longrightarrow
\left[\begin{array}{ll}
\textsc{ortho} & \mathrm{O} \\
\textsc{cat} & \mathrm{s} \\
\textsc{slash} & \mathrm{S} + \left\| \left[\begin{array}{ll}
\textsc{cat} & \mathrm{s} \\
\textsc{slash} & \left\| \left[\begin{array}{ll}
\textsc{cat} & \mathrm{np} \\
\textsc{slash} & \|\,\| \\
\textsc{order} & \mathrm{back} \\
\textsc{feats} & \mathrm{F} \\
\textsc{case} & \mathrm{subj} \\
\textsc{sem} & \mathrm{Sem1}
\end{array}\right] \right\| \\
\textsc{order} & \mathrm{fwd} \\
\textsc{feats} & \mathrm{F2} \\
\textsc{sem} & \mathrm{Sem2}
\end{array}\right] \right\| \\
\textsc{feats} & \mathrm{F2} \\
\textsc{sem} & \mathrm{Sem2}
\end{array}\right]
\end{array}
$$

This builds, from the standard entry for *Mary* (4.1), the following type-raised one:

$$
(4.21)\;\begin{bmatrix}
\textsc{ortho} & \text{'Mary'} \\
\textsc{cat} & \text{s} \\
\textsc{slash} & \left\| \begin{bmatrix}
\textsc{cat} & \text{s} \\
\textsc{slash} & \left\| \begin{bmatrix}
\textsc{cat} & \text{np} \\
\textsc{slash} & \|\,\| \\
\textsc{order} & \text{back} \\
\textsc{feats} & \begin{bmatrix} \textsc{person} & 3 \\ \textsc{number} & \text{sg} \\ \textsc{gender} & \text{fem} \end{bmatrix} \\
\textsc{case} & \text{subj} \\
\textsc{sem} & \text{F3} : \{\text{name(F3,mary)}\}
\end{bmatrix} \right\| \\
\textsc{order} & \text{fwd} \\
\textsc{feats} & \text{F2} \\
\textsc{sem} & \text{Sem2}
\end{bmatrix} \right\| \\
\textsc{feats} & \text{F2} \\
\textsc{sem} & \text{Sem2}
\end{bmatrix}
$$

This can combine by forward composition with a transitive verb, and we shall see an example of this below (4.35).

The above lexical rule (4.20) may also apply to the entry for the determiner *the* (4.4). The result is a sign looking for a noun to give a type-raised NP:

$$
(4.22)\;\begin{bmatrix}
\textsc{ortho} & \text{the} \\
\textsc{cat} & \text{s} \\
\textsc{slash} & \left\| \begin{bmatrix}
\textsc{cat} & \text{noun} \\
\textsc{slash} & \|\,\| \\
\textsc{order} & \text{fwd} \\
\textsc{feats} & \text{F} \\
\textsc{case} & \text{C} \\
\textsc{sem} & \text{X:Sem}
\end{bmatrix}
\begin{bmatrix}
\textsc{cat} & \text{s} \\
\textsc{slash} & \left\| \begin{bmatrix}
\textsc{cat} & \text{np} \\
\textsc{slash} & \|\,\| \\
\textsc{order} & \text{back} \\
\textsc{feats} & \text{F} \\
\textsc{case} & \text{subj} \\
\textsc{sem} & \text{X} : \{\text{definite(X)}\} \cup \text{Sem}
\end{bmatrix} \right\| \\
\textsc{order} & \text{fwd} \\
\textsc{feats} & \text{F2} \\
\textsc{sem} & \text{Sem2}
\end{bmatrix} \right\| \\
\textsc{feats} & \text{F2} \\
\textsc{sem} & \text{Sem2}
\end{bmatrix}
$$

It can therefore combine with a noun such as *book* (4.5) to yield the following type-raised NP:

$$
(4.23)\quad
\begin{bmatrix}
\textsc{ortho} & \text{the book} \\
\textsc{cat} & \text{s} \\
\textsc{slash} & \left\| \begin{bmatrix}
\textsc{cat} & \text{s} \\
\textsc{slash} & \left\| \begin{bmatrix}
\textsc{cat} & \text{np} \\
\textsc{slash} & \| \, \| \\
\textsc{order} & \text{back} \\
\textsc{feats} & \begin{bmatrix} \textsc{person} & 3 \\ \textsc{number} & \text{sg} \end{bmatrix} \\
\textsc{case} & \text{subj} \\
\textsc{sem} & \text{B} : \{\text{definite(B), book(B)}\}
\end{bmatrix} \right\| \\
\textsc{order} & \text{fwd} \\
\textsc{feats} & \text{F2} \\
\textsc{sem} & \text{Sem2}
\end{bmatrix} \right\| \\
\textsc{feats} & \text{F2} \\
\textsc{sem} & \text{Sem2}
\end{bmatrix}
$$

There is another occasion where [Steedman 87] advocates a slightly different version of NP type-raising in order to account for topicalisation, and we shall examine this in the next section.

The introduction of type-raising brings about the problem of spurious parses (that is, different parses yielding the same semantics), which is well-known in Categorial Grammar. [Steedman 90] argues that these different parses are interpretable in terms of different prosodic analysis (and therefore different information structures). Since we do not have an adequate representation of intonation and prosody here, we shall not address these issues.

4.7 Topicalisation and dislocation

Following the analysis in [Steedman 87], topicalised NPs arise from a **Topic type-raising** rule. A topicalised element in the sentence then becomes a functor over a sentence missing an NP, and is created by the following lexical rule, which also imposes some further constraints on its semantics (which are hinted at by saying that it is "focussed") as well as on its phonology if that is represented (as crudely suggested below). As with the Subject type-raising rule above (4.20), the rule applies to determiners and to lexical NPs:

NP – TopTRNP

$$\begin{bmatrix} \textsc{ortho} & \mathrm{O} \\ \textsc{cat} & \mathrm{np} \\ \textsc{slash} & \mathrm{S} \\ \textsc{feats} & \mathrm{F} \\ \textsc{case} & \mathrm{C} \\ \textsc{sem} & \mathrm{I1{:}Sem1} \end{bmatrix}$$

$$\longrightarrow \begin{bmatrix} \textsc{ortho} & \textsc{small caps}\ (\mathrm{O}) \\ \textsc{cat} & \mathrm{s} \\ \textsc{slash} & \mathrm{S} + \left\| \begin{bmatrix} \textsc{cat} & \mathrm{s} \\ \textsc{slash} & \left\| \begin{bmatrix} \textsc{cat} & \mathrm{np} \\ \textsc{slash} & \|\,\| \\ \textsc{order} & \mathrm{fwd} \\ \textsc{feats} & \mathrm{F} \\ \textsc{case} & \mathrm{C} \\ \textsc{sem} & \mathrm{I1{:}Sem1} \end{bmatrix} \right\| \\ \textsc{order} & \mathrm{fwd} \\ \textsc{feats} & \boxed{1} \\ \textsc{sem} & \mathrm{I2} : \mathrm{Sem2} \end{bmatrix} \right\| \\ \textsc{feats} & \boxed{1} \begin{bmatrix} \textsc{topic} & \mathrm{I1} \end{bmatrix} \\ \textsc{sem} & \mathrm{I2} : (\{\mathrm{focus(I1)}\} \cup \mathrm{Sem2}) \end{bmatrix} \tag{4.24}$$

This lexical rule produces, from an NP such as *beans*, the following one:

$$\begin{bmatrix} \textsc{ortho} & \textsc{beans} \\ \textsc{cat} & \mathrm{s} \\ \textsc{slash} & \left\| \begin{bmatrix} \textsc{cat} & \mathrm{s} \\ \textsc{slash} & \left\| \begin{bmatrix} \textsc{cat} & \mathrm{np} \\ \textsc{slash} & \|\,\| \\ \textsc{order} & \mathrm{fwd} \\ \textsc{case} & \mathrm{C} \\ \textsc{sem} & \mathrm{B} : \{\mathrm{beans(B)}\} \end{bmatrix} \right\| \\ \textsc{order} & \mathrm{fwd} \\ \textsc{feats} & \boxed{1} \\ \textsc{sem} & \mathrm{I2} : \mathrm{Sem2} \end{bmatrix} \right\| \\ \textsc{feats} & \boxed{1} \begin{bmatrix} \textsc{topic} & \mathrm{B} \end{bmatrix} \\ \textsc{sem} & \mathrm{I2} : (\{\mathrm{focus(B)}\} \cup \mathrm{Sem2}) \end{bmatrix} \tag{4.25}$$

This can combine with the following constituent, obtained by Subject type-raising *he*, and composing it with the verb:

$$
(4.26)\left[\begin{array}{ll}
\textsc{ortho} & \text{'he likes'} \\
\textsc{cat} & \text{s} \\
\textsc{slash} & \left\|\left[\begin{array}{ll} \textsc{cat} & \text{np} \\ \textsc{order} & \text{fwd} \\ \textsc{case} & \text{obj} \\ \textsc{sem} & \text{Y} : \text{Sem} \end{array}\right]\right\| \\
\textsc{feats} & \left[\textsc{fin}\ \ +\right] \\
\textsc{sem} & \text{S} : \left\{\begin{array}{l} \text{liking(S),role(S,agt,M3),} \\ \text{role(E,pat,Y)} \end{array}\right\} \cup \text{Sem}
\end{array}\right]
$$

The result of combining (4.25) with (4.26) (by function application) is the following:

$$
(4.27)\left[\begin{array}{ll}
\textsc{ortho} & \text{'}\textsc{beans}\text{ he likes'} \\
\textsc{cat} & \text{s} \\
\textsc{slash} & \|\,\| \\
\textsc{feats} & \left[\begin{array}{ll} \textsc{fin} & + \\ \textsc{topic} & \text{B} \end{array}\right] \\
\textsc{sem} & \text{S} : \left\{\begin{array}{l} \text{liking(S),role(S,agt,M3)} \\ \text{role(S,pat,B),focus(B),beans(B)} \end{array}\right\}
\end{array}\right]
$$

The fact that topicalisation is not clause-bounded can be accounted for by means of functional composition, to give sentences like BEANS *I believe he likes.*

Ungrammatical constructions like ** I believe* BEANS *he likes* can be ruled out by forcing the sentential complement of *believe* to have its FEATS:TOPIC feature specified as −, which will prevent the undesired unification.

Dislocation is similar to Spanish in that the dislocated element acts as a sentence modifier. It is not subcategorized for (and does not receive case), and may therefore co-occur with a pronoun. It is built from the standard NP and the entry for the comma, which is similar to the Spanish one (3.51).

$$(4.28)\left[\begin{array}{ll}\textsc{ortho} & \textbf{'that book,'} \\ \textsc{cat} & \mathrm{s} \\ \textsc{slash} & \left\| \left[\begin{array}{ll}\textsc{cat} & \mathrm{s} \\ \textsc{slash} & \|\,\| \\ \textsc{feats} & \mathrm{Feats} \\ \textsc{case} & \mathrm{Case} \\ \textsc{sem} & \mathrm{I : Sem}\end{array}\right] \right\| \\ \textsc{feats} & \mathrm{Feats} \\ \textsc{case} & \mathrm{Case} \\ \textsc{sem} & \mathrm{I : (\{role(I,R,B),\ book(B),\ focus(B)\} \cup Sem)}\end{array}\right]$$

This sign may modify in a very straightforward manner a complete sentence such as *Mary read it*:

$$(4.29)\left[\begin{array}{ll}\textsc{ortho} & \mathrm{'Mary\ read\ it'} \\ \textsc{cat} & \mathrm{s} \\ \textsc{sem} & \mathrm{E : (}\left\{\begin{array}{l}\mathrm{reading(E),role(E,pat,Y),} \\ \mathrm{role(E,agt,F),name(F,mary)}\end{array}\right\}\end{array}\right]$$

The result is:

$$(4.30)\left[\begin{array}{ll}\textsc{ortho} & \mathrm{'}\textbf{that book,}\mathrm{\ Mary\ read\ it'} \\ \textsc{cat} & \mathrm{s} \\ \textsc{sem} & \mathrm{E : (}\left\{\begin{array}{l}\mathrm{reading(E),role(E,pat,B),book(B),} \\ \mathrm{focus(B),role(E,agt,F),name(F,mary)}\end{array}\right\}\end{array}\right]$$

4.8 Clefts

It often occurs that the best translation equivalent to Spanish dislocated constructions is not an English dislocation, but a cleft, as shown in the example below:

(4.31) El libro, lo leyó María.
It was the book that Mary read.

It therefore seems appropriate to include, as part of the present English grammar fragment, some cleft constructions in this section. The whole process hinges on the cleft-making entry for *it was* (how this entry is built is irrelevant here).

$$(4.32)\ \begin{bmatrix} \textsc{ortho} & \text{'it was'} \\ \textsc{cat} & \text{s} \\ \textsc{slash} & \left\| \begin{array}{l} \boxed{1}\begin{bmatrix} \textsc{cat} & \text{np} \\ \textsc{order} & \text{fwd} \\ \textsc{sem} & \text{X:Sem1} \end{bmatrix} \\ \begin{bmatrix} \textsc{cat} & \text{s} \\ \textsc{slash} & \boxed{1} \\ \textsc{order} & \text{fwd} \\ \textsc{feats} & \begin{bmatrix} \textsc{compl} & \text{that} \end{bmatrix} \\ \textsc{sem} & \text{E:Sem} \end{bmatrix} \end{array} \right\| \\ \textsc{feats} & \boxed{2}\begin{bmatrix} \textsc{tense} & \text{past} \\ \textsc{fin} & + \end{bmatrix} \\ \textsc{sem} & \text{I} : (\text{Sem} \cup \text{Sem1} \cup \{\text{focus(X)}\}) \end{bmatrix}$$

The entry for *the book* has been seen already. This is the kind of constituent that (4.32) is looking for first.

Upon combination the result is:

$$(4.33)\ \begin{bmatrix} \textsc{ortho} & \text{'it was the book'} \\ \textsc{cat} & \text{s} \\ \textsc{slash} & \left\| \begin{bmatrix} \textsc{cat} & \text{s} \\ \textsc{slash} & \left\| \begin{bmatrix} \textsc{cat} & \text{np} \\ \textsc{order} & \text{fwd} \\ \textsc{sem} & \text{X:Sem1} \end{bmatrix} \right\| \\ \textsc{order} & \text{fwd} \\ \textsc{feats} & \\ \textsc{sem} & \text{E:Sem} \end{bmatrix} \right\| \\ \textsc{feats} & \boxed{2}\begin{bmatrix} \textsc{tense} & \text{past} \\ \textsc{fin} & + \end{bmatrix} \\ \textsc{sem} & \text{I} : (\text{Sem} \cup \{\text{focus(B), book(B), definite(B)}\}) \end{bmatrix}$$

This can then combine with an entry for *that Mary read*, which is built from the following entry for a cleft-making "that":

$$(4.34)\ \begin{bmatrix} \textsc{ortho} & \text{that} \\ \textsc{cat} & \text{s} \\ \textsc{slash} & \left\| \begin{array}{l} \begin{bmatrix} \textsc{cat} & \text{s} \\ \textsc{slash} & \left\| \boxed{1} \right\| \\ \textsc{sem} & \text{E : Sem2} \end{bmatrix} \\ \boxed{1} \begin{bmatrix} \textsc{cat} & \text{np} \\ \textsc{order} & \text{fwd} \\ \textsc{case} & \text{obj} \\ \textsc{sem} & \text{Y : Sem1} \end{bmatrix} \end{array} \right\| \\ \textsc{feats} & \begin{bmatrix} \textsc{compl} & \text{that} \end{bmatrix} \\ \textsc{sem} & \text{E : Sem1} \cup \text{Sem2} \end{bmatrix}$$

This subcategorizes first for a sentence missing its object, such as the following, obtained by combining the subject type-raised NP *Mary* (4.21) with the transitive verb *read* (4.7):

$$(4.35)\ \begin{bmatrix} \textsc{ortho} & \text{'Mary read'} \\ \textsc{cat} & \text{s} \\ \textsc{slash} & \left\| \begin{bmatrix} \textsc{cat} & \text{np} \\ \textsc{order} & \text{fwd} \\ \textsc{case} & \text{obj} \\ \textsc{sem} & \text{Y : Sem} \end{bmatrix} \right\| \\ \textsc{feats} & \begin{bmatrix} \textsc{fin} & + \end{bmatrix} \\ \textsc{sem} & \text{E : } \left\{ \begin{array}{l} \text{reading(E),role(E,agt,F3),name(F3,mary),} \\ \text{role(E,pat,Y)} \end{array} \right\} \cup \text{Sem} \end{bmatrix}$$

The result of the combination is the following:

$$(4.36)\ \begin{bmatrix} \textsc{ortho} & \text{'that Mary read'} \\ \textsc{cat} & \text{s\_that} \\ \textsc{slash} & \left\| \begin{bmatrix} \textsc{cat} & \text{np} \\ \textsc{order} & \text{back} \\ \textsc{case} & \text{obj} \\ \textsc{sem} & \text{Y : Sem} \end{bmatrix} \right\| \\ \textsc{feats} & \begin{bmatrix} \textsc{compl} & \text{that} \end{bmatrix} \\ \textsc{sem} & \text{E : } \left\{ \begin{array}{l} \text{reading(E),role(E,agt,F3),name(F3,mary),} \\ \text{role(E,pat,Y)} \end{array} \right\} \cup \text{Sem} \end{bmatrix}$$

The result will be

(4.37)
$$\begin{bmatrix} \textsc{ortho} & \text{'it was the book that Mary read'} \\ \textsc{cat} & \text{s} \\ \textsc{sem} & \text{E} : \left\{ \begin{array}{l} \text{reading(E),role(E,agt,F3),name(F3,mary),} \\ \text{role(E,pat,B),book(B),definite(B),focus(B)} \end{array} \right\} \end{bmatrix}$$

4.9 Conclusions

I hope to have shown here how the ideas developed in the previous chapter about Spanish can be put to good use in an English grammar as well. The case mechanism that was introduced for Spanish is less essential in an English grammar, but I believe it makes the lexical entries clearer and can therefore make the grammar-writer's task easier. Most of the English constructions presented correspond to rather similar ones (though not necessarily isomorphic) in Spanish, and this similarity will be used in the next chapters in developing a MT approach.

Chapter 5

The bilingual lexicon

5.1 Introduction

Having presented in the last two chapters monolingual UCG grammars for Spanish and English, I shall attempt here to put them into correspondence in a way that is suitable for doing MT in a lexicalist framework. This is explained in this chapter and the next one. The design and implementation issues involved in such an MT system are left until the next chapter. In this one, I will show how bilingual lexical entries may be represented in a way that is independent of the translation procedure to be used.

The main concern will be to describe how the monolingual signs are related. An important contention of lexically-based MT is that the information required for translation lies in these lexical entries. I would also like to claim that, from a practical point of view as well as for the sake of elegance and readability of the grammar, the bilingual entries should be as similar as possible to the kind of information that we may expect to find in a "real" bilingual dictionary. That is, the bilingual lexicon tells us which lexical entries correspond to which, possibly adding some constraints as to when the correspondence applies, and *nothing else*. How complex expressions are built from these lexical signs is wholly determined by the (monolingual) grammar of the target language.

This chapter presents several approaches to implementing MT using the two monolingual lexicons explained in the last three chapters, and shows how they may be put into correspondence in the form of a bilingual lexicon. Many of the ideas here were developed with Pete Whitelock [Whitelock 91] and Mike Reape [Reape 90, forthcoming], who used the term *Shake-and-Bake* to describe it, by analogy with the nickname given to a paper by Ewan Klein and Ivan Sag [Klein & Sag 85]. Whitelock and Reape's work concentrates mainly on the motivation and the theory behind *Shake-and-Bake*, and I believe the work reported here is the first implementation of these ideas. Of course, any responsibility for misrepresenting what they think the approach involves lies entirely with myself.

In the course of the implementation, many details that were not discussed in their work had to be considered in great detail, and this revealed some interesting problems and solutions, in particular as far as the aspect of generation is concerned.

5.2 Why Shake-and-Bake

In Chapter 1, we outlined the main problems that exist with the most common approaches to MT (transfer-based and interlingual), namely lack of portability and modularity, and it was argued that a lexical-based approach makes it possible to write the monolingual grammars quite independently from each other, and to put the information specific to the language pair under consideration in the place where it belongs, namely the bilingual lexicon.

Lexically-based translation also allows us to put into practice some of the main advances in Computational Linguistics of the last decade, namely the use of unification as the main operator for combining complex signs, which may contain constraints at various levels (syntactic, semantic and phonological).

5.3 Simple bilingual signs

Let us start with a very simple example: a transitive verb with a straightforward translation, as described in the previous two chapters.

We shall concentrate only on the part of the sign that is relevant for the translation. For this purpose, we will introduce a more concise notation to picture the signs without cluttering them with their unessential aspects. It is important to bear in mind that this will only be a shorthand notation. The full signs which this notation represents may occasionally appear in the main text.

Following the notation used in [Zeevat et al. 87], signs will appear as follows:

$$(5.1)\quad \begin{bmatrix} \textsc{Orthography} \\ \textsc{Syntax} \\ \textsc{Semantics} \end{bmatrix}$$

Alternatively, they may be represented as:

[Orthography:Syntax:Semantics]

The value of the syntax will either be a simple category, or one of the form A/B, where A is a simple category and B is an active sign (which will have uninstantiated Orthography). The slash will usually be non-directional. Complex feature structures will usually be abridged and the essential aspects of them will be shown in the text.

The value of the semantics be of the form **Index : Formula**. Index represents the entity that the expression refers to. It may appear, together with others, in the *Formula*. *Formula* is conjunction of predicates that will involve the Index. Under the usual Prolog notation, constants in the *Formula* will have names starting with lower-case letters, and variables will start with upper-case letters. The variables and constants are actually typed, and this will be hinted at by the choice of letters. For instance, the index E will usually be taken to denote an event, X3, Y3 and Z3 to denote third person entities, and so on. Since for these expository purposes nothing crucial depends on this typing, no more will be said about the subject in this chapter. It will be discussed in the next one, together with the other implementational details.

In this new notation, the monolingual sign for a Spanish transitive verb, subcategorizing for a type-raised NP as discussed in Chapter 3, may look like the following :

$$(5.2)\quad \left[\begin{array}{l} \text{leyó} \\ \text{s}/\left[\begin{array}{l} \text{s}/\left[\begin{array}{l}\text{s}\\ \text{E} : \{\text{leer(E),role(E,agt,X),role(E,pat,Y)}\}\end{array}\right] \\ \text{Sem}\end{array}\right] \\ \text{Sem}\end{array}\right]$$

Since the bilingual entries will make use of the semantic indices of this verb and its arguments, it is convenient to make the other indices of the Formula appear at the top level of the sign's semantics. They will be under the feature labels *arg0*, *arg1*, *arg2*, and so on, with the convention that *arg0* will be the semantic index of the entity for which the word itself stands, and the rest will be the various "arguments". The order in which they appear is arbitrary, and different orders could in principle be used for different words, but then the bilingual grammar writer would have to consult these monolingual entries all the time. To ease that task, it is helpful adopt some convention about the order in which these arguments appear, and to keep it consistent throughout the grammar. Some (arbitrary) notion of obliqueness will be used to order them here.

It is important to stress that the appearance of these ARG features does nothing to change the monolingual sign. All that has been done is provide some "handles" to act as "syntactic sugar" in making the bilingual lexical entries easier to write and to talk about.

Thus, the above sign may look alternatively like the following:

$$(5.3)\quad \left[\begin{array}{ll} \text{ORTHO} & \text{leyó} \\ \text{CAT} & \text{s}/\left[\begin{array}{l} \text{s}/\left[\begin{array}{l}\text{s}\\ \text{E} : \{\text{leer(E),role(E,agt,X),role(E,pat,Y)}\}\end{array}\right] \\ \text{Sem}\end{array}\right] \\ \text{SEM} & \text{Sem} \\ \text{ARG0} & \text{E} \\ \text{ARG1} & \text{X} \\ \text{ARG2} & \text{Y}\end{array}\right]$$

Both notations will be used together in the text, and since they are quite simple, it is hoped that no confusion will result from this. As far as the implementation is concerned, the full versions of the signs, which are as presented in Chapters 3 and 4, together with the ARG features mentioned above, are used.

The corresponding monolingual sign for an English transitive verb looks very much the same, the main difference being that it subcategorizes for its subject as well as for its object, and both of these have category NP. Again, we only show the parts that are relevant for the translation, using the notation above:

$$(5.4)\quad \left[\begin{array}{l} \text{read} \\ \text{s/} \left\| \begin{array}{l} \left[\begin{array}{l}\text{np}\\ \text{Y:Sem2}\end{array}\right] \\ \left[\begin{array}{l}\text{np}\\ \text{X:Sem1}\end{array}\right] \end{array} \right\| \\ \text{E:} \left\{ \begin{array}{l} \text{reading(E), role(E,agt,X),} \\ \text{role(E,pat,Y)} \end{array} \right\} \cup \text{Sem1} \cup \text{Sem2} \end{array}\right]$$

The alternative notation, with the indices made explicit, is as follows:

$$(5.5)\quad \left[\begin{array}{ll} \textsc{ortho} & \text{read} \\ \textsc{cat} & \text{s/} \left\| \begin{array}{l} \left[\begin{array}{l}\text{np}\\ \text{Y:Sem2}\end{array}\right] \\ \left[\begin{array}{l}\text{np}\\ \text{X:Sem1}\end{array}\right] \end{array} \right\| \\ \textsc{sem} & \text{E:} \left\{ \begin{array}{l} \text{reading(E), role(E,agt,X),} \\ \text{role(E,pat,Y)} \end{array} \right\} \cup \text{Sem1} \cup \text{Sem2} \\ \textsc{arg0} & \text{E} \\ \textsc{arg1} & \text{X} \\ \textsc{arg2} & \text{Y} \end{array}\right]$$

Having seen these monolingual signs, the bilingual entry is then very predictable: it has two features, called SPANISH and ENGLISH, and the values of these are the monolingual signs. The indices in these signs will typically be identified (or at least some of them will be). The following gives a picture of this:

$$(5.6)\quad \left[\begin{array}{ll} \textsc{spanish} & \boxed{5.3} \left[\begin{array}{ll} \textsc{arg0} & \text{E} \\ \textsc{arg1} & \text{X} \\ \textsc{arg2} & \text{Y} \end{array}\right] \\ \textsc{english} & \boxed{5.5} \left[\begin{array}{ll} \textsc{arg0} & \text{E} \\ \textsc{arg1} & \text{X} \\ \textsc{arg2} & \text{Y} \end{array}\right] \end{array}\right]$$

This bilingual entry simply sets up a correspondence between the (5.3) and (5.5), and co-indexes the events and participants in both signs.

It is the simplest form of bilingual entry that there can be, since the equations in it only involve semantic indices. In the more general case, there is nothing to prevent us putting constraints on other aspects of either monolingual sign, such as the phonology, the semantic types of the arguments, and so on.

Thus, if we had a more adequate representation of phonology and information structure, it would be possible to state in a bilingual sign that a monolingual sign carrying a certain information structure in one language corresponds to a certain combination of signs in the other language. This could be used, for instance, in order to state correspondences between dislocated NPs in Spanish and cleft constructions in English, as outlined in Section 7.5.

5.4 Outline of Shake-and-Bake translation

Shake-and-Bake will be described at a greater length in the next chapter, where I will give further details about implementational issues that arise. For the time being, it may be sketched as the following algorithm:

1. Parse the source language sentence using the monolingual Source Language lexicon and grammar only.

2. Each successful parse will further instantiate the lexical Source Language (SL) signs that were obtained from the lexicon by the parser. For instance, variables filling roles in the semantics will become instantiated with the indices of the fillers of these roles, which may be atoms, or, in this case, "typed" terms. This was shown with the monolingual grammars.

3. For each successful parse, carry out the following procedure:

    (a) Ignore the parse tree, keeping only the lexical entries, which have become further instantiated during the parse.

(b) Look up these entries in the bilingual lexicon to get the Target Language (TL) entries. In particular, this has the effect of copying across the relevant instantiations of the semantic indices as specified by the bilingual lexicon.

(c) We then end up with a *bag* of TL words, which are more instantiated than the entries in the TL lexicon, since they arise from the SL look-up: they are just like the TL words in the monolingual lexicon, but the variables in the semantic indices have been copied over from the SL sign, as prescribed by the bilingual lexicon.

(d) Generate from this bag of words, letting the lexical entries and the TL grammar drive the word order. Each of these strings generated corresponds to a translation of the specific SL parse under consideration. This process of generation from a bag of words using the grammar to instantiate the order is the part known as Shake-and-Bake proper. However, from here on we shall use that term for the whole algorithm described here, and the generation will be known as **baking**.

4. When all the successful parses of the SL have been processed, all possible translations have been found.

I shall comment on these points in greater detail in the next chapter.

5.5 Bilingual lookup

Having seen the monolingual entries for a simple transitive verb, and how these are put into correspondence by means of the bilingual lexicon, we shall now have a more detailed look at how the bilingual lookup works.

Recall that the parsing process gave us a bag of SL words which were more instantiated than the original lexical entries because various features had become unified during the parsing process. In particular, the semantics have become highly instantiated by the cross-referencing of indices. For each possible parse, these words are looked up in the monolingual SL lexicon to get some kind of "key" through which they will be referenced in the bilingual lexicon (this key may be simply the orthography, or the orthography together with some reference for the particular word-sense involved, in which case it could be another feature of the sign). The bilingual lexicon then finds a key for a corresponding TL word, which is then looked up in the monolingual TL lexicon. In addition to that, the bilingual lexicon imposes some further constraints on the TL word that has been looked up: the semantic indices will be cross-referenced, and possibly some further restrictions can be put at this stage. This process of looking up the words in the SL bag may yield more than one TL bag of words, since some of the words in the SL may have several possible translations.

The more detailed technical issues that arise will be left until the next chapter, but to summarise this initial explanation, for each SL bag of words we end up with one or more TL bag of words, from which we will have to generate. Each string successfully generated from these will produce a possible translation.

5.6 Sample translation

We now have all the necessary tools to follow through a very simple example of the sentence *María leyó el libro* to *Mary read the book*.

Let us assume the entries for the verb that appear earlier in this chapter, together with monolingual entries for the other words as they appear in Chapter 3 (for Spanish) and Chapter 4 (for English). They are reproduced below for the sake of convenience, and the shorthand notation introduced earlier on in this chapter is used for greater expository convenience, so it should be borne in mind that these are somewhat cut-down versions of the full signs. In particular, the case-assignment mechanism has been left out. Recall that its role was to match the various "arguments" that a verb, for instance, may have, with the indices appearing in the semantics.

5.6.1 Spanish monolingual entries

The Spanish signs look as follows, with the abbreviated notation used when necessary for the sake of clarity:

$$(5.7)\quad \begin{bmatrix} \textsc{ortho} & \text{'María'} \\ \textsc{cat} & \text{s/} \left\| \begin{bmatrix} \text{s} \\ \text{I1 : Sem1} \end{bmatrix} \right\| \\ \textsc{sem} & \text{I1 : }(\{\text{role(I1,\_R1,F3), name(F3,maria}\} \cup \text{Sem1}) \\ \textsc{arg0} & \text{F3} \end{bmatrix}$$

$$(5.8)\quad \begin{bmatrix} \textsc{ortho} & \text{el} \\ \textsc{cat} & \text{s/} \left\| \begin{matrix} \begin{bmatrix} \text{n} \\ \text{I1:Sem1} \end{bmatrix} \\ \begin{bmatrix} \text{s} \\ \text{I2 : Sem2} \end{bmatrix} \end{matrix} \right\| \\ \textsc{sem} & \text{I2 : }(\{\text{definite(I1)}\} \cup \text{Sem1} \cup \text{Sem2}) \\ \textsc{arg0} & \text{I1} \end{bmatrix}$$

$$(5.9)\quad \begin{bmatrix} \textsc{ortho} & \text{libro} \\ \textsc{cat} & \text{n} \\ \textsc{sem} & \text{L : }\{\text{libro(L)}\} \\ \textsc{arg0} & \text{L} \end{bmatrix}$$

5.6.2 English monolingual entries

The English entries corresponding to the above are:

$$(5.10)\quad \begin{bmatrix} \textsc{ortho} & \text{'Mary'} \\ \textsc{cat} & \text{np} \\ \textsc{sem} & \text{F3 : }\{\text{name(F3,mary)}\} \\ \textsc{arg0} & \text{F3} \end{bmatrix}$$

$$(5.11)\quad \begin{bmatrix} \textsc{ortho} & \text{the} \\ \textsc{cat} & \text{np/} \left\| \begin{bmatrix} \text{n} \\ \text{I : Sem} \end{bmatrix} \right\| \\ \textsc{sem} & \text{I : }(\{\text{definite(I1)}\} \cup \text{Sem1}) \\ \textsc{arg0} & \text{I1} \end{bmatrix}$$

$$(5.12)\quad \begin{bmatrix} \textsc{ortho} & \text{book} \\ \textsc{cat} & \text{n} \\ \textsc{sem} & \text{B : }\{\text{book(L)}\} \\ \textsc{arg0} & \text{B} \end{bmatrix}$$

5.6.3 Bilingual entries

The bilingual entries corresponding to these are as follows:

$$(5.13)\ \text{Sign for } \textit{María-Mary} \ : \left[\begin{array}{ll} \textsc{spanish} & \boxed{5.7}\left[\textsc{sem}\ \left[\textsc{arg}0\ \ \text{F3}\right]\right] \\ \textsc{english} & \boxed{5.10}\left[\textsc{sem}\ \left[\textsc{arg}0\ \ \text{F3}\right]\right] \end{array}\right]$$

$$(5.14)\ \text{Sign for } \textit{el-the} \ : \left[\begin{array}{ll} \textsc{spanish} & \boxed{5.8}\left[\textsc{sem}\ \left[\textsc{arg}0\ \ \text{D}\right]\right] \\ \textsc{english} & \boxed{5.11}\left[\textsc{sem}\ \left[\textsc{arg}0\ \ \text{D}\right]\right] \end{array}\right]$$

$$(5.15)\ \text{Sign for } \textit{libro-book} \ : \left[\begin{array}{ll} \textsc{spanish} & \boxed{5.9}\left[\textsc{sem}\ \left[\textsc{arg}0\ \ \text{B}\right]\right] \\ \textsc{english} & \boxed{5.12}\left[\textsc{sem}\ \left[\textsc{arg}0\ \ \text{B}\right]\right] \end{array}\right]$$

The bilingual entry for the verb is given in (5.6)).

5.6.4 Parsing the source language

Let us now see how the SL sentence is parsed (assuming that was the Spanish example). In the discussion below, only partial pictures of the signs will be shown, for the sake of clarity. In particular, the ARG features are left out, since their role is limited to the bilingual lookup process. The signs for *el* (5.8) and *libro* (5.9) combine together to produce the following:

$$(5.16)\left[\begin{array}{ll} \textsc{ortho} & \text{'el libro'} \\ \textsc{cat} & \text{s}/\left\| \begin{bmatrix} \text{s} \\ \text{I2 : Sem2} \end{bmatrix} \right\| \\ \textsc{sem} & \text{I2 : }(\{\text{definite(L), libro(L)}\} \cup \text{Sem2}) \end{array}\right]$$

This now combines with the verb *leyó* (5.3) to produce:

$$(5.17)\left[\begin{array}{ll} \textsc{ortho} & \text{'leyó el libro'} \\ \textsc{cat} & \text{s} \\ \textsc{sem} & \text{E : }\left\{\begin{array}{l} \text{leer(E), role(E,agt,X),} \\ \text{role(E,pat,L), definite(L),} \\ \text{libro(L)} \end{array}\right\} \end{array}\right]$$

This may finally be modified by its subject *María* (5.7), to give:

$$(5.18)\begin{bmatrix} \textsc{ortho} & \text{'María leyó el libro'} \\ \textsc{cat} & \text{s} \\ \textsc{sem} & \text{E} : \left\{ \begin{array}{l} \text{leer(E), role(E,agt,F3),} \\ \text{role(E,pat,L), definite(L),} \\ \text{libro(L), name(F3,maria)} \end{array} \right\} \end{bmatrix}$$

This derivation should be familiar by now.

5.6.5 Bilingual lookup

Upon parsing such a SL sentence as the previous Spanish one, *María leyó el libro*), the semantics of the above signs will get further instantiated as the various indices unify. After the successful parse of the SL, and the bilingual lookup, the TL signs which will be used for generation will be the same as the ones above, only with more specific semantics. The SL words that result from the previous parse, and which will be used for the lookup are just like the lexical entries (5.3, 5.7, 5.8 and 5.9), but somewhat more instantiated:

$$(5.19)\begin{bmatrix} \textsc{ortho} & \text{'María'} \\ \textsc{cat} & \text{s}/ \left\| \begin{bmatrix} \text{s} \\ \text{E} : \boxed{1} \end{bmatrix} \right\| \\ \textsc{sem} & \text{E} : (\{\text{role(E,agt,F3), name(F3,maria)}\} \cup \boxed{1}) \\ \textsc{arg0} & \text{F3} \end{bmatrix}$$

$$(5.20)\begin{bmatrix} \textsc{ortho} & \text{leyó} \\ \textsc{cat} & \text{s}/ \begin{bmatrix} \text{s}/ \begin{bmatrix} \text{s} \\ \boxed{2}\ \text{E} : \left\{ \begin{array}{l} \text{leer(E),role(E,agt,X),} \\ \text{role(E,pat,Y)} \end{array} \right\} \end{bmatrix} \\ \text{E} : \boxed{1} \end{bmatrix} \\ \textsc{sem} & \text{E} : \boxed{1} \\ \textsc{arg0} & \text{E} \\ \textsc{arg1} & \text{X} \\ \textsc{arg2} & \text{Y} \end{bmatrix}$$

$$
(5.21)\begin{bmatrix}
\textsc{ortho} & \text{el} \\
\textsc{cat} & \text{s}/\left\| \begin{array}{l} \begin{bmatrix} \text{n} \\ \text{L} : \{\text{libro(L)}\} \end{bmatrix} \\ \begin{bmatrix} \text{s} \\ \boxed{2} \end{bmatrix} \end{array} \right\| \\
\textsc{sem} & \text{E} : \boxed{1} \left\{ \begin{array}{l} \text{definite(L), libro(L), leer(E),} \\ \text{role(E,agt,X), role(E,pat,L)} \end{array} \right\} \\
\textsc{arg}0 & \text{L}
\end{bmatrix}
$$

$$
(5.22)\begin{bmatrix}
\textsc{ortho} & \text{libro} \\
\textsc{cat} & \text{n} \\
\textsc{sem} & \text{L} : \{\text{libro(L)}\} \\
\textsc{arg}0 & \text{L}
\end{bmatrix}
$$

Note that in the above signs, the reentrancy indices $\boxed{1}$ and $\boxed{2}$ are global to the four words.

The lookup process gets the lexical entries for the TL signs, and copies over the indices in the semantics (these are readily accessible via the ARG "handles"). Note that only the indices are copied, and the predicate names used in the semantics may be (and in this case, are) purely monolingual. The resultant TL signs are the following (new instantiations are indicated by boldface):

$$
(5.23)\begin{bmatrix}
\textsc{ortho} & \text{'Mary'} \\
\textsc{cat} & \text{np} \\
\textsc{sem} & \mathbf{F3} : (\{\text{name(}\mathbf{F3}\text{,mary)}\} \\
\textsc{arg}0 & \mathbf{F3}
\end{bmatrix}
$$

$$
(5.24)\begin{bmatrix}
\textsc{ortho} & \text{read} \\
\textsc{cat} & \text{s}/\left\| \begin{array}{l} \begin{bmatrix} \text{np} \\ \mathbf{L}\text{:Sem2} \end{bmatrix} \\ \begin{bmatrix} \text{np} \\ \mathbf{F3}\text{:Sem1} \end{bmatrix} \end{array} \right\| \\
\textsc{sem} & \text{E:} \left\{ \begin{array}{l} \text{reading(}\mathbf{E}\text{), role(}\mathbf{E}\text{,agt,}\mathbf{F3}\text{),} \\ \text{role(}\mathbf{E}\text{,pat,}\mathbf{L}\text{)} \end{array} \right\} \cup \text{Sem1} \cup \text{Sem2} \\
\textsc{arg}0 & \mathbf{E} \\
\textsc{arg}1 & \mathbf{F3} \\
\textsc{arg}2 & \mathbf{L}
\end{bmatrix}
$$

$$(5.25)\begin{bmatrix} \text{ORTHO} & \text{the} \\ \text{CAT} & \text{np}/\left\| \begin{bmatrix} \text{n} \\ \mathbf{L}\text{:Sem1} \end{bmatrix} \right\| \\ \text{SEM} & \mathbf{L}\text{:}(\{\text{definite}(\mathbf{L})\} \cup \text{Sem1}) \\ \text{ARG0} & \mathbf{L} \end{bmatrix}$$

$$(5.26)\begin{bmatrix} \text{ORTHO} & \text{book} \\ \text{CAT} & \text{n} \\ \text{SEM} & \mathbf{L}\text{:}\{\text{book}(\mathbf{L})\} \\ \text{ARG0} & \mathbf{L} \end{bmatrix}$$

The main thing to notice in the above signs is that in the semantics of *read* (5.24), it is specified that the agent must have index F3 (which is the index of *Mary* (5.23)), and the patient must have index L (which is the index of *the book*).

5.6.6 Parsing as generation: "baking"

A close inspection of these signs shows that only one possible permutation of them is licensed by the grammar. In particular, the incorrect translation *The book read Mary* is ruled out by the semantic restriction that *Mary* must be the agent and *the book* the patient. In this specific example, it happens that the typing restrictions on the semantic variables only allows one translation, but of course we cannot rely on that in the general case. It is possible however that the variables L and F3 could be wrongly identified at this stage. The solution to this is to *Skolemise* the variables in the semantics before bilingual lookup, a point which is explained in the next chapter (Section 6.2.3). The picture above is therefore slightly inaccurate in that all the variables in the semantics are actually grounded.

Generating from these signs is actually very similar to parsing. In fact, one way of thinking about the generation is to consider each possible permutation of these words and try to parse it, using either a left-corner approach or a CKY parser, or any other suitable method.

In this sense therefore, generating from a bag of words is very different from what is normally meant by the term *generation*, and the expression **baking** will be used from here on.

Two approaches to baking will be presented in the next chapter. They will involve some simple modifications to the parsing algorithms presented, in order to allow the combinatorial properties of the grammar to dictate the linear ordering of the words. Nevertheless, they will still be equivalent to trying to parse all possible permutations of the bag of TL words.

5.7 Concluding remarks

This chapter was intended to provide an overview of the general Shake-and-Bake approach. Of course, the particular example chosen was trivial to translate, since a word-for-word translation suffices for this sentence. It illustrates however the main point of the approach, by showing how generation takes place from the bag of TL words, and imposes a particular word ordering on them, namely the one consistent with the syntax of the TL and the semantics of the SL expression.

Many questions have been left unanswered (sometimes even unasked!), and I shall attempt to address some of them, as well as offering less trivial examples, in the next chapter.

Chapter 6

Shake-and-Bake translation

The previous chapter presented a brief overview of the Shake-and-Bake approach, describing the various phases involved. In this chapter and the next one, we will go into greater detail of the problems which may arise at each of these stages. We shall also see some specific examples of monolingual and bilingual entries to solve some linguistically interesting examples.

6.1 Outline of parser and generator (baker)

One of the advantages of this Shake-and-Bake approach is that parsing and baking use exactly the same monolingual lexical entries. The grammar rules are very few in number, as is standard with Categorial Grammars, and are almost identical for the two languages. The differences between the grammar rules used for parsing and baking have to do with the fact that expressions which are translations of each other need not have the same number of words. The solution to this involves modifying the straightforward lookup procedure, and we shall come back to this point later.

6.1.1 Parsing

The grammars presented in Chapters 3 and 4 use a fairly ordinary unification-based formalism. Any of the commonly available and well-known parsing strategies for such grammars will do fine. In particular, graph unification is implemented here by representing graphs as variable-tailed lists (as suggested in [Gazdar and Mellish 90] amongst many others), and the unification itself is built "on top" of standard Prolog term unification.

The most naive parser presented here uses Prolog's built-in control strategy directly to drive a DCG shift-reduce parser. Such a parsing strategy is very easy to write, but also a fairly inefficient one because of the large amount of work that gets wasted with backtracking.

The algorithm uses a simple stack to store intermediate phrases. For each word processed in the input string, it can either be *shifted* (i.e. put on the stack), or *reduced* (i.e. combined with the topmost entry in the stack using a grammar rule). If it is reduced, the existing entry on the stack with which the input was combined is removed from the stack, and the result is again either shifted or reduced.

The whole process finishes when there are no more words in the input string to process, and it does so successfully if the only entry left on the stack is a terminal symbol.

A more efficient solution is to use a chart-based parser. Since categorial grammars are binary-branching, some of the most popular parsers for them are based on the CKY algorithm [Pereira & Shieber 87], which is the algorithm used here.

The CKY algorithm can be outlined as follows. Suppose we are to parse a sentence of length n using a binary-branching grammar. The basic idea is to use a table of well-formed substrings, or chart, in which smaller substrings are always added before the larger ones. Figure 6.1 shows the steps involved.

If the CKY chart parser is used, it is important, when these entries are combined, not to alter the edges corresponding to the original lexical entries by instantiating variables in them, since by the very nature of chart parsing, these entries may be used to build more than one edge. So when new edges are added to the chart, these edges must have recorded in them the versions of their daughter edges with all the right instantiations. In other words, care should be taken when building new edges not to unify destructively parts of existing ones (since this unification may cause further parses to be missed), but merely to check whether the unification can be made, and record the result of the unification at the resulting edge.

1. Initialise the chart by adding in all the words (well-formed substrings of length 1). The chart may be represented by a graph consisting of $n + 1$ nodes labelled 0 to n, and the well-formed substrings represented by labelled arcs in this graph. Thus the rth word in the chart will be an arc between nodes $r - 1$ and r.

2. For r between 2 and n, add all the well-formed substrings of length r. Since the grammar is binary branching, this is done as follows:

   (a) For i between 0 and $n - r$, find all new strings of length r which start at position i, by trying to combine two existing arcs, as follows:

      i. For j between 1 and $r - 1$

         A. For each pair of arcs between i and $i + j$, and $i + j$ and $i + r$, find all grammar rules that allow these arcs to combine into a longer one from i to $i + r$

         B. For each of these rules add the new arc to the chart.

         C. End

      ii. End

   (b) End

3. Any arc from 0 to n represents a parse of the string.

4. End

Figure 6.1: The basic CKY parsing algorithm.

When the chart is complete, the edges which offer complete parses of the sentence therefore include copies of the lexical entries with all their semantics instantiated. This is rather wasteful of memory and could no doubt be improved upon (for instance, by having some kind of structure sharing).

For each parse, we then gather the instantiated lexical entries, ignoring the parse tree and the linear order. In other words, we end up with a bag (represented as a list) of lexical entries with instantiated semantics. It is this bag that is used for generation. The combinatoric properties of the signs in this bag and the TL grammar, together with the fact that the semantics are instantiated, allow only the correct translations to be generated from here. Two algorithms are suggested for the generation, and they will be discussed in a later Subsection (6.1.3 and 6.1.4).

6.1.2 Bilingual lookup

Bilingual lookup is carried out by a predicate called `lookup`, which searches for the SL entry in the SL lexicon (to find the index of the correct homograph) in order to index it in the bilingual lexicon, then uses the bilingual lexicon to find its equivalent in the TL lexicon, and adding all the extra constraints that may be found in the bilingual lexicon. These constraints will identify the various semantic arguments appearing in the two signs, and may add further monolingual constraints (none of the signs in the current system do).

The only thing that the semantics of the SL and TL are required to have in common are the indices involved. In particular, the names of the predicates, and even the role names, need not be the same. In this way, the whole system is kept completely modular, and the monolingual grammars can be re-used for many language pairs. This is further discussed in Subsection 6.2.1. The only module that needs to have access to both semantics is the bilingual lexicon, and it is often necessary to refer to the full semantics there. For instance, we may wish to say that the English word *leg* corresponds to the Spanish *pierna* if it has a human owner, and to *pata* if it has a non-human owner.

The `lookup` predicate first calls the `lex` predicate, which takes care of looking up the words in the monolingual SL lexicon. `lex` takes a lexical entry from the bag of SL words produced by the parse, and finds the key for that lexical entry which is used in the bilingual lexicon (in this case, the key is just the monolingual phonology).

The lexical sign, together with the key, are then passed to the predicate `lex_bi`, which carries out the bilingual lookup using that key, and creates a TL sign with the constraints imposed by the bilingual lexicon entry (so it is a very basic sign, typically with very little syntactic information, since the bilingual lexicon is concerned mainly with stating correspondences between semantic indices).

Finally, that sign is used by the `lex` predicate again, to find a lexical entry to put into the bag. What is put in the bag is the unification of the entry appearing in the lexicon with the preliminary sign built by `lex_bi` containing the bindings of the semantic indices.

This is the basic picture for the cases when there is a 1-1 correspondence between a source and a target word.

In cases when there is not such a correspondence, in other words, when a word in SL corresponds to nothing, or to several words in the TL, the bilingual lookup procedure is slightly modified from what was described above, so that the bilingual lexicon, instead of containing pairs of words (one in SL, the other in TL), contains correspondences between *bags* of words (which may be empty, or contain several words, though in most cases these bags will contain one word each, and where the words may have some linear ordering constraints). These bags can then be appended together to build the bag of TL words which will be used for baking.

6.1.3 Naive baking

The most straightforward way to bake the TL bag of words is a generalisation of the Shift-Reduce algorithm, and is due to Pete Whitelock and Mike Reape and described in Evelyn van de Veen's MSc thesis [van de Veen 90], where it is used for parsing discontinuous constituents. See also [Whitelock 91]. The difference between this and standard shift-reduce is that the entry currently being processed does not have to be reduced with the topmost entry on the stack: instead, it can be reduced with anything in the stack.

This is equivalent to trying out all possible permutations of the words and trying to parse them using an ordinary Shift-Reduce parser.

The following piece of code does that. The predicate `bake` first calls `pop`, which unifies the "topmost" element of the stack represented by its second argument (`Bag0`) with its first argument (`LeftDaughter`), and returns the rest of the stack as its third argument (`Bag1`). It then calls `delete`, which non-deterministically takes any element from the stack represented by its second argument (`Bag1`), unifying it with its first argument (`RigthDaughter`), and returning the rest as its third argument (`Bag2`). These two elements are then passed to the predicate `rule`, which finds a rule in the TL grammar to put them together, and finds the resulting sign. This sign is then replaced into the bag of TL signs to be baked, and the procedure finally calls itself recursively, until there is only one sign in the baking bag.

```
% bake(Constituents, Result)
% tries to make a Tree out of the (list / bag of) Constituents by
% keeping on picking two Constituents (nondeterministically) and putting
% them together with rule. The predicate delete removes its first
% argument from its second, returning the third. Thus the two deletes
% get two ``daughters'' out of Bag0, and Bag2 is what is left. The two
% daugthers are then combined into Mom (by the predicate rule) and Mom
% with Bag2 are passed on to bake again.

bake([Sign], [Sign]).

bake(Bag0, Tree):-
        pop(LeftDaughter,Bag0,Bag1),
        delete(RightDaughter,Bag1,Bag2),
        rule(Mom,LeftDaughter,RightDaughter),
        bake([Mom|Bag2], Bag).
```

It looks at first as if this is a very inefficient way of doing the baking, and to a large extent, it is. This algorithm is attempting to parse all possible permutations of the target words, and this is in addition to the well-known inefficiencies that arise from using Prolog's backtracking as a control strategy.

The inefficiencies of parsing will be partly solved in the next section, where we will see an alternative where well-formed constituents are recorded.

As for the former problem of having to look at all permutations, it is not as bad as it might seem, and this for two reasons. The first one is that most bakes (or parses of the permutation of the bag) fail at a very early stage, and hence the overheads are not too high. The second reason is that the word order in the translation can be changed around quite dramatically (consider, for instance, Dutch crossed-dependencies, or the word order in German subordinate clauses, compared with their English translations). It follows that word orderings very different to the original one have to be considered.

This does not necessarily mean that all permutations need to be looked at. One could attempt to drive the baking by perhaps starting with the "heads" (assuming we have such a notion available). There is certainly room for further work here, but it should be noted that simply looking at the syntactic category of the constituents in order to know where to start is not good enough, since so many constituents are type-raised. To take an example from the Spanish grammar under consideration, recall that a NP has the syntactic category S/S, which allows it to become the subject of a VP, since these have category S. On the other hand, a transitive verb subcategorizes for such NPs. This makes it impossible to tell, by simply looking at the NP, whether it is going to be the functor or the argument of larger constituent.

6.1.4 Generalised CKY baking

I give here a method for doing the baking which uses a table of well-formed constituents to overcome some of the inefficiencies mentioned in the previous section, which are inherent to naive baking using backtracking as the sole control strategy, and caused by wasting work by having to analyse the same substring several times if backtracking occurs.

The approach is to find all well-formed constituents of length 1, then all of length 2, and

```
For i = 2 up to n (the number of words in the bag) do:
    For all subsets S of the bag with i elements do:
        For all partitions of S into nonempty subsets S1 and S2 do:
            For all permutations of S1 and S2 which are recorded
            as well-formed constituents in the chart, do:
                see if the concatenation of S1 and S2 is licensed
                by some grammar rule;
                if so, record it in the chart;
            done.
        done.
    done.
done.
```

Figure 6.2: CKY baking algorithm.

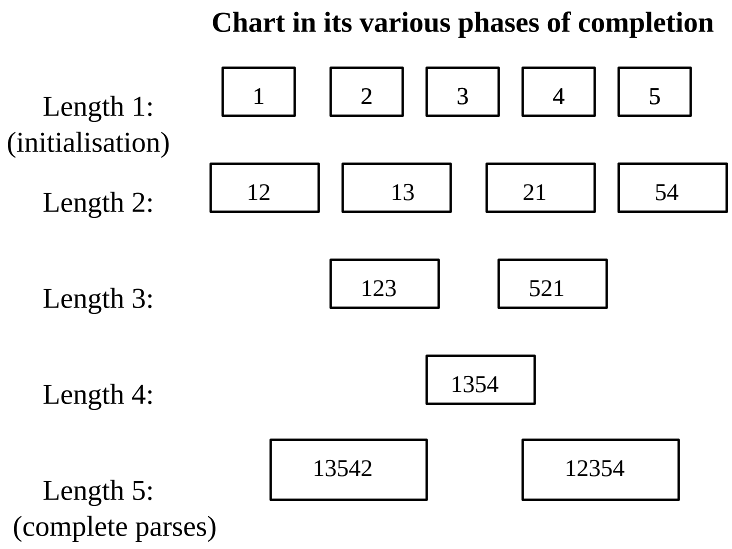
so on until we have found all well-formed constituents of the same length as the sentence. These constituents are recorded in a chart, and when looking for a constituent of length n, we therefore need to consider all the ways of combining together previously found (and recorded) constituents of length r and $n - r$. Because of the great similarity with the CKY approach, I have called this CKY baking.

The outline of the algorithm is therefore as follows:

Recall that we have a "bag" of words that we have to generate from. First of all, initialise the "chart" by recording the lexical entries for the elements in the bag. Figure 6.2 shows the steps involved.

At the end of this, each recorded permutation of S is an expression made from the original words, and therefore a possible translation of the SL expression.

A sample of such a baking process is given in Figure 6.3. The bag originally contains expressions labelled 1, 2, 3, 4 and 5, and the "edges" of the chart are represented by the boxes. At the end of the process, two baked strings are left.



Grammar

1, 2 -> 12
1, 3 -> 13
2, 1 -> 21
5, 4 -> 54
12, 3-> 123
13, 54 -> 1354
5, 21 -> 521
1354, 2 -> 13542
123, 54 -> 12354

Figure 6.3: CKY baking in progress.

6.2 Semantics

There are a couple of points that should be made about the semantics, concerning the nature of the predicates used (and in particular whether they should be language-specific), and about the need to Skolemise variables.

6.2.1 Semantic predicates

The first question is why the whole semantics is not simply copied across from the SL lexical entries to the TL ones during the lookup process, the intuition being that somehow the semantics have to be preserved during the translation process. However, this brings up an issue of modularity in the monolingual lexicons. If the monolingual lexicons are to be developed independently of each other, possibly by monolingual speakers, as seems desirable, then it makes sense to have semantic predicates (which act somewhat as semantic primitives) corresponding to predicates of the Natural Language in question. When the bilingual lexicon puts these signs into correspondence, all that is necessary is to unify the indices involved (rather than the predicates). The bilingual sign needs to have access to these, but access to the names of the predicates is not strictly necessary (since the bilingual lexicon co-indexes words by their phonologies).

This is therefore an essential difference with the interlingual approach. It enables us to avoid issues about the existence of universal semantic primitives, since the predicates are monolingual and are not carried across in the translation (only the indices are). It also avoids what [Landsbergen 87b] refers to as the Subset Problem, since there is no need to manipulate or carry out inferences on the semantic representation.

To illustrate this point, let us see what happens to the semantics of the sentence *Mary sees Madrid* as it gets translated to *María ve Madrid*. We start with the following entries, which only show the phonology and the semantics (it should be clear by now how the syntax puts these words together).

$$(6.1) \quad \begin{bmatrix} \text{Mary} \\ \text{M1} : \{\text{name(M1,mary)}\} \end{bmatrix}$$

The corresponding entry in Spanish is:

$$(6.2)\quad \begin{bmatrix} \text{María} \\ \text{I2} : (\{\text{name(M2,maria)}\} \cup \text{Sem2}) \end{bmatrix}$$

In the above example, *I2 : Sem2* is the semantics for the sentence that *María* modifies (recall that Spanish NPs are S/S). In other words, *María* has an index M2 which (bearing in mind what verbs look like) will fill a role slot of the sentence in which it appears.

Let's have a similar pair of entries for *Madrid-Madrid*:

$$(6.3)\quad \begin{bmatrix} \text{Madrid} \\ \text{X3} : \{\text{name(X3,madrid)}\} \end{bmatrix}$$

$$(6.4)\quad \begin{bmatrix} \text{Madrid} \\ \text{I4} : (\{\text{name(X4,madrid)}\} \cup \text{Sem4}) \end{bmatrix}$$

Finally, another couple for *sees-ve*:

$$(6.5)\quad \begin{bmatrix} \text{sees} \\ \text{I5} : (\left\{ \begin{array}{l} \text{see(I5), role(I5,agt,X5),} \\ \text{role(I5,pat,Y5)} \end{array} \right\} \cup \text{Sem1} \cup \text{Sem3}) \end{bmatrix}$$

$$(6.6)\quad \begin{bmatrix} \text{ve} \\ \text{I6} : \{\text{ver(I6), role(I6,agt,X6), role(I5,pat,Y6)}\} \end{bmatrix}$$

Mary sees Madrid produces the following unifications

$$\text{M1} = \text{X5, X3} = \text{Y5, Sem1} = \Big\{\text{name(M1,mary)}\Big\}, \text{Sem3} = \Big\{\text{name(X3,madrid)}\Big\}$$

The semantics then are:

$$I5 : \left\{ \begin{array}{l} see(I5),\ role(I5,agt,M1),\ name(M1,mary), \\ role(I5,pat,X3),\ name(X3,madrid) \end{array} \right\}$$

Now suppose the bilingual lexical entries put the following constraints on the above entries:

$$(6.7)\quad \begin{bmatrix} \text{Mary} \longleftrightarrow \text{Maria} \\ \text{M1} = \text{M2} \end{bmatrix}$$

$$(6.8)\quad \begin{bmatrix} \text{Madrid} \longleftrightarrow \text{Madrid} \\ \text{X3} = \text{X4} \end{bmatrix}$$

$$(6.9)\quad \begin{bmatrix} \text{sees} \longleftrightarrow \text{ve} \\ \text{I5} = \text{I6} \\ \text{X5} = \text{X6} \\ \text{Y5} = \text{Y6} \end{bmatrix}$$

Upon lexical lookup we end up with the following bag of Spanish words:

$$(6.10)\begin{bmatrix} \text{María} \\ \text{I2} : \{\text{name(M1,maria)}\} \cup \text{Sem2} \end{bmatrix}$$

$$(6.11)\begin{bmatrix} \text{Madrid} \\ \text{I4} : \{\text{name(M2,madrid)}\} \cup \text{Sem4} \end{bmatrix}$$

$$(6.12)\begin{bmatrix} \text{ve} \\ \text{I5} : \{\text{ver(I5), role(I5,agt,M1), role(I5,pat,X3)}\} \end{bmatrix}$$

This, together with the monolingual syntactic information for the target language (which comes from the monolingual signs) is enough to generate the correct target language sentence, as long as M1 and X3 do not unify. This is further explained in Subsection 6.2.2 below.

6.2.2 Grounding the semantics

There is one further key point involved with the semantics: before generation, the semantics of the terms have to be grounded or Skolemised, as we do not want M1 and X3 in the above example to unify. If that happened, we would end up with the two sentences *Maria ve Madrid* and *Madrid ve Maria*, both with the same semantics:

$$\text{I5} : \left\{ \text{ver(I5), role(I5,agt,M1), name(M1,maria), role(I5,pat,M1), name(M1,madrid)} \right\}$$

The sentences are infelicitous if *Madrid* and *María* do not designate the same entity.

The solution to this is to ground the variables before bilingual lookup takes place. This can be done with the `numbervars` predicate in Prolog, which instantiates the variables in an expression with unique non-variable terms. If the terms were, for instance, var(n) (where n is a number), then the semantics that would be produced by the time we get to bilingual lookup would be, instead of the above:

$$(6.13)\begin{bmatrix} \text{Maria} \\ \text{var(1)} : \{\text{role(var(1),agt,var(2)), name(var(2),maria)]}\} \cup \text{Sem2} \end{bmatrix}$$

$$(6.14)\begin{bmatrix} \text{madrid} \\ \text{var(1)} : \{\text{role(var(1),pat,var(3)), name(var(3),madrid)}\} \cup \text{Sem4} \end{bmatrix}$$

$$(6.15)\begin{bmatrix} \text{ve} \\ \text{var(1)} : \left\{ \begin{array}{l} \text{ver(var(1), role(var(1),agt,var(2)),} \\ \text{role(var(1),pat,var(3))} \end{array} \right\} \end{bmatrix}$$

Only the variables for semantic indices should be grounded, and not any other variables appearing in the semantics (such as uninstantiated argument names for instance, since in that case we would preclude the possibility of their being later instantiated to any "legal" argument name).

6.3 Different numbers of words

One other practical issue that arises is what to do when two expressions standing in a translation relation have different number of words. Up to now, we have been describing the bilingual lookup as a process which establishes a one-to-one correspondence between entries in the two monolingual lexicons. It is clearly not always the case that two corresponding expressions have the same number of words.

6.3.1 Zero-to-many correspondences

Two basic cases may arise. The first possibility is a lexical entry corresponds to nothing at all. This can be seen in the translation of *Mary wants to sing* as *María quiere cantar*, where the infinitival *to* marker in English has no equivalent in Spanish.

We may want to argue that such zero-to-many (zero-to-one in this case) correspondences do not really occur, and that what we have in this example is a correspondence between *want + to* and *querer* (ignoring verb inflections). Such a claim seems eminently plausible, and quite desirable. If we adopt this position, the discussion in this subsection is irrelevant. However, if we do not want to put such constraints on the bilingual lexicon, the alternative is to say, when going from English to Spanish, that *to* translates as the empty string. There is little danger in saying this, since after all it is not contributing anything to the semantics of the sentence. If then *sing* translates as *cantar*, the translation goes through in that direction.

In the other direction, we have a problem that is a common one encountered by anyone attempting generation: if there are words that do not contribute anything to the semantics, how do we put them into the sentence? If we just look up the Spanish words in the bilingual lexicon, we end up with the bag of English words for *Mary, wants, sing*, from which it is impossible to generate a grammatical sentence. The proposed solution is to modify the TL baking rules, so that an extra valid baking method is to take a TL word with no translation in the SL whenever it is needed (i.e. whenever it can immediately combine with a TL expression from the bag). The intuition behind this is that these words with empty translation are semantically empty (they are function words), so this is a safe operation to carry out. Furthermore, they belong to closed classes, so the search problem involved by having a set of words to which we may help ourselves at will during baking does not get out of hand.

Thus, if we are baking using either algorithm presented above (the naive one or the generalised CKY one), the rules that put two expressions together need to be modified, to include what can be thought of as a modification to the `lookup` procedure. Suppose that two expressions A and B are to be combined. An alternative to finding a rule that puts them together into an expression E is to find a function word F such that one of A or B (let's say A, without losing any generality) can combine with it to produce an expression C, which will in turn combine with B to give E. Changing A into C is similar to a unary rule, and as such, its application needs to be restricted if we are to avoid an excessively large search space (it could even be infinite if the unary rules are not sufficiently restrictive in their application). One way of avoiding this is to allow such modified lookup rule to take place only once before applying a “proper” binary rule.

(6.16) María quiere cantar.
María wants sing
‘Mary wants to sing.’

In the above example, English baking can only proceed of we allow *sing* to combine with *to*. The result of this will be the appropriate category for combining immediately with *wants*.

This restriction on unary rules keeps the search space reasonable, but is occasionally incorrect, since we may want to have more than one function word applying. An example is the sentence

(6.17) María quiere que llueva.
María wants that it-rain
‘Mary wants it to rain.’

When translating from Spanish to English, if we assume that the Spanish complementizer *que* translates as the English empty string, we end up with the bag with the three words *Mary*, *wants* and *rain*.

Before *wants* can combine with *rain*, two function words *to* and *it* have to apply to *rain*.

It seems unclear how to limit the number of such unary rules which introduce function words, and at the moment only one is allowed. This means that the above example cannot be handled.

6.3.2 Many-to-many correspondences

Another case in which we have to consider different number of words in the source and the target strings is the one that arises is in expressions such as *go out* which translates as *salir* (literally *exit*).

When translating from English to Spanish in such an example, it would be possible to say that the verb *go* translates as *salir* if it is in the context of *out*. One way to realise this in the current categorial approach is to say that there is an entry for *go* which subcategorizes for *out*, and whose translation is *salir*, and that a possible translation for *out* is the empty string. This is analogous to an entry in a standard dictionary, which would, under *go*, list *go out: salir*.

When translating in the other direction, we find ourselves with two ways of solving the problem. The first one is to use the entries used for translating from English to Spanish. We therefore end up with a translation of *salir* as the version of *go* subcategorizing for *out*, but the word *out* is not in the bag we are to bake from, and could be accessed as a "function word" using the modification outlined above.

However, this may lead to an unnecessary proliferation of such function words, as it could be argued that in this context *out* contributes just as much to the semantic content as *go*.

The second solution, which is along the lines of what was suggested in the previous chapter, is to modify the lookup algorithm so that bags of words may be put into correspondence with a single word. Thus when going from Spanish to English *salir* is taken out of the bag of Spanish words and *go* and *out* put into the English one, and when translating from English to Spanish, *go* and *out* are taken out of the English SL bag and *salir* put into the Spanish TL one.

The algorithm for generation (baking) is therefore unchanged, and only the lookup algorithm needs to be modified: it needs to scan the SL bag of (skolemised) words, and for any subset of this that appears in the bilingual lexicon, take the bag of words that constitutes the translation. In principle this may lead to an untractable complexity in the lookup algorithm (since we may end up looking up an exponentially large number of subsets of SL entries in the bilingual lexicon). However, in actual case it seems likely that the use of good indexing mechanisms together with the fact that these many-to-many correspondences are relatively small in number should keep the lookup procedure manageable.

6.4 Complexity issues

The computational complexity of Shake-and-Bake MT arises mainly from three places: the generation algorithm itself, the ambiguities of the bilingual lexicon, and the lexical redundancy rules. We shall briefly examine these issues.

6.4.1 Generation

The generation algorithms presented here are correct in the sense that they will produce all possible parses from the generation bag, but since they are equivalent to trying out all possible permutations of the elements in the bag, their complexity is factorial in the number of words.

[Brew 92] shows that Shake-and-Bake generation is NP-complete by reducing it to the of the *ménage à trois* problem, also known as 3-dimensional matching.

As for other NP-complete problems, although the general case is intractable there is however some scope for developing algorithms with reasonable average-case performance, and Brew offers some interesting insights on this using count invariants (functions over the syntactic categories of the elements bag, that remain invariant during the parse).

6.4.2 Ambiguities in the bilingual lexicon

Another source of combinatorial explosion is the bilingual lexicon itself. This arises from the fact that its equivalences are not one-to-one: any word may have several possible translations.

In the toy system developed, the vocabulary is not large enough for this to be a problem. In a more realistically-sized one, the bilingual signs would index the monolingual entries by word-sense identifiers, rather than by their orthographies as has been done here. If the semantic typing system is sufficiently sophisticated, it may be fair to assume that the SL word senses can be correctly identified.

Even with this assumption, it is likely that any word-sense in the SL may have more than one translation in the TL, since word-senses in the two languages are likely to overlap rather than being in a one-to-one correspondence (if there was such a correspondence, there would be no problem with the notion of universal semantic primitives). Thus bilingual lookup is a source of exponential complexity in the translation.

It should be pointed out that this is not a problem specific so Shake-and-Bake, but will arise in any other approach where lexical transfer is involved: if we want to obtain all linguistically possible translations, transfer ambiguities in the bilingual lexicon will produce exponentially large numbers of translations. One possible solution to this would be to use probabilistic methods to order the TL lexical choices and generate the most likely translations first. Another possible way of solving this would be if our linguistic formalism could express disjunctions. If the possible TL expressions corresponding to a SL one have the same syntax, and only differ in the name of the semantic predicates involved and the orthography, much of the complexity can be eliminated by representing the alternative TL signs as a single one with disjunctive values for orthography and semantic predicate. These approaches are beyond the scope of this work.

6.4.3 Ambiguities from lexical redundancy rules

The final source of complexity, and one which is closely tied to the above, arises from our use of lexical redundancy rules and our choice of matching monolingual entries by name: even if the bilingual lexicon contains a single TL entry corresponding to a given SL one, that TL entry may be "multiplied" out by TL lexical rules to produce a situation similar to the one described in the previous subsection.

Some of these lexical redundancy rules do not affect the semantics (taken in its broader sense, i.e. including information structure). This happens for instance with the one that produces dative shift in English. In this case solutions along the lines suggested in the previous subsection may be applicable.

Other lexical rules, such as the one that does Spanish subject inversion, have an effect on information structure (e.g. given vs. new information). If a better representation of information structure is used, some of the ambiguities arising from such rules may be eliminated, since it may be possible to express in the bilingual lexicon the fact that the information structure is preserved (instead of just having equations identifying semantic indices), forcing the use of lexical entries derived by means of suitable redundancy rules instead of those taken straight from the lexicon.

6.5 Concluding remarks

The last two chapters have set the scene for the definitive version of the Shake-and-Bake algorithm to be used, starting from the first approximation and gradually refining it.

In the next chapter we will see more complex examples of translations, and how they can be handled by Shake-and-Bake. In particular, we shall see some examples which cause some difficulties to more conventional approaches to MT, and we shall also see how it handles the small grammar fragment developed in Chapters 2, 3 and 4.

Chapter 7

More complex examples of translations

In this section we will see how more complex sample translations may be derived from lexical entries that will be built for these purposes. On the whole, we shall follow the order of exposition presented in Chapters 3 and 4 for the the monolingual grammars, and will use the notation presented in the last two chapters.

7.1 Morphology

Very little has been said about morphology so far. In the current system, it is simply not covered, which leads to a great amount of redundancy in the lexicons. For instance, the English verb *read* has to be put explicitly into correspondence with all the tensed Spanish forms, and all these have to be in the monolingual Spanish lexicon.

I shall briefly suggest how derivational morphology should be done in such a framework. Let us assume that the contents of the monolingual lexicons are morphemes, and that there is a morphological component to the parsing and generation processes. When a sentence is input, it is passed to this component, which strips it into individual morphemes. The translation process continues from there as described so far: the string of morphemes is parsed using the SL grammar and lexicon, bilingual lookup takes place, and the resulting bag of TL morphemes is passed to the generation process, which finds the possible translations in the form of strings of morphemes. This is then processed by a TL morphological generation component, which produces the final output.

Under this assumption, the bilingual lexicon would have correspondences between verb stems, and between inflectional affixes.

The translation of the sentence *Mary wants to sing* (6.16) would then be carried out as follows.

Let us assume that the correspondences between sets of monolingual entries are set up in the bilingual lexicon (using a shorthand notation where the "categories" contain syntactic feature information).

$$(7.1) \left\{ \begin{bmatrix} \text{'Mary'} \\ \text{np} \end{bmatrix} \right\} \left\{ \begin{bmatrix} \text{'María'} \\ \text{s/s} \end{bmatrix} \right\}$$

$$(7.2) \left\{ \begin{matrix} \begin{bmatrix} \text{want-} \\ \text{s\_base/np/vp\_to} \end{bmatrix} \\ \begin{bmatrix} \text{to} \\ \text{vp\_to/vp\_base} \end{bmatrix} \end{matrix} \right\} \left\{ \begin{matrix} \begin{bmatrix} \text{quer-} \\ \text{s\_stem/s\_inf} \end{bmatrix} \\ \begin{bmatrix} \text{inf} \\ \text{s\_inf/s\_stem} \end{bmatrix} \end{matrix} \right\}$$

$$(7.3) \left\{ \begin{bmatrix} \text{sing-} \\ \text{vp\_base} \end{bmatrix} \right\} \left\{ \begin{bmatrix} \text{cant-} \\ \text{s\_stem} \end{bmatrix} \right\}$$

$$(7.4) \left\{ \begin{bmatrix} \text{3sg} \\ \text{s/s\_base} \end{bmatrix} \right\} \left\{ \begin{bmatrix} \text{3sg} \\ \text{s/s\_stem} \end{bmatrix} \right\}$$

The translation (from English to Spanish) would then proceed as follows: from the input sentence (7.5 a), the morphological component obtains a string of morphemes (7.5 b). The English grammar parses them, and the lexical lookup process finds the Spanish equivalents (7.5 d) corresponding to the English bracketing of (7.5 c). The Spanish baking yields bracketing (7.5 e), and that is fed to the Spanish morphological component, producing (7.5 f).

(7.5) a Mary wants to sing.

b Mary want- 3sg to sing-

c Mary $\{$want- to$\}$ 3sg sing-

d María $\{$quer- inf$\}$ 3sg cant-

e María $\{$quer- 3sg$\}$ $\{$cant- inf$\}$

f María quiere cantar.

As was said above, the entries in the lexicons actually used are fully tensed (as can be seen in the Appendices).

7.2 Word order

The first set of examples that we will examine deals with Word Order, and in particular how to account for the much greater flexibility that Spanish allows compared with English.

The linear order of words is controlled, in standard UCG as well as here, by the ORDER feature, which may or may not be instantiated. If it is not instantiated, the word order is quite free. If we merely instantiate this ORDER feature in English, but not so often in Spanish (in order to reflect the greater rigidity of word order in one language compared with the other), we end up with a translation system where many possible Spanish translations are produced from a single English one, and, conversely, many Spanish sentences differing only in word order are translated by the same English one. This is just what the grammars stipulate, except that it might be desirable in the first case to choose the one with the most usual (or less marked) word order, and in the second case to capture somehow the stylistic differences conveyed by the different word orders.

The first issue will not be addressed here for lack of space. One possible way in which it could be approached is by the introduction of some measure of "non-canonicity" or "markedness" (perhaps with respect to a given interpretation involving the distinction between given and new information) in the word order of an expression. Some mechanism to handle preferences over word orders would be required (rather than the current binary distinction between compulsory and free order), so that the grammar-writer would be able to stipulate what the preferred order is, and how that ordering can be changed at the expense of increasing the "markedness" of the expression.

The second issue will be addressed under the section about clefts later on in this chapter, since many of the "non-standard" word orderings in Spanish can be translated as cleft constructions in English.

Let us start with a simple sentence consisting of a subject NP, a transitive verb and an object NP, such as

(7.6) a Mary visited Madrid.

b María visitó Madrid.

c Visitó Madrid María.

d Visitó María Madrid.

c Madrid visitó María.

As was pointed out in Chapter 2, Spanish allows all possible permutations for the verb and the NPs *except* those with verb-final orderings.

The English lexical entry for the verb simply subcategorizes for the object NP to its right, and for the subject NP to its left

$$(7.7)\quad \begin{bmatrix} \text{ORTHO} & \text{visited} \\ \text{CAT} & \text{s/}\left\{\begin{array}{l}\begin{bmatrix}\text{np:fwd:Patient}\end{bmatrix}\\ \begin{bmatrix}\text{np:back:Agent}\end{bmatrix}\end{array}\right\} \\ \text{SEM} & \text{E : }\left\{\text{visiting(E),role(E,agt,Agent),role(E,Patient)}\right\} \\ \text{ARG0} & \text{E} \\ \text{ARG1} & \text{Agent} \\ \text{ARG2} & \text{Patient} \end{bmatrix}$$

This sign will account for a unique English word ordering. When it comes to the Spanish one though, recall that the verb only subcategorizes for its object NP, and that the subject is a sentence modifier. Several entries for the verb exist, built by the use of lexical rules, as outlined in Chapter 3, which account for all the above permutations, while disallowing the ungrammatical verb-final ones. These rules may put some constraints on the syntax. For instance, we have one that effectively allows the transitive verb to be *preceded* by its object, *provided* some NP follows it, and this is controlled by the VFINAL feature of the verb.

Let us illustrate this with an example.

An entry for the above Spanish transitive verb is:

$$(7.8)\quad \begin{bmatrix} \textsc{ortho} & \text{visitó} \\ \textsc{cat} & \text{s}/\left\{ \begin{bmatrix} \textsc{cat} & \text{np} \\ \textsc{order} & \text{fwd} \\ \textsc{sem} & \text{P : Sem} \end{bmatrix} \right\} \\ \textsc{sem} & \text{E} : \left\{ \text{visitar(E),role(E,agt,A),role(E,P)} \right\} \cup \text{Sem} \\ \textsc{arg0} & \text{E} \\ \textsc{arg1} & \text{Agent} \\ \textsc{arg2} & \text{Patient} \end{bmatrix}$$

The necessary NPs are:

$$(7.9)\quad \begin{bmatrix} \textsc{ortho} & \text{'Madrid'} \\ \textsc{cat} & \text{np} \\ \textsc{sem} & \text{P} : \{\text{name(P,madrid)}\} \\ \textsc{arg0} & \text{P} \end{bmatrix}$$

$$(7.10)\quad \begin{bmatrix} \textsc{ortho} & \text{'María'} \\ \textsc{cat} & \text{s} / \begin{bmatrix} \textsc{cat} & \text{s} \\ \textsc{order} & \text{fwd} \\ \textsc{sem} & \text{E : Sem} \end{bmatrix} \\ \textsc{sem} & \text{E} : \{\text{name(F3,maria), role(F3,R,E)}\} \cup \text{Sem} \\ \textsc{arg0} & \text{F3} \end{bmatrix}$$

The two NPs above have the same structure. Different versions of the shorthand notation have been adopted to make it easier to understand their different behaviour with respect to the verb.

A second entry for the subject NP, which allows it to look *backwards* for the rest of the sentence, provided it it not verb-final, is produced by means of a lexical redundancy rule, yielding:

$$(7.11)\begin{bmatrix} \textsc{ortho} & \text{'María'} \\ \textsc{cat} & \text{s} \;/ \begin{bmatrix} \textsc{cat} & \text{s} \\ \textsc{order} & \text{back} \\ \textsc{vfinal} & - \\ \textsc{sem} & \text{E : Sem} \end{bmatrix} \\ \textsc{sem} & \text{E} : \left\{\text{name(F3,maria), role(F3,R,E)}\right\} \cup \text{Sem} \\ \textsc{arg0} & \text{F3} \end{bmatrix}$$

The above signs, and in particular the way in which the only word order specification is that the verb is looking for the object to its right, allow three orderings, all of which are grammatical. SVO and VOS (7.6 b & c) are produced by using function application, and VSO (7.6 d) is produced by composition. The fourth one, namely OVS (7.6 e), can be obtained with a second entry for the verb, which allows the object to precede it, as long as the subject follows it. This entry is built from the original one above using a lexical rule, and it *does* require the subject to be present. This will rule out OV as ungrammatical (recall that VO is perfectly grammatical). The resulting Spanish sign will therefore be quite similar to the English, except for the Word order and the NPs being different:

$$(7.12)\begin{bmatrix} \textsc{ortho} & \text{visitó} \\ \textsc{cat} & \text{s}/ \left\{ \begin{matrix} \begin{bmatrix} \textsc{cat} & \text{np} \\ \textsc{order} & \text{back} \\ \textsc{sem} & \text{P:Sem1} \end{bmatrix} \\ \begin{bmatrix} \textsc{cat} & \text{np} \\ \textsc{order} & \text{fwd} \\ \textsc{sem} & \text{P:Sem2} \end{bmatrix} \end{matrix} \right\} \\ \textsc{sem} & \text{E} : \left\{ \begin{matrix} \text{visitar(E),} \\ \text{role(E,agt,A),role(E,pat,P)} \end{matrix} \right\} \cup \text{Sem1} \cup \text{Sem2} \\ \textsc{arg0} & \text{E} \\ \textsc{arg1} & \text{Agent} \\ \textsc{arg2} & \text{Patient} \end{bmatrix}$$

This second version combines with its object NP to its left, and then with its subject NP to its left, providing us with a coverage for the fourth sentence, and only that one.

The correspondence between the English *visited* 7.7 and the two Spanish equivalents (7.8) and (7.12) arises from the Spanish lexical redundancy rules, so there is only one entry in the bilingual lexicon.

7.3 Subject pro-drop

Spanish subject pro-drop is handled by putting English subject pronouns into correspondence with the empty string in Spanish, as well as with the Spanish pronoun (so there are two bilingual entries for each English pronoun). If morphology was covered adequately (which it is not here), a better alternative would be to put the English subject pronouns in correspondence with the Spanish person and number inflectional morphemes.

Under the current approach, when translating from English into Spanish, the English subject may not be translated into Spanish, and since the Spanish verb does not require a subject (an intransitive verb, for instance, is a sentence), the translation may proceed in a straightforward manner.

Going in the other direction, the Spanish intransitive verb is translated into the English one. This alone is not sufficient to make a sentence, and at this stage the treatment of "function words" described in the previous chapter comes into play. Since the English pronouns do not translate into Spanish, they are similar to function words, and therefore may be freely used for English baking, even though they are not semantically empty.

To give an example from the previous subsection, consider the sentences:

(7.13) She visited Madrid
He visited Madrid
Visitó Madrid

Suppose we are to translate from Spanish into English. After the Spanish sentence is parsed and the bilingual lookup takes place, the only English constituent that can be formed is:

$$
(7.14)\begin{bmatrix}
\textsc{ortho} & \text{'visited Madrid'} \\
\textsc{cat} & \text{s/}\left\{\begin{bmatrix}\text{NP:back:Agent}\end{bmatrix}\right\} \\
\textsc{sem} & \text{e : }\left\{\begin{array}{l}\text{visiting(e),role(e,agt,x3),} \\ \text{role(e,m),name(m,madrid)}\end{array}\right\} \\
\textsc{arg0} & \text{e} \\
\textsc{arg1} & \text{x3} \\
\textsc{arg2} & \text{m}
\end{bmatrix}
$$

This on its own is not a sentence, as it subcategorizes for its subject NP, but we have run out of words from the baking bag.

At this stage, we may use the lookup procedure rule which introduces the English function word *she* or *he*, either of which is syntactically an NP, and is just what the above constituent requires. No other pronouns may be selected, since the index for the agent is constrained to be of semantic type third person singular human (as suggested by the choice of constant name). Either of the English sentences above then results.

7.4 Clitics

Note how in the above examples the bilingual entries do very little for us. This is an advantage of the present approach, in that the bilingual element of the translation is reduced to a bare minimum, and all the monolingual aspects of the translation are where they belong, namely in the monolingual grammars. If these correctly cover the individual languages involved, writing the bilingual correspondence is, as it should be, a straightforward task. The bilingual lexicon just accounts for the insertion of the clitics into the TL Shake-and-Bake bag (most of the time there is a 1-1 correspondence), and the monolingual grammars put them in the right order.

Using the monolingual grammars discussed in Chapters 3 and 4, the correspondences are trivial. The only thing that is not completely straightforward in the bilingual lexicon is clitic doubling (see below), but that is fairly simple too.

The clitics in the two languages were presented in Chapters 3 and 4, and most of the time there is a direct correspondence between them. The variations in word order in Spanish (where the clitic may precede the verb or follow it, and where there is the possibility of having "clitic climbing") are accounted for by syntactic features that indicate whether the verb is in finite form or not, and whether it is a lexical item (since clitics must combine directly with the verb, with no intervening elements).

Consider, for instance, the following sentences:

(7.15) María lo leyó
Mary read it

(7.16) Leer-lo
(To) read it

Provided that the bilingual lexicon puts *it* and *lo* into correspondence, and that the monolingual grammars properly cover the fact that Spanish clitics precede non-finite verbs, but English clitics follow them, the translation offers no problem. The small orthographic peculiarity that in Spanish, clitics that follow the verb are written as one word with it can be handled by a simple morphological processor. One other point that should be made is that in these first few examples, the case-assignment mechanism will be ignored, although direct object clitics do absorb case. This mechanism will be used in full for the indirect object clitics later on in this section.

The signs required for the above sentence are those for *María-Mary*, and *leyó-read*, which have already been shown and are reproduced below for the reader's convenience:

$$(7.17)\begin{bmatrix} \textsc{ortho} & \text{'María'} \\ \textsc{cat} & \text{s} \,/\, \begin{bmatrix} \textsc{cat} & \text{s} \\ \textsc{sem} & \text{E:Sem} \end{bmatrix} \\ \textsc{sem} & \text{E} : \{\text{name(F3,maria), role(F3,R,E)}\} \cup \text{Sem} \\ \textsc{arg0} & \text{F3} \end{bmatrix}$$

$$(7.18)\begin{bmatrix} \textsc{ortho} & \text{'Mary'} \\ \textsc{cat} & \text{np} \\ \textsc{sem} & \text{F3} : \{\text{role(F3,R,E)}\} \\ \textsc{arg0} & \text{F3} \end{bmatrix}$$

$$(7.19)\begin{bmatrix} \textsc{spanish} & \boxed{7.17} \begin{bmatrix} \textsc{arg0} & \text{M} \end{bmatrix} \\ \textsc{english} & \boxed{7.18} \begin{bmatrix} \textsc{arg0} & \text{M} \end{bmatrix} \end{bmatrix}$$

$$(7.20)\begin{bmatrix} \textsc{ortho} & \text{leyó} \\ \textsc{cat} & \text{s/}\left\{\begin{bmatrix} \textsc{cat} & \text{np} \\ \textsc{order} & \text{fwd} \\ \textsc{sem} & \text{P : Sem} \end{bmatrix}\right\} \\ \textsc{feats} & \begin{bmatrix} \textsc{fin} & + \end{bmatrix} \\ \textsc{lex} & + \\ \textsc{sem} & \text{E} : \left\{ \text{leer(E),role(E,agt,A),role(E,P)} \right\} \cup \text{Sem} \\ \textsc{arg0} & \text{E} \\ \textsc{arg1} & \text{Agent} \\ \textsc{arg2} & \text{Patient} \end{bmatrix}$$

$$(7.21)\begin{bmatrix} \textsc{ortho} & \text{read} \\ \textsc{cat} & \text{s/}\left\{\begin{matrix} \begin{bmatrix}\text{NP:fwd:Patient}\end{bmatrix} \\ \begin{bmatrix}\text{NP:back:Agent}\end{bmatrix} \end{matrix}\right\} \\ \textsc{sem} & \text{E} : \left\{ \text{reading(E),role(E,agt,Agent),role(E,Patient)} \right\} \\ \textsc{arg0} & \text{E} \\ \textsc{arg1} & \text{Agent} \\ \textsc{arg2} & \text{Patient} \end{bmatrix}$$

$$(7.22)\begin{bmatrix} \textsc{spanish} & \boxed{7.20}\begin{bmatrix} \textsc{arg0} & \text{E} \\ \textsc{arg1} & \text{A} \\ \textsc{arg2} & \text{P} \end{bmatrix} \\ \textsc{english} & \boxed{7.21}\begin{bmatrix} \textsc{arg0} & \text{E} \\ \textsc{arg1} & \text{A} \\ \textsc{arg2} & \text{P} \end{bmatrix} \end{bmatrix}$$

The first relevant signs for the clitics are the following:

$$(7.23)\begin{bmatrix} \textsc{ortho} & \text{lo} \\ \textsc{cat} & \text{s /}\begin{bmatrix} \textsc{cat} & \text{s} \\ \textsc{feats} & \begin{bmatrix} \textsc{fin} & + \end{bmatrix} \\ \textsc{lex} & + \\ \textsc{sem} & \text{E:Sem} \end{bmatrix} \\ \textsc{order} & \text{back} \\ \textsc{sem} & \text{E} : (\{\text{role(F3,R,X3)}\} \cup \text{Sem}) \\ \textsc{arg0} & \text{X3} \end{bmatrix}$$

$$(7.24)\begin{bmatrix} \textsc{ortho} & \text{it} \\ \textsc{cat} & \text{np} \\ \textsc{sem} & \text{X3} : \{\text{role(E,R,X3)}\} \\ \textsc{arg0} & \text{X3} \end{bmatrix}$$

$$(7.25)\begin{bmatrix} \textsc{spanish} & \boxed{7.23} \begin{bmatrix} \textsc{arg0} & \text{X3} \end{bmatrix} \\ \textsc{english} & \boxed{7.24} \begin{bmatrix} \textsc{arg0} & \text{X3} \end{bmatrix} \end{bmatrix}$$

We then have the signs for the infinitives:

$$(7.26)\begin{bmatrix} \textsc{ortho} & \text{leer} \\ \textsc{cat} & \text{s/}\left\{ \begin{bmatrix} \textsc{cat} & \text{np} \\ \textsc{order} & \text{fwd} \\ \textsc{sem} & \text{P : Sem} \end{bmatrix} \right\} \\ \textsc{feats} & \begin{bmatrix} \textsc{fin} & - \end{bmatrix} \\ \textsc{lex} & + \\ \textsc{sem} & \text{E} : (\left\{ \text{leer(E),role(E,agt,A),role(E,P)} \right\} \cup \text{Sem}) \\ \textsc{arg0} & \text{E} \\ \textsc{arg1} & \text{Agent} \\ \textsc{arg2} & \text{Patient} \end{bmatrix}$$

$$(7.27)\begin{bmatrix} \textsc{ortho} & \text{read} \\ \textsc{cat} & \text{vp-inf/}\left\{ \begin{bmatrix} \text{NP:fwd:Patient} \end{bmatrix} \right\} \\ \textsc{sem} & \text{E} : \left\{ \text{reading(E),role(E,agt,Agent),role(E,Patient)} \right\} \\ \textsc{arg0} & \text{E} \\ \textsc{arg1} & \text{Agent} \\ \textsc{arg2} & \text{Patient} \end{bmatrix}$$

$$(7.28)\begin{bmatrix} \textsc{spanish} & \boxed{7.26} \begin{bmatrix} \textsc{arg0} & \text{E} \\ \textsc{arg1} & \text{A} \\ \textsc{arg2} & \text{P} \end{bmatrix} \\ \textsc{english} & \boxed{7.27} \begin{bmatrix} \textsc{arg0} & \text{E} \\ \textsc{arg1} & \text{A} \\ \textsc{arg2} & \text{P} \end{bmatrix} \end{bmatrix}$$

Finally, we have the clitics that will combine with these infinitives:

$$(7.29)\begin{bmatrix} \textsc{ortho} & \text{lo} \\ \textsc{cat} & \text{s} \ / \begin{bmatrix} \textsc{cat} & \text{s} \\ \textsc{feats} & \begin{bmatrix} \textsc{fin} & - \end{bmatrix} \\ \textsc{lex} & + \\ \textsc{sem} & \text{E : Sem} \end{bmatrix} \\ \textsc{order} & \text{fwd} \\ \textsc{sem} & \text{E} : \{\text{role(F3,R,X3)}\} \cup \text{Sem} \\ \textsc{arg0} & \text{X3} \end{bmatrix}$$

$$(7.30)\begin{bmatrix} \textsc{spanish} & \boxed{7.29} \begin{bmatrix} \textsc{arg0} & \text{X3} \end{bmatrix} \\ \textsc{english} & \boxed{7.24} \begin{bmatrix} \textsc{arg0} & \text{X3} \end{bmatrix} \end{bmatrix}$$

These signs may be summarised by saying that we have two versions of the Spanish direct object clitic, one which combines with a verb in finite form to its right, the other which combines with a verb in non-finite form to its left. Both of these translate as the single version of the English clitic, which always follows its verb.

Spanish clitic doubling, where a dative clitic and its corresponding lexical NP may coexist, is also equally straightforward. Recall from Chapter 3 that it was covered by having essentially two versions of the clitic, one which absorbed case (and therefore precluded the existence of the NP), and one which did not (and which was compatible with the presence of such an NP). In English, on the other hand, all clitics are case-absorbing under this treatment, and thus may not co-occur with the lexical NP. The Spanish case-absorbing clitic is put into correspondence in the bilingual lexicon with the English clitic, and the Spanish non case-absorbing clitic is put into correspondence with the empty string in English, which makes it behave like the "function words" described in the last chapter.

The consequence of this is that by the end of the parse, if the Spanish clitic turned out to be case-absorbing (i.e. if no corresponding lexical NP was found), it gets translated into the English clitic, and otherwise it gets translated into English as the empty string.

To illustrate this, let us consider the entries for the Spanish indirect object clitic *le*. The case-absorbing version is just like the one for the direct object clitic (there are actually two of these, depending on which side of the verb they attach, but we shall only consider the one combining with a finite verb).

$$(7.31)\begin{bmatrix} \textsc{ortho} & \text{le} \\ \textsc{cat} & \text{s} \ / \begin{bmatrix} \textsc{cat} & \text{s} \\ \textsc{case} & \{\text{dat}\} \cup \text{Cases} \\ \textsc{feats} & \begin{bmatrix} \textsc{fin} & + \end{bmatrix} \\ \textsc{lex} & + \\ \textsc{sem} & \text{E:Sem} \end{bmatrix} \\ \textsc{order} & \text{back} \\ \textsc{case} & \text{Cases} \\ \textsc{sem} & \text{E:}\{\text{role(F3,R,X3)}\} \cup \text{Sem} \\ \textsc{arg0} & \text{X3} \end{bmatrix}$$

The non-case absorbing version is:

$$(7.32)\begin{bmatrix} \textsc{ortho} & \text{le} \\ \textsc{cat} & \text{s} \ / \begin{bmatrix} \textsc{cat} & \text{s} \\ \textsc{case} & \text{Cases} \\ \textsc{feats} & \begin{bmatrix} \textsc{fin} & + \end{bmatrix} \\ \textsc{lex} & + \\ \textsc{sem} & \text{E:Sem} \end{bmatrix} \\ \textsc{order} & \text{back} \\ \textsc{case} & \text{Cases} \\ \textsc{sem} & \text{E} : (\{\text{role(F3,R,X3)}\} \cup \text{Sem}) \\ \textsc{arg0} & \text{X3} \end{bmatrix}$$

The first corresponds to the case when no NP is found, and thus translates as the English clitic *it*, and the second to the case when the lexical NP is present, and thus does not get translated into English.

The English clitic then is:

$$(7.33)\begin{bmatrix} \textsc{ortho} & \text{it} \\ \textsc{cat} & \text{np} \\ \textsc{sem} & \text{X3} : \{\text{role(E,R,X3)}\} \\ \textsc{arg0} & \text{X3} \end{bmatrix}$$

The first of these Spanish clitics translates as the English *it*:

$$(7.34)\begin{bmatrix} \textsc{spanish} & \boxed{7.31} \begin{bmatrix} \textsc{arg0} & \text{X3} \end{bmatrix} \\ \textsc{english} & \boxed{7.33} \begin{bmatrix} \textsc{arg0} & \text{X3} \end{bmatrix} \end{bmatrix}$$

The second one translates as the empty string:

$$(7.35)\begin{bmatrix} \text{SPANISH} & \boxed{7.32} \\ \text{ENGLISH} & \end{bmatrix}$$

When translating from Spanish to English, if the indirect object lexical NP is present, then the case-absorbing clitic is ruled out, as the verb will only be able to assign case to one of the two (as explained in Chapter 3), and we are forced to choose the non-case absorbing version, and hence the empty string in English translates as the Spanish clitic. This puts the right words in the Target Language baking bag, and the translation proceeds. If on the other hand the lexical NP is absent from the Spanish, the non-case absorbing clitic is ruled out, because the verb subcategorizes for an indirect object (which may be a full lexical NP or a clitic), and it stipulates that this indirect object must absorb case. It follows that the clitic translates as the English clitic, and since there is no lexical indirect object NP, we end up again with the right words in the baking bag.

Going in the other direction, if a clitic is found in English, it is translated as the Spanish (case-absorbing) clitic, and if a lexical NP is found, the question arises of whether a clitic is required in Spanish or not. According to the above description, both options are possible, since the Spanish non-case-absorbing clitic is treated as a function word (in the sense of the last chapter). Hence the following English sentence has the two possible Spanish translations:

(7.36) Mary gave the book to John
María dio el libro a Juan
María le dio el libro a Juan

As was mentioned in Chapter 2, both Spanish sentences are grammatical, and indeed both are produced. However, the second version is slightly preferred for stylistic reasons. If some ordering of the preferences in the target language was to be included, this could be done by the mechanism outlined in the previous section, but has not been done here.

When going in the other direction, the asymmetry of the monolingual clitics, namely that there are case absorbing and non-case versions for the Spanish one, whereas all English clitics are case absorbing, means that we do not generate clitic-doubled sentences in English, such as:

(7.37) *Mary gave him the book to John

The reason is simply that, even if there is a (doubled) clitic in Spanish, it is identified as such in the parse (because there is a NP that will absorb the case and force the clitic to be non-case-absorbing), and hence it will be translated as the empty string in English.

Finally, as far as the more interesting aspects of Spanish clitic behaviour are concerned (such as climbing), these are purely monolingual issues which are covered in the monolingual grammar. This means that as long as the bilingual lexicon puts the clitics into correspondence, the linear ordering properties of the Spanish clitics will allow them to "climb" when translating into Spanish. Consequently, the following English sentence has the Spanish translations shown below:

(7.38) Mary wants to read it.
María quiere leer-lo.
María lo quiere leer.

The syntactic features that allow the Spanish clitic to "climb" over its verb were outlined in Chapter 3. The first sentence is produced in a fairly standard way by letting *leer* (which subcategorizes for an object) combine with the clitic *lo*. The result of that then serves as an argument for *quiere*, which gives us the "verb phrase". Alternatively, the clitic and the "upper" verb may be put together using function composition, giving the constituent *lo quiere*, which subcategorizes for the "lower verb".

This illustrates one of the main strengths of the Shake-and-Bake approach. All the clitic climbing behaviour is a purely monolingual aspect of Spanish. As such, so long as it is properly covered in the monolingual Spanish grammar, the bilingual component of the translation system has little to say about it (and is even blatantly uninteresting in this respect). This compartmentalisation of information is generally seen as a desirable feature in Machine Translation systems, and it is one that standard approaches have great difficulty meeting.

7.4.1 Leísmo

Leísmo (see subsection 2.2.2) occurs when a human direct object, which is constructed with the particle *a*, because of the similarity with the indirect object, is replaced with the usual dative clitic *le* instead of the accusative one. The behaviour of this clitic follows the standard pattern, in that clitic doubling is then allowed.

(7.39) Mary saw John.
María vio a Juan.
María le vio.
María le vio a Juan.

This is very common in many Spanish dialects. If we take the approach that *a* behaves like an object marker for NPs which are semantically typed as denoting human entities, and for which the monolingual lexical entry specifies a nominative case in its bare form, then the *leísmo* clitics may be covered by entries very similar to the ones for the dative clitic, in that they may optionally absorb case, but which differ from them in that their case is accusative and they constrain their semantics index to stand for a human.

This means that if we want to have *leísmo* in our grammar, we need to include these entries for *le*, together with bilingual entries putting these into correspondence with *it* (for the case-absorbing versions) and the empty string (for the non-case-absorbing ones).

7.5 Dislocation, topicalisation and it-clefts

Topicalisation and Dislocation were discussed in Chapter 2 in the context of Spanish grammar. They describe situations in which an NP appears in a non-standard position, and there an unusual intonational pattern.

Dislocation involves a clear break between the dislocated NP and the rest of the sentence, and occurs both in Spanish and in English:

(7.40) **El libro,** lo compré esta tarde.
'The book, I bought it this afternoon'

Topicalisation on the other hand, which is often translated as it-clefts in English, occurs when the NP is strongly stressed, but without an intonational break between it and the rest of the sentence, as in:

(7.41) EL LIBRO compré esta tarde (no el disco).
The book I bought-1sg this afternoon.
'It was the book that I bought this afternoon (not the record).'

7.5.1 Dislocation

It was argued in earlier chapters that dislocated elements behave very much like sentential adjuncts in both languages. These may be generated by lexical rules from the lexical entries, and these lexical rules clearly affect the phonology, something that will be crudely represented here by putting a comma after the NP, in a treatment that bears some resemblance to that in [Popowich 88].

As was mentioned in Chapter 2, the dislocated element seems to behave like other extracted constituents in obeying the Complex NP Constraint (7.42 a), but that is the only similarity as shown in (7.42 b, c, and d).

The following sentences were used as examples in Chapter 2:

(7.42) a ? **Carlos,** María conoce al periodista que lo entrevistó.
Charles, Mary knows to the journalist who CL interviewed
'Charles, Mary knows the journalist who interviewed him'

b **Carlos,** ¿ quién sabe quién lo ha visto?
Charles, who knows who CL has seen
'Charles, who knows who has seen him?'

c **A Juan, Carlos,** se lo ha presentado María.
To John Charles CL-IO CL-DO has introduced Mary.
'Mary introduced Charles to John'

d **Carlos, a Juan,** se lo ha presentado María.

In principle, it is quite easy to write signs that can be freely "extracted" (to use the standard term in the literature, although in the current approach there is no such thing as extraction or movement). As shown in Chapter 3, we require signs which act as sentence modifiers, further specifying one of the arguments in the semantics of the sentence with which they combine:

$$(7.43)\begin{bmatrix} \text{ORTHO} & \text{',Carlos,'} \\ \text{CAT} & \text{s} \\ \text{SLASH} & \left\| \begin{bmatrix} \text{CAT} & \text{s} \\ \text{SEM} & \text{I : Sem} \end{bmatrix} \right\| \\ \text{SEM} & \text{I : }(\{\text{role(J,R,C), name(C,carlos), focus(C)}\} \cup \text{Sem}) \end{bmatrix}$$

As can be seen from the semantics, it is specified that the index C (standing for *Carlos*) plays some role R in an event J, but this event need not be the one referred to by the main sentence being modified, and this is what allows the "extraction".

For instance, it may combine with the following sentence, where the two *wh*-words are crudely represented in the semantics by *which*, which behaves somewhat like a quantifier.

$$(7.44)\begin{bmatrix} \text{ORTHO} & \text{'Quién sabe quién lo ha visto?'} \\ \text{CAT} & \text{s} \\ \text{SEM} & \text{E : (}\left\{ \begin{array}{l} \text{which(X),which(Y),saber(E),} \\ \text{role(E,agt,X), role(E,prop,I),} \\ \text{ver(I),role(I,agt,Y), role(I,pat,Z)} \end{array} \right\} \end{bmatrix}$$

Clearly, when we combine the two signs, we want C (for Carlos) to unify with Z, which is the only variable which is still "unbound" in the semantics. This requires a special treatment for the quantifiers during the unification process of the semantics, which has not been dealt with in detail. Assuming such a treatment, the two expressions above can combine easily to give:

$$(7.45)\begin{bmatrix} \text{ORTHO} & \text{'Carlos, quién sabe quién lo ha visto?'} \\ \text{CAT} & \text{s} \\ \text{SEM} & \text{E : (}\left\{ \begin{array}{l} \text{which(X),which(Y),saber(E),} \\ \text{role(E,agt,X), role(E,prop,I), ver(I)} \\ \text{role(I,agt,Y), role(I,pat,c), name(c,carlos)} \end{array} \right\} \end{bmatrix}$$

This example translates directly into English, and so we merely require to put the Spanish dislocated NP with its English counterpart. Their syntax will be the same in both languages (they are sentence modifiers), as will be their semantics.

However, this dislocated sign does not account for the questionable nature of dislocation out of complex NPs, such as as in the following sentence:

$$(7.46)\begin{bmatrix} \text{ORTHO} & \text{'María ha conocido al periodista que lo ha entrevistado'} \\ \text{CAT} & \text{s} \\ \text{SEM} & \text{E : (}\left\{\begin{array}{l} \text{conocer(E),role(E,agt,M),name(M,maria),} \\ \text{role(E,pat,P),periodista(P),restriction(P,I),} \\ \text{entrevistar(I),role(I,agt,P),role(I,pat,Z)} \end{array}\right\} \end{bmatrix}$$

The above dislocated NP combines with this by means of the unification of C with Z above. It could be argued that the problem with (7.42 a) is a semantic one, and that the difference between the roles of Z in (7.44) and in 7.46 is that the subordinate event I is a semantic argument of the main one (E) in the first case, but is just a restriction on one of E's arguments in the second one. Exactly what restrictions are at play is unclear, and hence this treatment is only a rough approximation.

7.5.2 Topicalisation and it-clefts

Spanish topicalisation occasionally corresponds to English topicalisation, but as there is less freedom in English as to what can be topicalised, it often has to be rendered by a cleft construction instead. The difference in Spanish between topicalised NPs and ordinary ones is that the topicalised ones are stressed (as before, we shall represent this crudely by capitalising them), and they refer to new information in the semantics. Other than that, they behave exactly like other NPs in that they absorb case, and hence, from a monolingual point of view, they are treated in the usual way for NPs. The topicalised versions can be generated from the base lexical entries using monolingual lexical redundancy rules.

The same can be said for English, although the same elements cannot always be topicalised. In other words, if we merely said that a topicalised NP in Spanish corresponds to the same topicalised NP in English, it would not necessarily give a grammatical result. Instead, very often we need to translate this as a cleft construction. Consider, for instance, the following example:

(7.47) A MARÍA vi esta tarde (no a Juan).
?MARY I saw this afternoon (not John).
It was Mary that I saw this afternoon (not John).

A cleft construction offers a better translation than topicalising A MARÍA. The Spanish particle *a* acts as a "human" marker for object NPs, and as such it is treated as a "function word" with no translation in English (as discussed in Section 6.3). Consequently, when translating from Spanish to English it does not enter the generation bag during lexical lookup, and when translating from English to Spanish it is introduced during the generation.

The correspondence that is therefore required from the bilingual lexicon is between the topicalised NP *MARÍA* and the phrase *it was Mary that*, with the Spanish particle *a* introduced when the topicalised element is an object, but not if it is the subject (as in *It was Mary that came this afternoon*).

Having such a correspondence between these two expressions would then solve the problem:

$$(7.48)\begin{bmatrix} \text{SPANISH} & \boxed{7.49} & \begin{bmatrix}\text{ARG0} & \text{F3}\end{bmatrix} \\ \text{ENGLISH} & \boxed{7.50} & \begin{bmatrix}\text{ARG0} & \text{F3}\end{bmatrix} \end{bmatrix}$$

Such an entry would have to contain pointers to the Spanish topicalised NP and to the English clitic, which are as follows:

$$(7.49)\begin{bmatrix} \text{ORTHO} & \text{'MARÍA'} \\ \text{CAT} & \text{s} \ / \begin{bmatrix} \text{CAT} & \text{s} \\ \text{CASES} & \{\_\text{Case}\} \cup \text{Rest} \\ \text{SEM} & \text{E:Sem} \end{bmatrix} \\ \text{CASES} & \text{Rest} \\ \text{SEM} & \text{E} : (\{\text{role(E,R,F3), name(F3,maria), new(F3)}\} \cup \text{Sem}) \\ \text{ARG0} & \text{F3} \end{bmatrix}$$

$$(7.50)\begin{bmatrix} \textsc{ortho} & \text{'it was Mary that'} \\ \textsc{cat} & \text{s} \\ \textsc{slash} & \left\| \begin{bmatrix} \textsc{cat} & \text{s} \\ \textsc{slash} & \left\| \begin{bmatrix} \textsc{cat} & \text{np} \\ \textsc{slash} & \| \, \| \\ \textsc{sem} & \text{E} : (\{\text{name(F3,mary)}, \text{new(F3)}\} \cup \text{Sem}) \end{bmatrix} \right\| \\ \textsc{order} & \text{fwd} \\ \textsc{feats} & \text{F} \\ \textsc{sem} & \text{Sem} \end{bmatrix} \right\| \\ \textsc{feats} & \text{F} \\ \textsc{sem} & \text{Sem} \\ \textsc{arg0} & \text{F3} \end{bmatrix}$$

Except for the extra element of the semantics telling us that it is new information, the English one is is just like a standard type-raised NP.

Clearly, this process generalises to other NPs, so we want some way of stating a global correspondence between Spanish topicalised elements and English clefts.

The bilingual lexicon establishes a correspondence between NPs such as *María* and *Mary*. A possible solution for getting a second correspondence like the one outlined above would be to introduce **bilingual lexical redundancy rules**.

Such rules would build bilingual correspondences from existing ones, in a manner analogous to the way in which their monolingual counterparts build monolingual lexical entries from the ones defined.

Let us assume that the English cleft is built from suitable lexical entries which identify the item being topicalised somewhere amongst their ARG features. Without going into the details of how they are combined syntactically, they can be represented as:

$$(7.51)\begin{bmatrix} \textsc{ortho} & \text{it} \\ \textsc{arg0} & \text{X} \end{bmatrix}$$

$$(7.52)\begin{bmatrix} \textsc{ortho} & \text{was} \\ \textsc{arg1} & \text{X} \end{bmatrix}$$

$$(7.53)\begin{bmatrix}\textsc{ortho} & \text{that}\\ \textsc{arg2} & \text{X}\end{bmatrix}$$

A bilingual lexical redundancy rule representing the correspondence required would then look like the following:

Bi – Top – Cleft

$$(7.54)\quad \begin{bmatrix}\textsc{spanish} & \boxed{1}\left\{\begin{bmatrix}\textsc{ortho} & \text{SO}\\ \textsc{cat} & \text{s/s}\\ \textsc{sem} & \text{X : SSem}\\ \textsc{arg0} & \text{E}\end{bmatrix}\right\}\\ \textsc{english} & \boxed{2}\left\{\begin{bmatrix}\textsc{ortho} & \text{EO}\\ \textsc{cat} & \text{np}\\ \textsc{sem} & \text{Y : ESem}\\ \textsc{arg0} & \text{E}\end{bmatrix}\right\}\end{bmatrix} \longrightarrow \begin{bmatrix}\textsc{spanish} & \boxed{3}\left\{\begin{bmatrix}\textsc{ortho} & \textsc{small caps}\ \text{(SO)}\\ \textsc{cat} & \text{s/s}\\ \textsc{sem} & \text{Y : (\{new(E)\}} \cup \text{SSem)}\\ \textsc{arg0} & \text{E}\end{bmatrix}\right\}\\ \textsc{english} & \boxed{4}\left\{\begin{array}{ll}\boxed{7.51} & \begin{bmatrix}\textsc{ortho} & \text{it}\\ \textsc{arg0} & \text{E}\end{bmatrix}\\ \boxed{7.52} & \begin{bmatrix}\textsc{ortho} & \text{was}\\ \textsc{arg1} & \text{E}\end{bmatrix}\\ \boxed{2} & \\ \boxed{7.53} & \begin{bmatrix}\textsc{ortho} & \text{that}\\ \textsc{arg3} & \text{E}\end{bmatrix}\end{array}\right\}\end{bmatrix}$$

It can be interpreted as saying that whenever we have a bilingual correspondence between items unifying with $\boxed{1}$ and $\boxed{2}$, we may build a bilingual correspondence between items $\boxed{3}$ and $\boxed{4}$.

A procedural description of how such a rule operates can be outlined as follows. If we are translating from Spanish to English, the lookup procedure will be presented with an item like $\boxed{3}$ (of course, it will be more instantiated, at least in its orthography). If there is no such entry in the bilingual lexicon, it will use the bilingual lexical rule to see of there is an entry like $\boxed{1}$. If so, it will find $\boxed{2}$, and return $\boxed{4}$ as the translation equivalent of $\boxed{3}$.

The above bilingual lexical redundancy rule (7.54) applies to bilingual entries with two lexical NPs to produce an entry setting up a correspondence between a Spanish topicalised lexical NP and an English cleft construct. In order to achieve a similar effect with non-lexical NPs, a further bilingual redundancy rule is required for determiners, which would build, from a bilingual entry relating two determiners, one which would put into correspondence a topicalised Spanish determiner with an English cleft.

These bilingual lexical rules have not been implemented in the current system, but they are faithful to the principles of lexicalist MT. Such rules are liable to be a source of ambiguity in the bilingual lexicon, but if the entries are adequately constrained by their information structure (as hinted by having *new(E)* in the semantics of $\boxed{3}$), this ambiguity need not lead us to a combinatorial explosion.

7.6 Two final examples

To finish off this section, let us see two small examples which are slightly awkward to describe in a transfer-based approach because of the complexity of the transfer rules involved, but which are quite simple using Shake-and-Bake. These are known as Argument Switching and Head Switching.

Much recent debate about this subject was sparked off by [Kaplan et al. 89]. See [Sadler and Thompson 91] and [Whitelock 91] for further discussion of the topics.

7.6.1 Argument switching

Argument switching occurs for instance with verbs like *to like* which translates as *gustar* (literally, *to please*), where the two argument roles are switched around. In other words, *John likes Mary* translates as *María gusta a Juan.*

The entries required to do this are quite simple. For the monolingual verbs, they are almost identical copies of any other transitive verbs. In fact, only the semantic predicates are different, and we could have (ignoring everything but the semantics, and writing that in the shorthand notation introduced in Chapter 5):

$$(7.55)\begin{bmatrix}\text{likes}\\ \text{I5} : \{\text{like(I5), role(I5,agt,X5), role(I5,pat,Y5)}\}\end{bmatrix}$$

$$(7.56)\begin{bmatrix}\text{gusta}\\ \text{I6} : \{\text{gustar(I6), role(I6,agt,X6), role(I5,pat,Y6)}\}\end{bmatrix}$$

The bilingual entry merely requires to change the arguments around:

$$(7.57)\begin{bmatrix}\textsc{spanish} & \boxed{7.56}\begin{bmatrix}\textsc{arg0} & \text{E}\\ \textsc{arg1} & \text{X}\\ \textsc{arg2} & \text{Y}\end{bmatrix}\\ \textsc{english} & \boxed{7.55}\begin{bmatrix}\textsc{arg0} & \text{E}\\ \textsc{arg1} & \text{Y}\\ \textsc{arg2} & \text{X}\end{bmatrix}\end{bmatrix}$$

In the translation of the sentence *John likes Mary* into Spanish, *likes* first combines with *Mary*, and the result of that with *John*. This results in the unification of Y with the index for *John* and of X with that of Mary. Let us assume that after Skolemisation, these are called j and m, respectively. Upon bilingual lexicon lookup, we end up with the following words in our baking bag:

$$(7.58)\begin{bmatrix}\textsc{ortho} & \text{'María'}\\ \textsc{cat} & \text{s} \,/\, \begin{bmatrix}\textsc{cat} & \text{s}\\ \textsc{case} & \text{nom}\\ \textsc{sem} & \text{e:Sem}\end{bmatrix}\\ \textsc{sem} & \text{e} : (\{\text{name(m,maria), role(e,R,m)}\} \cup \text{Sem})\\ \textsc{arg0} & \text{m}\end{bmatrix}$$

$$(7.59)\begin{bmatrix}\textsc{ortho} & \text{gusta}\\ \textsc{cat} & \text{s/}\left\{\begin{bmatrix}\textsc{cat} & \text{s} \,/\, \left\{\begin{bmatrix}\textsc{cat} & \text{s}\\ \textsc{case} & \text{dat}\\ \textsc{arg0} & \text{e}\\ \textsc{arg1} & \text{m}\\ \textsc{arg2} & \text{j}\end{bmatrix}\right\}\\ \textsc{sem} & \text{e} : \boxed{1}\left\{\begin{array}{l}\text{gustar(e),role(e,agt,m)}\\ \text{role(e,pat,j)}\end{array}\right\}\end{bmatrix}\right\}\\ \textsc{sem} & \text{e:}\boxed{1}\\ \textsc{arg0} & \text{e}\\ \textsc{arg1} & \text{m}\\ \textsc{arg2} & \text{j}\end{bmatrix}$$

$$(7.60)\begin{bmatrix} \textsc{ortho} & \text{'Juan'} \\ \textsc{cat} & \text{s} \,/\, \begin{bmatrix} \textsc{cat} & \text{s} \\ \textsc{case} & \text{nom} \\ \textsc{sem} & \text{e:Sem} \end{bmatrix} \\ \textsc{sem} & \text{e} : (\{\text{name(j, juan), role(e,R,j)}\} \cup \text{Sem}) \\ \textsc{arg0} & \text{j} \end{bmatrix}$$

The semantics already stipulate which of the two NPs will be the agent of the verb. The only difficulty is that the verb requires an accusative NP, and both of these are nominative since in Spanish, direct objects standing for humans require the particle *a* and hence these bare NPs without that particle must be nominative. At this stage, the only way to proceed with the baking is to use the function word *a*, which may make the NP with which it combines accusative. Having it apply to *Juan* is the only way that will give an accusative NP with semantics as stipulated by the verb *gusta* (i.e. an NP whose semantic index is *j*).

7.6.2 Head switching

Head switching is the phenomenon where the head word in one language translates as a non-head in the other. An example is when words of movement are lexicalized differently, such as in *Mary swam across the river*, which translates as *María cruzó el río nadando* (literally *Mary crossed the river swimming*).

This problem is an important one for transfer-based approaches, since they require a explicit transfer rule to deal with such cases. The solution to these in Shake-and-Bake is again straightforward. We merely need to say that, in the above example *cruzó* corresponds to *across*, and *swam* to *swimming* (in addition to the standard correspondences *cruzó-crossed* and *swam-nadó*, etc). This, together with an appropriate correspondence between the indices, gets us the translation required.

As this is part of quite a general pattern, such correspondences should, in a more sophisticated system, be built by some general lexical rule applicable to many verbs of movement, rather than on an *ad-hoc* basis.

The monolingual entries for *across* and *cruzó* are as follows:

$$(7.61)\left[\begin{array}{ll}
\textsc{ortho} & \text{across} \\
\textsc{cat} & \text{s}/\left\{\left[\begin{array}{ll}\textsc{cat} & \text{np} \\ \textsc{sem} & \text{Crossed:Sem1}\end{array}\right]\left[\begin{array}{ll}\textsc{cat} & \text{s} \\ \textsc{sem} & \text{E1:Sem2} \\ \textsc{arg0} & \text{E1} \\ \textsc{arg1} & \text{Crosser}\end{array}\right]\right\} \\
\textsc{sem} & \text{E1} : (\text{Sem1} \cup \text{Sem2} \cup \left\{\text{role(E1,across,Crossed)}\right\}) \\
\textsc{arg0} & \text{E1} \\
\textsc{arg1} & \text{Crosser} \\
\textsc{arg2} & \text{Crossed}
\end{array}\right]$$

$$(7.62)\left[\begin{array}{ll}
\textsc{ortho} & \text{cruzó} \\
\textsc{cat} & \text{s/NP} \\
\textsc{sem} & \text{E2} : \left\{\text{cruzar(E2), role(E2,agt,Crosser), role(E2,pat,Crossed)}\right\}
\end{array}\right]$$

The bilingual entry that puts these two together is:

$$(7.63)\left[\begin{array}{lll}
\textsc{spanish} & \boxed{7.61} & \left[\begin{array}{ll}\textsc{arg0} & \text{E} \\ \textsc{arg1} & \text{Crosser} \\ \textsc{arg2} & \text{Crossed}\end{array}\right] \\
\textsc{english} & \boxed{7.62} & \left[\begin{array}{ll}\textsc{arg0} & \text{E} \\ \textsc{arg1} & \text{Crosser} \\ \textsc{arg2} & \text{Crossed}\end{array}\right]
\end{array}\right]$$

A similar pair of monolingual entries, together with the bilingual entry to put them into correspondence, is needed for *swam-nadando*.

$$(7.64)\left[\begin{array}{ll}
\textsc{ortho} & \text{swam} \\
\textsc{cat} & \text{s}/\left[\begin{array}{ll}\textsc{cat} & \text{NP:Swimmer} \\ \textsc{sem} & \boxed{1}\ \text{E3} : \left\{\begin{array}{l}\text{swimming(E3),} \\ \text{role(E3,agt,Swimmer)}\end{array}\right\}\end{array}\right] \\
\textsc{sem} & \boxed{1} \\
\textsc{arg0} & \text{E3} \\
\textsc{arg1} & \text{Swimmer}
\end{array}\right]$$

$$(7.65)\begin{bmatrix} \textsc{ortho} & \text{nadando} \\ \textsc{cat} & \text{s}/\begin{bmatrix} \textsc{cat} & \text{S} \\ \textsc{sem} & \text{E4:Sem} \\ \textsc{arg0} & \text{E4} \\ \textsc{arg1} & \text{Swimmer} \end{bmatrix} \\ \textsc{sem} & \text{E4} : (\{\text{swimming(E4)}\} \cup \text{Sem}) \\ \textsc{arg0} & \text{E4} \\ \textsc{arg1} & \text{Swimmer} \end{bmatrix}$$

$$(7.66)\begin{bmatrix} \textsc{spanish} & \boxed{7.64}\begin{bmatrix} \textsc{arg0} & \text{E} \\ \textsc{arg1} & \text{Swimmer} \end{bmatrix} \\ \textsc{english} & \boxed{7.65}\begin{bmatrix} \textsc{arg0} & \text{E} \\ \textsc{arg1} & \text{Swimmer} \end{bmatrix} \end{bmatrix}$$

Although the lexical entries are fairly simple, it is appropriate here to follow the translation process through to see in detail how the various indices become unified.

Let us assume that we are translating the following sentence from English to Spanish:

(7.67) Mary swam across the river.

Some of the constituents produced by the parse, which give us the required unifications, are the following:

$$(7.68)\begin{bmatrix} \textsc{ortho} & \text{'across the river'} \\ \textsc{cat} & \text{s}/\begin{bmatrix} \textsc{cat} & \text{s} \\ \textsc{sem} & \text{E1:Sem2} \\ \textsc{arg0} & \text{E1} \\ \textsc{arg1} & \text{Crosser} \end{bmatrix} \\ \textsc{sem} & \text{E1} : \{\text{crossing(E1),arg(E1,across,R),river(R),definite(R)}\} \\ \textsc{arg0} & \text{E1} \\ \textsc{arg1} & \text{Crosser} \\ \textsc{arg2} & \text{R} \end{bmatrix}$$

This is basically a sentence modifier, which modifies the following sentence:

$$(7.69)\begin{bmatrix} \textsc{ortho} & \text{'Mary swam'} \\ \textsc{cat} & \text{s} \\ \textsc{sem} & \text{E} : \{\text{swimming(E),arg(E,agt,F3),name(F3,mary)}\} \\ \textsc{arg0} & \text{E} \\ \textsc{arg1} & \text{F3} \end{bmatrix}$$

This sign may be interpreted by saying that *Mary*, or rather its index *F3*, is the agent of the event *E*, which is a swimming event. *E* and *E1* (a crossing event) then become unified, and *F3* (Mary) then gets also identified with the Crosser variable in sign 7.68. The result is that *E* ends up being both a crossing and a swimming event, with *Mary* as its agent and *the river* as the entity being crossed (this being a special role introduced by the PP *across the river*:

$$(7.70)\begin{bmatrix} \textsc{ortho} & \text{'Mary swam across the river'} \\ \textsc{cat} & \text{s} \\ \textsc{sem} & \text{E} : \left\{ \begin{array}{l} \text{swimming(E),arg(E,agt,F3),name(F3,mary)} \\ \text{crossing(E),arg(E,crossed,R)} \\ \text{river(R),definite(R)} \end{array} \right\} \\ \textsc{arg0} & \text{E} \\ \textsc{arg1} & \text{F3} \\ \textsc{arg2} & \text{R} \end{bmatrix}$$

After Skolemisation (which we shall represent by using lower-case names for the Skolemised constants), bilingual lookup, and instantiating the relevant indices in the semantics, we end up with the following Spanish signs:

$$(7.71)\begin{bmatrix} \textsc{ortho} & \text{'María'} \\ \textsc{cat} & \text{s/}\begin{bmatrix} \text{s} \\ \text{e : Sem} \end{bmatrix} \\ \textsc{sem} & \text{e} : (\{\text{role(e,agt,f3),name(f3,maria)}\} \cup \text{Sem}) \\ \textsc{arg0} & \text{f3} \end{bmatrix}$$

$$(7.72)\begin{bmatrix} \textsc{ortho} & \text{cruzó} \\ \textsc{cat} & \text{s/NP} \\ \textsc{sem} & \text{e} : \{\text{cruzar(e), role(e,agt,f3), role(e,pat,r)}\} \end{bmatrix}$$

$$(7.73)\begin{bmatrix} \textsc{ortho} & \text{'el río'} \\ \textsc{cat} & \text{s}/\begin{bmatrix}\text{s}\\ \text{e:Sem2}\end{bmatrix} \\ \textsc{sem} & \text{e:}\{\text{role(e,pat,r),rio(r),definite(r)}\} \cup \text{Sem2} \\ \textsc{arg0} & \text{f3} \end{bmatrix}$$

(This sign was put together from lexical entries for *el* and *río*).

$$(7.74)\begin{bmatrix} \textsc{ortho} & \text{nadando} \\ \textsc{cat} & \text{s}/\begin{bmatrix}\textsc{cat} & \text{s}\\ \textsc{sem} & \text{e:Sem2}\\ \textsc{arg0} & \text{e}\\ \textsc{arg1} & \text{f3}\end{bmatrix} \\ \textsc{sem} & \text{e:}\{\text{nadar(e)}\} \cup \text{Sem2} \\ \textsc{arg0} & \text{e} \\ \textsc{arg1} & \text{m} \end{bmatrix}$$

These signs can be baked together to make the following constituents:

$$(7.75)\begin{bmatrix} \textsc{ortho} & \text{'cruzó el río'} \\ \textsc{cat} & \text{s} \\ \textsc{sem} & \text{e:}\left\{\begin{array}{l}\text{cruzar(e), role(e,agt,f3), role(e,pat,r)}\\ \text{rio(r),definite(r)}\end{array}\right\} \end{bmatrix}$$

$$(7.76)\begin{bmatrix} \textsc{ortho} & \text{'María cruzó el río'} \\ \textsc{cat} & \text{s} \\ \textsc{sem} & \text{e:}\left\{\begin{array}{l}\text{cruzar(e), role(e,agt,f3), role(e,pat,r)}\\ \text{rio(r),definite(r),name(f3,maria)}\end{array}\right\} \end{bmatrix}$$

$$(7.77)\begin{bmatrix} \textsc{ortho} & \text{'María cruzó el río nadando'} \\ \textsc{cat} & \text{s} \\ \textsc{sem} & \text{e:}\left\{\begin{array}{l}\text{cruzar(e), role(e,agt,f3), role(e,pat,r)}\\ \text{rio(r),definite(r),name(f3,maria),nadar(e)}\end{array}\right\} \end{bmatrix}$$

We then end up with an event e which is a *nadar* event as well as a *cruzar* event, of which *María* (*f3*) is the agent, and which has a patient role, r, which stands for *el río.*

It should be pointed out at this stage that the person, number and tense features of the verb have not been put into correspondence: clearly, the 3rd person singular feature of *cruzó* should correspond to the same feature of *swims*. This could be done by means of a morphological component to the grammars, which is beyond the scope of this work.

Chapter 8

Conclusion

8.1 Contribution

We have seen how lexically-driven Machine Translation solves many of the problems that exist with current methods such as Transfer methods and Isomorphic Grammars. In particular, such an approach makes it possible to write modern, unification-based monolingual grammars with great independence from each other, and to put them into correspondence by means of a bilingual lexicon of a similar degree of complexity as one might expect to find in a commonly available bilingual dictionary.

I hope to have demonstrated these points by constructing two monolingual Unification Categorial Grammars for small fragments Spanish and English, which nevertheless included some linguistically interesting phenomena. They were written independently, and with purely monolingual considerations in mind, which led to some noticeable differences in the grammar design. The grammars were put into correspondence by means of a bilingual lexicon, and algorithms for parsing, doing bilingual lookup and generation were suggested, which together constitute what has been named **Shake-and-Bake Translation**.

This consists of parsing the Source Language in any usual way, then looking up the words in the bilingual lexicon, and finally generating from the set of translations of these words, but allowing the Target Language grammar to instantiate the relative word ordering, taking advantage of the fact that the parse produces lexical and phrasal signs which are highly constrained (specifically in the semantics). The main algorithm presented was a variation on the CKY one.

The modularity demonstrated in the grammar design (both within each individual grammar, and by the use of lexical redundancy rules and templates within the monolingual grammars) means that the tasks involved may be more easily distributed between several teams of computational linguists, and that the monolingual grammars, developed without any consideration of other languages, may be re-usable in a multi-lingual translation system. The bilingual lexicon contains all the information specific to the language pair, and nothing but that. This would seem to be the optimal distribution of information, both from the point of view of building a larger, possibly multi-lingual system in the first place, and for maintaining and extending it. A further point that should be noticed is that nothing in the description of the translation process is specific to a single direction of translation: the grammars may be used both for parsing and generation, and the entries in a bilingual lexicon can be looked up either way round. The resulting system is therefore completely reversible. This is an advantage that is brought about by the clear distinction between procedural and declarative knowledge, and which is encouraged by modern unification-based theories of grammar.

8.2 Further research

Several issues presented here require further research. The grammar coverage is of course very limited, and if large grammars and lexicons were to be used, more efficient ways of indexing the entries are required, since at the moment the whole of the lexicon is held in Prolog's working memory, an approach that is unlikely to be feasible for larger systems.

Morphology is currently handled in a very sketchy manner, with lexical redundancy rules producing a multiplicity of lexical entries many of which should be properly covered by means of a morphological processor. Used during parsing, this component would map the input string into individual morphemes, which are contained in the monolingual lexicons, and are put into correspondence by means of the bilingual lexicon. The process should then be reversed in order to produce the target string from the TL sequence of morphemes.

The semantic representations are also somewhat crude, since they fail to account properly for any quantification and temporal and aspectual information, or of any discourse effects and the way in which these interact with word order. However, the handle exists onto which more sophisticated semantic representations can be attached, and the unification-based mechanisms and notation provided is powerful enough to represent such information, provided we had an adequate theory for these aspects of semantics.

Finally the generation algorithm is somewhat inefficient, and the possibility of improving it along the lines suggested in Subsection 6.1.3, namely driving the generation by attempting to identify the "heads" and working from them, should be explored further.

Nevertheless, I hope to have shown that pursuing further research in this direction is a worthwhile aim, and one likely to result in usable Machine Translation tools in the not too distant future.

Bibliography

[Appelo & Landsbergen 86] Appelo, L., and Landsbergen, J. *The Machine Translation Project ROSETTA*. Philips Research M.S. 13.801. Presented at the First International Conference on the State of the Art in Machine Translation, Saarbruecken, August 1986.

[Beaven 87] Beaven, J.L. *Machine Translation between English and Spanish using logically isomorphic Unification Categorial Grammars*, MSc Dissertation, Department of Artificial Intelligence, University of Edinburgh, 1987.

[Beaven 90] Beaven, J.L. A Unification-Based Treatment of Spanish Clitics, in Engdahl, E., Reape, M., Mellor, M. and Cooper, R. (eds.) *Edinburgh Working Papers in Cognitive Science, Volume 6. Parametric Variation in Germanic and Romance: Proceedings from a DYANA Workshop, September 1989.* Centre for Cognitive Science, Edinburgh, 1990.

[Beaven and Whitelock 88] Beaven, J.L., and Whitelock, P.J. Machine Translation Using Isomorphic UCGs, in *Proceedings of the 12th International Conference on Computational Linguistics (COLING 88)*, pages 32–35. Budapest, 1988.

[Bello 84] Bello, A. *Gramática de la lengua castellana.*, EDAF, Madrid, 1984.

[Bès & Gardent 89] Bès. G. G. & Gardent, C. French Order Without Order, in *Proceedings of the Fourth Conference of the European Chapter of the Associacion of Computational Linguistics*, pages 249–255, Manchester, 10-12 April 1989.

[Boitet et al. 85] Boitet, Ch., Guillaume, P., and Quézel-Ambrunaz. A case study in software evolution: from ARIANE-78.4 to ARIANE-85. In *Proceedings of the Conference on Theoretical and Methodological Issues in Machine Translation of Natural Languages*, Colgate University, Hamilton, NY, August 1985, pages 27–58.

[Brew 92] Brew, C. Letting the cat out of the bag: generation for Shake-and-Bake MT. To appear in *Proceedings of the 14th International Conference on Computational Linguistics (COLING 92)*, Nantes, 1992.

[Calder et al. 88] Calder, J., Klein, E. and Zeevat, H. Unification Categorial Grammar — A Concise, Extendable Grammar for Natural Language Processing. In *Proceedings of the 12th International Conference on Computational Linguistics (COLING 88)*, pages 83–86, Budapest, 1988.

[Chomsky 86] Chomsky, N. *Knowledge of Language. Its Nature, Origin and Use.* Praeger, New York, 1986.

[Davidson 67] Davidson, D. The Logical Form of Action Sentences. In Rescher, N. (ed.), *The Logic of Decision and Action.* Pittsburgh: University of Pittsburgh Press, 1967.

[Dowty et al. 81] Dowty, D.R., Wall, R.E. and Peters, S *Introduction to Montague Semantics*, D. Reidel, Dordrecht, 1981.

[Dowty 88] Dowty, D.R. Type Raising, Functional Composition, and Non-Constituent Conjunction. In Oehrle, R., Bach, E. and Wheeler, D. (eds.) *Categorial Grammars and Natural Language Structures*, pages 153–197, D. Reidel, Dordrecht, 1988.

[Dowty 89] Dowty, D. On the Semantic Content of Notion "Thematic Role". In Chierchia, G., Partee, B. and Turner, R. (eds.) *Property Theory, Type Theory and Natural Language Semantics.* Dordrecht: D. Reidel, 1989.

[Foster 90] Foster, J. *A Theory of Word Order in Categorial Grammar with Special Reference to Spanish*, D. Phil Thesis, University of York, Department of Language and Linguistic Science, 1990.

[Garey and Johnson 79] Garey, M.J., and Johnson, D.S. (1979) *Computers and Intractability: A Guide to the Theory of NP-Completeness.* W.H. Freeman & Co, New York, 1979.

[Gazdar & Mellish 89] Gazdar, G. and Mellish, C. *Natural Language Processing in PROLOG*, Addison-Wesley, 1989.

[Gili y Gaya 69] Gili y Gaya, S. *Curso Superior de Sintaxis Española.* Bibliograf, S.A., Barcelona, 1969.

[Jaeggli 86] Jaeggli, O.A. Three Issues in the Theory of Clitics: Case, Doubled NPs, and Extraction. In Borer, H. (ed.) *Syntax and Semantics, Vol. 19: the Syntax of Pronominal Clitics.* Academic Press, 1986.

[Hewson 81] Hewson, J. (1981) *More on Spanish selo.* Linguistics 19, pages 439–447, 1981.

[Hutchins 78] Hutchins, W.J. Progress in Documentation: Machine Translation and Machine-Aided Translation. *Journal of Documentation* Volume 34,2. Pages 119-159. 1978.

[Kamp 81] Kamp, H. A theory of truth and semantic representation. In Groenenijk, J. A. G., Janssen, T. M. V. and Stokhof, M. B. J. (eds.) *Formal Methods in the Study of Language*, Volume 136, pages 227–322. Amsterdam: Mathematical Centre Tracts, 1981.

[Kaplan et al. 89] Kaplan, R.M., Netter, K., Wedekind, J. and Zaenen, A. (1989). Translation by Structural Correspondences. In *Proceedings of the Fourth Conference of the European Chapter of the Association for Computational Linguistics*, pages 272–281, Manchester, 10-12 April 1989.

[Klein & Sag 85] Klein, E and Sag, I. Type-Driven Translation, *Linguistics and Philosophy*, Vol. 8, pages 163–201, 1985.

[Landsbergen 87a] Landsbergen, J. Isomorphic Grammars and their Use in the ROSETTA Translation System. In King, M. (ed.) *Machine Translation Today: the State of the Art: Proceedings of the Third Lugano Tutorial, Lugano, Switzerland, 2-7 April 1984.* Edinburgh University Press, 1987. Pages 351–372.

[Landsbergen 87b] Landsbergen, J. Montague Grammar and Machine Translation. In Whitelock, P.J., Wood, M.M., Somers, H., Bennett, P. and Johnson, R. (eds.) (1987) *Linguistic Theory and Computer Applications.* Academic Press, 1987.

[Maas 87] Maas, H.D. The MT system SUSY. In King, M. (ed.) *Machine Translation Today: the State of the Art: Proceedings of the Third Lugano Tutorial, Lugano, Switzerland, 2-7 April 1984*. Edinburgh University Press, 1987. Pages 209–246.

[Malevaje 88] Malevaje. Es que hueles a pellejo. In *Un Momentito* (LP record). 3 Cipreses, Madrid, 1988.

[Manandhar forthcoming] , Manandhar, S. *A Unification Based Framework for Lexical Knowledge Representation*, PhD Thesis, University of Edinburgh, (forthcoming).

[Marcos Marín 80] Marcos Marín, F. *Curso de Gramática Española.* Editorial Cincel, S.A., Madrid, 1980.

[Mellish 88] Mellish, C. Implementing Systemic Classification by Unification, in *Computational Linguistics*, Vol 14 No 1, Winter 1988, pages 40–51, 1988.

[Moortgat 88] Moortgat, M. Mixed Composition and Discontinuous Dependencies. In Oehrle, R., Bach, E. and Wheeler, D. (eds.) *Categorial Grammars and Natural Language Structures*, pages 319–348, D. Reidel, Dordrecht, 1988.

[Pereira & Shieber 87] Pereira, F.C.N. and Shieber, S.I. *Prolog and Natural-Language Analysis*, CSLI Lecture Notes, Stanford, 1987.

[Pollard and Sag 87] Pollard, C. and Sag, I.A. *Information-Based Syntax and Semantics - Volume 1: Fundamentals.* Lecture Notes Number 13. Center for the Study of Language and Information, Stanford University, 1987.

[Popowich 88] Popowich, F.P. *Reflexives and tree unification grammar*, PhD Thesis, University of Edinburgh, 1988.

[Reape 89] Reape, M. A logical treatment of semi-free word order and bounded discontinuous constituency. In *Proceedings of the Fourth Conference of the European Chapter of the Associacion of Computational Linguistics*, pages 103–110, Manchester, 10-12 April 1989.

[Reape 90] Reape, M. Parsing semi-free word order and bounded discontinuous constituency: generalizations of the shift-reduce and CKY algorithms. Paper presented at the First Computational Linguistics in the Netherlands Day, Rijksuniversiteit Utrecht, Utrecht, The Netherlands, 26 October 1990.

[Reape forthcoming] Reape, M. Parsing semi-free word order and bounded discontinuous constituency and "shake 'n' bake" machine translation (or 'generation as parsing'). To appear in Emele, M., Heid, U., Momma S. and Zajac, R. (eds) *Proceedings of the Workshop on Constraint-based Approaches to Natural Language Generation.* Bad Teinach, Germany, forthcoming.

[Rounds 89] Rounds, W.C. *Set Values for Unification-Based Grammar Formalisms and Logic Programming*, Draft, Electric Engineering and Computer Science Department, University of Michigan, and CSLI, Stanford University and Xerox PARC, 1989.

[Rupp 86] Rupp, CJ. *Machine Translation between German and English using Logically Isomorphic Grammars.* MSc Dissertation, University of Sussex, 1986.

[Sadler and Thompson 91] Sadler, L. and Thompson, H.S. (1991). Structural Non-Correspondence in Translation. In *Proceedings of the 5th Conference of the European Chapter of the Association for Computational Linguistics*, pages 293–298, Berlin 9-11 April 1991.

[Sanfilippo 90a] Sanfilippo, A. Thematic Accessibility in Discontinuous Dependencies. In *Proceedings of the Symposium on Discontinuous Constituency*, Institute for Language Technology and Artificial Intelligence, Tilburg, The Netherlands, 25–27 Jan 1990.

[Sanfilippo 90b] Sanfilippo, A. Clitic Doubling and Dislocation in Italian in Engdahl, E., Reape, M., Mellor, M. and Cooper, R. (eds.) *Edinburgh Working Papers in Cognitive Science, Volume 6. Parametric Variation in Germanic and Romance: Proceedings from a DYANA Workshop, September 1989.* Centre for Cognitive Science, Edinburgh, 1990.

[Sanfilippo 90c] Sanfilippo, A. Inversion, Dislocation and the Null-Subject Parameter, in Bès, G. (ed.) *Proceedings of the Workshop on Categorial Grammar and Word Order, Clermont Ferrand, May 1990.* Université Blaise Pascal, Clermont Ferrand.

[Shieber 86] Shieber, S. *An Introduction to Unification-based Approaches to Grammar.* Lecture Notes Number 4. Center for the Study of Language and Information, Stanford University, 1986.

[Steedman 85] Steedman, M.J. Dependency and Coordination in the Grammar of Dutch and English, *Language*,61,3, pages 523–568, September 1985.

[Steedman 87] Steedman, M.J. Combinatory Grammars and Parasitic Gaps, *Natural Language and Linguistic Theory*,5,4, 1987.

[Steedman 90] Steedman, M.J. Syntax and intonational structure in a combinatory grammar. In Altmann, G.T.M. (Ed.), *Cognitive Models of Speech Processing: Psycholinguistic and Computational Perspectives.* Cambridge, Mass.: MIT Press, pages 457–482, 1990.

[Suñer 88] Suñer, M. The Role of Agreement in Clitic-Doubled Constructions, *Natural Language and Linguistic Theory*, Vol. 6, pages 391–434, 1988.

[Uszkoreit 86] Uszkoreit, H. (1986). Categorial Unification Grammars. *Proceedings of the 11th International Conference on Computational Linguistics (COLING 86)*, pages 187–194, Bonn, 1986.

[van de Veen 90] van de Veen, E. *Discontinuous Constituency and Parsing*, MSc thesis, Department of Artificial Intelligence, University of Edinburgh, 1990.

[Vergnaud 82] Vergnaud, J.-R. *Dépendances et Niveaux de Représentation en Syntaxe,* Thèse de Doctorat d'Etat, Université de Paris VII, 1982.

[Weaver 49] . Weaver, W. *Translation*, Unpublished memorandum, New York.

[Whitelock 88] Whitelock, P. *A Feature-based Categorial Morpho-Syntax for Japanese.* DAI research paper no 324, Dept. of Artificial Intelligence, Univ. of Edinburgh. Also in Rohre, C., and Reyle, U. (eds.) *Natural Language Parsing and Linguistic Theories.* D.Reidel, Dordrecht, 1988.

[Whitelock 91] Whitelock, P. Shake and Bake Translation. Unpublished Draft, 1991.

[Zeevat et al. 87] Zeevat, H., Klein, E. and Calder, J. An Introduction to Unification Categorial Grammar. In Haddock, N.J., Klein, E. and Morrill, G. (eds.): *Edinburgh Working Papers in Cognitive Science, volume I: Categorial Grammar, Unification Grammar and Parsing.*, pages 195–222, Edinburgh, 1987.

Appendix A

Sample run

This appendix contains a small sample run of the toy Machine Translation system written, in order to demonstrate its capabilities. It is shown running under SICStus Prolog, version 0.7, on a SUN SPARCstation SLC.

A.1 Viewing controls

The amount of information shown to the user is controlled with the `view` predicate, used for determining and modifying what information is output to the screen. By default, only the source string, the lexical entries and compounds asserted, and the target string are shown.

```
| ?- view.
The current view levels are: no_parse src no_tgt bake no_trans no_trees
To change any of them, use view(Item), where Item is one of:
parse    : see the parse tree
no_parse : do not see the parse tree
src      : see the source words
no_src   : do not see the source words
tgt      : see the target words
no_tgt   : do not see the target words
bake     : see the edges being baked
no_bake  : do not see the edges being baked
trees    : see the full trees
no_trees : see the derivations only
trans    : see the translation
no_trans : do not see the translation

yes
```

```
| ?- translate(['Maria',canta]).
Source:
Maria canta

Source language words for this parse:

[[lex,Maria],
 [lex,canta]]

Asserting lexical entry: [lex,Mary]
Asserting lexical entry: [lex,sings]
Asserting compound: [back,[lex,Mary],[lex,sings]]
Translation string:
Mary sings

no
| ?- view(no_src).
Not showing source words

yes
| ?- view(no_bake).
Not showing edges being baked

yes
| ?- translate(['Maria',canta]).
Source:
Maria canta

Translation string:
Mary sings

no
| ?-
```

A.2 Reversibility

The program assumes the translation is between English and Spanish, finds out from the input which is the Source Language, and produces its output in the other language. The same predicate `translate` therefore works in both directions:

```
| ?- translate(['Mary',sings]).
Source:
Mary sings

Translation string:
Maria canta

no

| ?-
```

A.3 Argument switching and head switching

Going back to the default mode, the following example shows how the examples of argument and head switching mentioned in Chapter 7 are handled.

```
| ?- translate(['Juan',gusta,a_Maria]).
Source:
Juan gusta a_Maria

Source language words for this parse:

[[lex,Juan],
 [lex,gusta],
 [lex,a_Maria]]

Asserting lexical entry: [lex,John]
Asserting lexical entry: [lex,likes]
Asserting lexical entry: [lex,to_Mary]
Asserting compound: [fwd,[lex,likes],[lex,John]]
Asserting lexical entry: [lex,John]
Asserting lexical entry: [lex,likes]
Asserting lexical entry: [lex,Mary]
Asserting compound: [fwd,[lex,likes],[lex,John]]
Asserting compound: [back,[lex,Mary],[fwd,[lex,likes],[lex,John]]]
Translation string:
Mary likes John


no
```

```
| ?- translate(['Maria',cruzo,el_rio,nadando]).
Source:
Maria cruzo el_rio nadando

Source language words for this parse:

[[lex,Maria],
 [lex,cruzo],
 [lex,el_rio],
 [lex,nadando]]


Asserting lexical entry: [lex,Mary]
Asserting lexical entry: [lex,across]
Asserting lexical entry: [lex,the_river]
Asserting lexical entry: [lex,swam]
Asserting compound: [back,[lex,Mary],[lex,swam]]
Asserting compound: [fwd,[lex,across],[lex,the_river]]
Asserting compound:
[back,[back,[lex,Mary],[lex,swam]],[fwd,[lex,across],[lex,th
e_river]]]
Translation string:
Mary swam across the_river

no
| ?-
```

Appendix B

Spanish entries

This appendix contains the code for sample Spanish entries. These entries are built from three files: `temp_sp.pl` contains the basic templates used for the lexical entries, `lex_sp.pl` the lexical entries themselves, and `lrul_sp.pl` the lexical redundancy rules that allow us to expand the lexicon as built from the basic entries and the templates.

The standard implementation of graph unification using Prolog terms is based on Prattle, a program originally written by Mark Johnson.

B.1 Templates

```
% Title: temp_sp.pl
% Description: Templates for a Spanish grammar illustrating
% illustrating the contents of Chapter 3 of my thesis.
%

% Author: John Beaven
% Last Update: 4 Sept 1991

% This file contains the basic templates used for building the lexical
% entries in lex_sp.pl

% Format:
% template(Sign, Template_Name, Parameter_List, Goals).
% where Sign is the whole feature-structure, Template_Name is a
% unique identifier for that particular template, Parameter_List
% are parameters passed from the lexical entry, and Goals have to
% be satisfied.

% Format of cases feature:
% c(Nom,Acc,Dat,Prep,List_of_Preps)
```

```
% where the first 4 arguments are boolean: either 0 (not allowed) or
% (instead of 1, for better legibility) nom, obj, gen, prep respectively.
% If more then one case is possible, anonymous variables are used
% in the places of the possible cases.


% Names (NP) - Type 1: look forward for (vfinal = -) sentence.
% Human, so on their own must be nominative.
template(X,name_1_s,[Deriv,Pers,Num,Gen,SemInd,Pred],
        [X:derivation = Deriv,
         X:cat = s,
         X:slash = [Y],
                Y:cat = s,
                Y:slash = [],
                Y:order = fwd,
                Y:feats = Feats,
                Y:vfinal = -,
                Y:cases = set([Case | OtherCases]),
                        Case:agr:person = Pers,
                        Case:agr:number = Num,
                        Case:agr:gender = Gen,
                        Case:agr:case = c(nom,0,0,0,0),
                        Case:index = SemInd,
                Y:sem:index = I,
                Y:sem:conj = Sem,
         X:cases = set(OtherCases),
         X:feats = Feats,
                Feats:fin = +,
                X:sem:index = I,
         X:sem:conj = [Pred|Sem],
         X:sem:indices = [SemInd]
        ]).


% Nouns - Type 1: Human,
% hence nominative once they combine with a Determiner.
template(X,noun_1_s,[Deriv,Pers,Num,Gen,SemInd,Pred],
        [X:derivation = Deriv,
         X:cat = noun,
         X:slash = [],
         X:cases = set([Case]),
                Case:agr:person = Pers,
                Case:agr:number = Num,
                Case:agr:gender = Gen,
                Case:agr:case = c(nom,0,0,0,0),
                Case:index = SemInd,
```

```
                        SemInd = type(_,[human]),
         X:sem:index = SemInd,
         X:sem:conj = [Pred|_Sem],
         X:sem:indices = [SemInd]
        ]).

% Nouns - Type 2: Inanimate  - nominative or accusative
% when combined with a Determiner.
template(X,noun_2_s,[Deriv,Pers,Num,Gen,SemInd,Pred],
        [X:derivation = Deriv,
         X:cat = noun,
         X:slash = [],
         X:cases = set([Case]),
                Case:agr:person = Pers,
                Case:agr:number = Num,
                Case:agr:gender = Gen,
                Case:agr:case = c(_Nom,_Acc,0,0,0),
                Case:index = SemInd,
                      SemInd = type(_,[inanimate]),
         X:sem:index = SemInd,
         X:sem:conj = [Pred|_Sem],
         X:sem:indices = [SemInd]
        ]).

% Determiners - Type 1
template(X,det_1_s,[Deriv,Pers,Num,Gen,SemInd,Pred],
        [X:derivation = Deriv,
         X:cat = s,
         X:slash = [Y1,Y2],
                Y1:cat = noun,
                Y1:slash = [],
                Y1:order = fwd,
                Y1:cases = set([Case]),
                Y1:sem:index = SemInd,
                Y1:sem:conj = [Pred1],

                Y2:cat = s,
                Y2:slash = [],
                Y2:order = fwd,
                Y2:feats = Feats,
                Y2:vfinal = -,
                Y2:cases = set([Case | OtherCases]),
                        Case:agr:person = Pers,
                        Case:agr:number = Num,
                        Case:agr:gender = Gen,
                        Case:agr:case = c(_Nom,_Acc,0,0,0),
```

```
                        Case:index = SemInd,
                Y2:sem:index = I,
                Y2:sem:conj = Sem2,

         X:cases = set(OtherCases),
         X:feats = Feats,
                X:sem:index = I,
         X:sem:conj = [Pred,Pred1|Sem2],
         X:sem:indices = [SemInd]
        ]).


% Intransitive Verbs - Type 1 (active, finite).
template(X,intrans_1_s,[Deriv,Pers,Num,Vform,Pred,SemInd,Agt],
        [X:derivation = Deriv,
         X:cat = s,
         X:slash = [],
         X:feats = Feats,
                Feats:fin = +,
                Feats:vform = Vform,
         X:cases = set([Nom]),
                Nom:agr:number = Num,
                Nom:agr:person = Pers,
                Nom:agr:case = c(nom,0,0,0,0),
                Nom:index = Agt,
        X:sem:index = SemInd,
        X:sem:conj = [Pred,role(SemInd,agt,Agt)],
        X:sem:index = SemInd,
        X:sem:indices = [SemInd,Agt]
        ]).

% Intransitive non-finite Verbs - Type 1 (active).
% These are just like intransitive tensed verbs (a.k.a. sentences),
% but they are not sentences since they are not in finite form.
template(X,intrans_nf_1_s,[Deriv,Vform,Pred,SemInd,Agt],
        [X:derivation = Deriv,
         X:cat = vp,
         X:slash = [],
         X:feats = Feats,
                Feats:fin = -,
                Feats:vform = Vform,
         X:cases = set([Nom]),
                Nom:agr:case = c(nom,0,0,0,0),
                Nom:index = Agt,
         X:sem:index = SemInd,
         X:sem:conj = [Pred,role(SemInd,agt,Agt)],
```

```
         X:sem:index = SemInd,
         X:sem:indices = [SemInd,Agt]
        ]).


% Transitive Verbs - Type 1 (Active, Finite)
template(X,trans_1_s,[Deriv,Pers,Num,Vform,Pred,SemInd,Agt,Pat],
        [X:derivation = Deriv,
         X:cat = s,
         X:slash = [Y],
                Y:cat = s,
                Y:slash = [Z],
                  Z:cat = s,
                  Z:slash = [],
                  Z:order = fwd,
                  Z:feats = Feats,
                  Z:cases = set([Acc]),
                        Acc:agr:case = c(0,acc,0,0,0),
                        Acc:index = Pat,
                   Z:sem:index = SemInd,
                   Z:sem:conj = [Pred, role(SemInd,agt,Agt),
                                 role(SemInd,pat,Pat)],
                Y:order = fwd,
                Y:vfinal = -,
                Y:cases = set(MoreCases),
                Y:sem:index = SemInd,
                Y:sem:conj = Sem3,
         X:feats = Feats,
                Feats:fin = +,
                Feats:vform=Vform,
         X:cases = set([Nom | MoreCases]),
                Nom:agr:person = Pers,
                Nom:agr:number = Num,
                Nom:agr:case = c(nom,0,0,0,0),
                Nom:index = Agt,
         X:sem:index = SemInd,
         X:sem:conj = Sem3,
         X:sem:index = SemInd,
         X:sem:indices = [SemInd,Agt,Pat]
        ]).


% Ditransitive Verbs - Type 1 (Active, Finite, Object + a)
template(X,ditrans_1_s,[Deriv,Pers,Num,Vform,Pred,SemInd,Agt,Pat,Goal],
        [X:derivation = Deriv,
         X:cat = s,
```

```
X:slash = [Y1, Y2],
       Y1:cat = s,
       Y1:slash = [Z1],
         Z1:cat = s,
         Z1:slash = [],
         Z1:order = fwd,
         Z1:feats = Feats,
         Z1:cases = set([Acc]),
               Acc:agr:case = c(0,acc,0,0,0),
               Acc:index = Pat,
         Z1:sem:index = SemInd,
         Z1:sem:conj = [Pred,
                        role(SemInd,agt,Agt),
                        role(SemInd,pat,Pat),
                        role(SemInd,goal,Goal)],
       Y1:order = fwd,
       Y1:vfinal = -,
       Y1:cases = set(MoreCases1),
       Y1:sem:index = SemInd,
       Y1:sem:conj = Sem1,

           Y2:cat = s,
       Y2:slash = [Z2],
         Z2:cat = s,
         Z2:slash = [],
         Z2:order = fwd,
         Z2:feats = Feats,
         Z2:cases = set([Dat|MoreCases1]),
                Dat:agr:case = c(0,0,dat,0,0),
               Dat:index = Goal,
         Z2:sem:index = SemInd,
         Z2:sem:conj = Sem1,
       Y2:order = fwd,
       Y2:vfinal = -,
       Y2:cases = set(MoreCases2),
       Y2:sem:index = SemInd,
       Y2:sem:conj = Sem2,

X:feats = Feats,
       Feats:fin = +,
       Feats:vform = Vform,
X:cases = set([Nom | MoreCases2]),
       Nom:agr:person = Pers,
       Nom:agr:number = Num,
       Nom:agr:case = c(nom,0,0,0,0),
       Nom:index = Agt,
```

```
         X:sem:index = SemInd,
         X:sem:conj = Sem2,
         X:sem:index = SemInd,
         X:sem:indices = [SemInd,Agt,Pat,Goal]
        ]).



% Volition Verbs - Type 1 (active, finite, subcategorize for an
% infinitival sentence, subject control) eg quiere.
template(X,vol_1_s,[Deriv,Pers,Num,Vform,Pred,SemInd,Agt,Wanted],
        [X:derivation = Deriv,
         X:cat = s,
         X:slash = [Y],
                Y:cat = vp,
                Y:slash = [],
                Y:order = fwd,
                Y:feats:vform = inf,
                Y:cases = set([Nom|MoreCases]),
                Y:sem:index = Wanted,
                Y:sem:conj = Sem,

         X:feats = Feats,
                Feats:fin = +,
                Feats:vform = Vform,
         X:cases = set([Nom|MoreCases]),
                Nom:agr:person = Pers,
                Nom:agr:number = Num,
                Nom:agr:case = c(nom,0,0,0,0),
                Nom:index = Agt,
         X:sem:index = SemInd,
         X:sem:conj = [Pred,
                       role(SemInd,agt,Agt),
                       role(SemInd,pat,Wanted)|Sem],
         X:sem:index = SemInd,
         X:sem:indices = [SemInd,Agt,Wanted]
        ]).

% Adverb (post-sentential modifier).
template(X,adv_1_s,[Deriv,SemInd,Modification],
        [X:derivation = Deriv,
         X:cat = s,
         X:slash = [Y],
                Y:cat = s,
                Y:slash = [],
```

```
                Y:order = back,
                Y:feats = Feats,
                Y:cases = Cases,
                Y:sem:index = SemInd,
                Y:sem:conj = Sem,
         X:cases = Cases,
         X:feats = Feats,
                X:sem:index = SemInd,
         X:sem:conj = [Modification|Sem],
         X:sem:indices = Y:sem:indices
        ]).


% Case-absorbing clitic as it appears in chapter 3 of thesis
template(X,ca_clitic_1a_s,[Deriv,Pers,Num,Gen,Case],
        [X:derivation = Deriv,
         X:cat = s,
         X:slash = [Y],
                Y:cat = s,
                Y:slash = [],
                Y:feats = Feats3,
                Y:lex = +,
                Y:cases = set([Cse | OtherCases3]),
                        Cse:agr:person = Pers,
                        Cse:agr:number = Num,
                        Cse:agr:gender = Gen,
                        Cse:agr:case = Case,
                        Cse:index = I,
                Y:sem:index = I3,
                Y:sem:conj = Sem3,
         X:cases = set(OtherCases3),
         X:feats = Feats3,
         X:sem:index = I3,
         X:sem:conj = Sem3,
         X:sem:indices = [I]
        ]).

% Case-absorbing clitic,
% More complicated version, where the clitic subcategorizes for
% (is type-raised over) the (di)transitive verb. This version
% was not used in thesis, but I think it is better for getting
% all the vfinal mechanism working.

template(X,ca_clitic_1b_s,[Deriv,Pers,Num,Gen,Case],
        [X:derivation = Deriv,
         X:cat = s,
```

```
         X:slash = [A|Rest],
                A:cat = s,
                A:slash = [B|Rest],
                        B:cat = s,
                        B:slash = [C],
                                C:cat = s,
                                C:slash = [],
                                C:order = fwd,
                                C:feats = Feats1,
                                C:cases = set([Cse | MoreCases3]),
                                       Cse:agr:person = Pers,
                                       Cse:agr:number = Num,
                                       Cse:agr:gender = Gen,
                                       Cse:agr:case = Case,
                                       Cse:index = Ind,
                                C:sem:index = I,
                                C:sem:conj = Sem,
                        B:order = fwd,
                        B:feats = Feats2,
                        B:cases = set(MoreCases3),
                        B:sem:index = I,
                        B:sem:conj = Sem,
                A:order = fwd,
                A:feats = Feats1,
                        Feats1:fin = +,
                A:cases = Cases1,
                A:sem:index = I,
                A:sem:conj = Sem1,
         X:feats = Feats2,
         X:cases = Cases1,
         X:sem:index = I,
         X:sem:conj = Sem1,
         X:sem:indices = [Ind]
        ]).


% Non-case-absorbing clitic as in Chapter 3 of thesis
template(X,nca_clitic_1a_s,[Deriv,Pers,Num,Gen,Case],
        [X:derivation = Deriv,
         X:cat = s,
         X:slash = [Y],
                Y:cat = s,
                Y:slash =  [],
                Y:feats = Feats1,
                        Feats1:fin = +,
```

```
             Y:lex = +,
             Y:cases = Cases,
                     Cases = set([Cse | _OtherCases]),
                     Cse:agr:person = Pers,
                     Cse:agr:number = Num,
                     Cse:agr:gender = Gen,
                     Cse:agr:case = Case,
                     Cse:index = Z3,
             Y:sem:index = I5,
             Y:sem:conj = Sem5,
         X:order = back,
         X:feats = Feats1,
         X:cases = Cases,
         X:sem:index = I5,
         X:sem:conj = [role(I5,_R5,Z3) | Sem5],
         X:sem:indices = [Z3]
        ]).



% Non-Case-absorbing clitic
% More complicated version, where the clitic subcategorizes for
% the verb. This version was not used in thesis, but I think
% it is better for getting all the vfinal mechanism working.

template(X,nca_clitic_1b_s,[Deriv,Pers,Num,Gen,Case],
        [X:derivation = Deriv,
         X:cat = s,
         X:slash = [A|Rest],
             A:cat = s,
             A:slash = [B|Rest],
                     B:cat = s,
                     B:slash = [C],
                             C:cat = s,
                             C:slash = [],
                             C:order = fwd,
                             C:feats = Feats1,
                             C:cases = Cases,
                                    Cases = set([Cse | _MoreCases]),
                                    Cse:agr:person = Pers,
                                    Cse:agr:number = Num,
                                    Cse:agr:gender = Gen,
                                    Cse:agr:case = Case,
                                    Cse:index = Ind,
                             C:sem:index = I,
                             C:sem:conj = Sem,
```

```
                        B:order = fwd,
                        B:feats = Feats2,
                        B:cases = Cases,
                        B:sem:index = I,
                        B:sem:conj = Sem,
                A:order = fwd,
                A:feats = Feats1,
                        Feats1:fin = +,
                A:cases = Cases1,
                A:sem:index = I,
                A:sem:conj = Sem1,
         X:feats = Feats2,
         X:cases = Cases1,
         X:sem:index = I,
         X:sem:conj = Sem1,
         X:sem:indices = [Ind]
        ]).
```

B.2 Lexicon

```
% Title: lex_sp.pl
% Description: Lexical entries for a Spanish grammar
% illustrating the contents of Chapter 3 of my thesis.

% Most of these entries invoke templates from the file
% temp_sp.pl
% There are a few however for which templates do not exist, and are
% given here in full.

% Author: John Beaven
% Last Update: 4 Sept 1991

% Format:
% s(Sign,Language) --- Orthography, Goals
% where Sign is the representation of the attribute-feature structure,
% and Goals have to be satisfied.


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                 Spanish Monolingual lexical entries
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Names %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
s(X,spanish) --- 'Maria',
   [template(X,name_1_s,[[lex,'Maria'],3,sg,fem,F3,name(F3,maria)],_),
    F3 = type(_Ind_F3,[fem,sing])
   ].

s(X,spanish) --- 'Juan',
   [template(X,name_1_s,[[lex,'Juan'],3,sg,masc,M3,name(M3,juan)],_),
    M3 = type(_Ind_M3,[male,sing])
   ].

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%  Nouns %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% human

s(X,spanish) --- hombre,
   [template(X,noun_1_s,[[lex,hombre],3,sg,masc,X3,hombre(X3)],_),
    X3 = type(_Ind_X3,[male,sing])
   ].

% nonhuman

s(X,spanish) --- libro,
   [template(X,noun_2_s,[[lex,libro],3,sg,masc,X3,libro(X3)],_),
    X3 = type(_Ind_X3,[inanimate, sing])
   ].

s(X,spanish) --- rio,
   [template(X,noun_2_s,[[lex,rio],3,sg,masc,X3,[rio(X3)]],_),
    X3 = type(_Ind_X3,[inanimate, sing])
   ].


%%%%%%%%%%%%%%%%%%%%%%%%%% Intransitive Verbs %%%%%%%%%%%%%%%%%%%%%%%%

% Finite

s(X,spanish) --- canta,
   [template(X,intrans_1_s,[[lex, canta],3,sg,'3sg',cantar(E),E,Agt],_),
    Agt= type(_Ind_Agt,[animate]),
    E= type(_Ind_E,[event])
   ].

s(X,spanish) --- lava,
   [template(X,intrans_1_s,[[lex, lava],3,sg,'3sg',lava(E),E,Agt],_),
    Agt= type(_Ind_Agt,[animate]),
    E= type(_Ind_E,[event])
```

```
  ].


% Non-finite

s(X,spanish) --- cantar,
   [template(X,intrans_nf_1_s,[[lex, cantar],inf,cantar(E),E,Agt],_),
    Agt= type(_Ind_Agt,[animate]),
    E= type(_Ind_E,[event])
   ].

%%%%%%%%%%%%%%%%%%%%%%%%% Transitive Verbs %%%%%%%%%%%%%%%%%%%%%%%%%

s(X,spanish) --- leyo,
   [template(X,trans_1_s,[[lex, leyo],3,sg,'3sg',leer(E),E,Agt,Pat],_),
    Agt= type(_Ind_Agt,[human]),
    Pat= type(_Ind_Pat,[inanimate]),
    E= type(_Ind_E,[event])
   ].

s(X,spanish) --- vio,
   [template(X,trans_1_s,[[lex, vio],3,sg,'3sg',ver(E),E,Agt,Pat],_),
    Agt= type(_Ind_Agt,[human]),
    Pat= type(_Ind_Pat,[entity]),
    E= type(_Ind_E,[event])
   ].

s(X,spanish) --- aguanta,
   [template(X,
             trans_1_s,
             [[lex, aguanta],3,sg,'3sg',aguantar(E),E,Agt,Pat],
             _),
    Agt= type(_Ind_Agt,[human]),
    Pat= type(_Ind_Pat,[entity]),
    E= type(_Ind_E,[event])
   ].

s(X,spanish) --- cruzo,
   [template(X,
             trans_1_s,
             [[lex, cruzo],3,sg,'3sg',cruzar(E),E,Agt,Pat],
             _),
    Agt= type(_Ind_Agt,[nontemporal]),
    Pat= type(_Ind_Pat,[inanimate]),
    E= type(_Ind_E,[event])
   ].
```

```
s(X,spanish) --- gusta,
   [template(X,
            trans_1_s,
            [[lex, gusta],3,sg,'3sg',gustar(E),E,Agt,Pat],
            _),
    Agt= type(_Ind_Agt,[animate]),
    Pat= type(_Ind_Pat,[entity]),
    E= type(_Ind_E,[state])
   ].

%%%%%%%%%%%%%%%%%%%%%%%%%% Ditransitive Verbs %%%%%%%%%%%%%%%%%%%%%

s(X,spanish) --- dio,
   [template(X,
            ditrans_1_s,
            [[lex, dio],3,sg,'3sg',dar(E),E,Agt,Pat,Goal],
            _),
    Agt= type(_Ind_Agt,[animate]),
    Pat= type(_Ind_Pat,[inanimate]),
    Goal= type(_Ind_Goal,[animate]),
    E= type(_Ind_E,[event])
   ].



%%%%%%%%%%%%%%%%%%%%%%%%%% Volition Verbs %%%%%%%%%%%%%%%%%%%%%%%%%

s(X,spanish) --- quiere,
   [template(X,vol_1_s,[[lex, quiere],3,sg,'3sg',querer(E),E,Agt,I],_),
    Agt= type(_Ind_Agt,[animate]),
    E= type(_Ind_E,[event]),
    I= type(_Ind_I,[event])
   ].


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Adverbs %%%%%%%%%%%%%%%%%%%%%%%%%%%

s(X,spanish) --- nadando,
   [template(X,adv_1_s,[[lex,nadando],I,nadando(I)],_),
    I = type(_Ind_I,[temporal])
   ].

%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Determiners %%%%%%%%%%%%%%%%%%%%%%%%

s(X,spanish) --- el,
```

```
   [template(X,det_1_s,[[lex,el],3,sg,masc,X3,definite(X3)],_)].


%%%%%%%%%%%%%%%%%%%%%%%%%%%% Clitics %%%%%%%%%%%%%%%%%%%%%%%%%%%%

% case-absorbing

s(X,spanish) --- lo,
   [template(X,ca_clitic_1b_s,[[lex,lo],3,sg,masc,c(0,acc,0,0,0)],_)].

s(X,spanish) --- le,
    [template(X,ca_clitic_1b_s,[[lex,le_ca],3,sg,_,c(0,0,dat,0,0)],_)].

% non-case-absorbing

s(X,spanish) --- le,
   [template(X,nca_clitic_1b_s,[[lex,le_nca],3,sg,_,c(0,0,dat,0,0)],_)].


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% "a" used as a dative case marker (maps from NPs marked as
% nominative or accusative to NPs marked as dative).
% This version is used for non-human NPs (the ones that are ambiguous
% about nominative or accusative).

s(X,spanish) --- a,
        [X:derivation = [lex,a],
         X:cat = s,
         X:slash = [Y1,Y2],
                Y1:cat = s,
                Y1:slash = [Z1],
                        Z1:cat = s,
                        Z1:slash = [],
                        Z1:order = fwd,
                        Z1:feats = Feats,
                        Z1:cases = set([Case1]),
                               Case1:agr:person = Pers,
                               Case1:agr:number = Num,
                               Case1:agr:gender = Gen,
                               Case1:agr:case = c(nom,acc,0,0,0),
                               Case1:index = SemInd,
                        Z1:sem:index = I,
                        Z1:sem:conj = Sem,
                Y1:order = fwd,
```

```
                Y1:cases = set([]),
                Y1:feats = Feats,
                Y1:sem:index = I,
                Y1:sem:conj = [SemConj|Sem],
                Y1:sem:indices = Z1:sem:indices,

                Y2:cat = s,
                Y2:slash = [],
                Y2:feats = Feats,
                Y2:cases = set([Case2|OtherCases]),
                        Case2:agr:person = Pers,
                        Case2:agr:number = Num,
                        Case2:agr:gender = Gen,
                        Case2:agr:case = c(0,0,dat,0,0),
                        Case2:index = SemInd,
                Y2:sem:index = I,
                Y2:sem:conj = Sem,
        X:cases = set(OtherCases),
        X:feats = Feats,
        X:sem:index = I,
        X:sem:conj = [SemConj|Sem],
        X:sem:indices = Y2:sem:indices
      ].


% "a" used as an animacy object marker (maps from NPs marked as
% nominative with human index to NPs marked as accusative
% or dative).

s(X,spanish) --- a,
        [X:derivation = [lex,a],
         X:cat = s,
         X:slash = [Y1,Y2],
                Y1:cat = s,
                Y1:slash = [Z1],
                        Z1:cat = s,
                        Z1:slash = [],
                        Z1:order = fwd,
                        Z1:feats = Feats,
                        Z1:cases = set([Case1]),
                               Case1:agr:person = Pers,
                               Case1:agr:number = Num,
                               Case1:agr:gender = Gen,
                               Case1:agr:case = c(nom,0,0,0,0),
                               Case1:index = SemInd,
                        Z1:sem:index = I,
```

```
                Z1:sem:conj = Sem,
            Y1:order = fwd,
            Y1:cases = set([]),
            Y1:feats = Feats,
            Y1:sem:index = I,
            Y1:sem:conj = [SemConj|Sem],
            Y1:sem:indices = Z1:sem:indices,

            Y2:cat = s,
            Y2:slash = [],
            Y2:feats = Feats,
            Y2:cases = set([Case2|OtherCases]),
                Case2:agr:person = Pers,
                Case2:agr:number = Num,
                Case2:agr:gender = Gen,
                Case2:agr:case = c(0,_Acc,_Dat,0,0),
                Case2:index = SemInd,
            Y2:sem:index = I,
            Y2:sem:conj = Sem,
    X:cases = set(OtherCases),
    X:feats = Feats,
    X:sem:index = I,
    X:sem:conj = [SemConj|Sem],
    X:sem:indices = Y2:sem:indices
   ].
```

B.3 Lexical redundancy rules

```
% Title: lrul_sp.pl
% Description: lexical redundancy rules for Spanish, as described
% in Chapter 3 of my thesis.
%
% Author: John Beaven
% Last Update: 1 Sept 1991
%

% Format:
% lexrule(Language,In,Out,Goals).
% where Language is the language used, In is the input of the rule,
```

```
% Out is the Output, and Goals have to be satisfied.

% Object inversion rule for transitive verbs (looks for object on
% left rather than right: maps "VP"/"NP" into "VP"\"NP").

lexrule(spanish,In,Out,
        [In:derivation = Deriv1,
         In:cat = s,
         In:slash = [Y1],
                Y1:cat = s,
                Y1:slash = [Z1],
                       Z1:cat = s,
                       Z1:slash = [],
                       Z1:order = fwd,
                       Z1:feats = Feats1,
                       Z1:cases = set([Acc]),
                              Acc:agr:case = c(0,acc,0,0,0),
                Y1:order = fwd,
                Y1:vfinal = -,
                Y1:cases = set(More),
                Y1:sem = Sem2,
         In:feats = Feats1,
                Feats1:fin = +,
         In:cases = set([Nom | More]),
         In:sem = Sem3,

         Out:derivation = [lrule_vo_ov,Deriv1],
         Out:cat = s,
         Out:slash = [Y2],
                Y2:slash = [Z1],
                Y2:order = back,
                Y2:cases = set(More),
                Y2:sem = Sem2,
         Out:feats = Feats1,
         Out:cases = set([Nom | More]),
         Out:sem = Sem3
       ]).


% Subject inversion rule.
% Maps S/S into S\S.

lexrule(spanish,In,Out,
        [In:derivation = Deriv,
         In:cat = s,
         In:slash = [Y1],
```

```
              Y1:cat = s,
              Y1:slash = [],
              Y1:order = fwd,
              Y1:feats = F1,
              Y1:vfinal = -,
              Y1:cases = C1,
              Y1:sem = Sem1,
       In:cases = C2,
       In:feats = F2,
       In:sem = Sem2,
       Out:derivation = [lrule_subj_inv_n,Deriv],
       Out:cat = s,
       Out:slash = [Y2],
              Y2:cat = s,
              Y2:slash = [],
              Y2:order = back,
              Y2:feats = F1,
              Y2:cases = C1,
              Y2:sem = Sem1,
       Out:cases = C2,
       Out:feats = F2,
       Out:vfinal = -,
       Out:sem = Sem2
     ]).

% operates on determiners and maps (S/S)/noun into (S\S)/noun

lexrule(spanish,In,Out,
      [In:derivation = Deriv,
       In:cat = s,
       In:slash = [Y1,Y2],
              Y1:cat = noun,

              Y2:cat = s,
              Y2:slash = [],
              Y2:order = fwd,
              Y2:feats = Feats,
              Y2:vfinal = -,
              Y2:cases = Cases1,
              Y2:sem = Sem1,
       In:feats = Feats,
       In:cases = Cases2,
       In:sem = Sem2,

       Out:derivation = [lrule_subj_inv_d,Deriv],
       Out:cat = s,
```

```
  Out:slash = [Y1,Y3],
         Y3:cat = s,
         Y3:slash = [],
         Y3:order = back,
         Y3:feats = Feats,
         Y3:cases = Cases1,
         Y3:sem = Sem1,
  Out:feats = Feats,
  Out:cases = Cases2,
  Out:vfinal = -,
  Out:sem = Sem2
]).
```

Appendix C

English entries

This appendix contains the code for sample English entries. Just like the Spanish ones, they are built from `temp_eng.pl` (containing the basic templates), `lex_eng.pl` (the lexical entries themselves), and `lrul_eng.pl` (lexical redundancy rules).

C.1 Lexicon

```
% Title: temp_eng.pl
% Description: Templates for an English grammar illustrating
% the contents of Chapter 4 of my thesis.

% Author: John Beaven
% Last Update: 4 Sept 1991

% This file contains the basic templates used for building the lexical
% entries in lex_eng.pl

% Format:
% template(Sign, Template_Name, Parameter_List, Goals).
% where Sign is the whole feature-structure, Template_Name is a
% unique identifier for that particular template, Parameter_List
% are parameters passed from the lexical entry, and Goals have to
% be satisfied.

% Format of cases feature:
% c(Nom,Obj,Gen,Prep,List_of_Preps)
% where the first 4 arguments are boolean: either 0 (not allowed) or
% (instead of 1, for better legibility) nom, obj, gen, prep respectively.
% If more then one case is possible, anonymous variables are used
% in the places of the possible cases.
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%% Names %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

template(X,name_e,[Deriv,Pers,Num,Gen,SemInd,Pred],
        [X:derivation = Deriv,
         X:cat = np,
         X:slash = [],
         X:cases = set([Case]),
                Case:agr:person = Pers,
                Case:agr:number = Num,
                Case:agr:gender = Gen,
                Case:agr:case = c(_Nom,_Obj,0,0,0),
                Case:index = SemInd,
         X:sem:conj = [Pred|T]-T,
         X:sem:index = SemInd,
         X:sem:indices = [SemInd]
        ]).


%%%%%%%%%%%%%%%%%%%%%%%%%%%% Nouns %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

template(X,noun_e,[Deriv,Pers,Num,Gen,SemInd,Pred],
        [X:derivation = Deriv,
         X:cat = noun,
         X:slash = [],
         X:cases = set([Case]),
                Case:agr:person = Pers,
                Case:agr:number = Num,
                Case:agr:gender = Gen,
                Case:agr:case = c(_Nom,_Obj,0,0,0),
                Case:index = SemInd,
         X:sem:conj = [Pred|T]-T,
         X:sem:index = SemInd,
         X:sem:indices = [SemInd]
        ]).


%%%%%%%%%%%%%%%%%%%%%%%%%%%% Determiners %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

template(X,det_e,[Deriv,Num,SemInd,Pred],
        [X:derivation = Deriv,
         X:cat = np,
         X:slash = [Y],
                 Y:cat = noun,
                 Y:slash = [],
                 Y:order = fwd,
                 Y:cases = set([Case]),
```

```
                 Y:sem:conj = Sem1-Sem2,
                 Y:sem:index = SemInd,
         X:cases = set([Case]),
                 Case:agr:number = Num,
         X:sem:conj = [Pred|Sem1]-Sem2,
         X:sem:index = SemInd,
         X:sem:indices = [SemInd]
        ]).

%%%%%%%%%%%%%%%%%%%%%% Personal Pronouns %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

template(X,pers_pron_e,[Deriv,Pers,Num,Gen,Cse,SemInd],
        [X:derivation = Deriv,
         X:cat = np,
         X:slash = [],
         X:cases = set([Case]),
                Case:agr:person = Pers,
                Case:agr:number = Num,
                Case:agr:gender = Gen,
                Case:agr:case = Cse,
                Case:index = SemInd,
         X:sem:index = SemInd,
         X:sem:indices = [SemInd]
        ]).


%%%%%%%%%%%%%%%%%%%%%% Intransitive Verbs %%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Intransitive Verbs - Type 1 (active, finite) (subj is nominative).
template(X,intrans_1_e,[Deriv,Pers,Num,Vform,Pred,SemInd,Agt],
        [X:derivation = Deriv,
         X:cat = s,
         X:slash = [Y1],
                Y1:cat = np,
                Y1:slash = [],
                Y1:cases = set([Nom]),
                        Nom:agr:number = Num,
                        Nom:agr:person = Pers,
                        Nom:agr:case = c(nom,0,0,0,0),
                        Nom:index = Agt,
                Y1:order = back,
                Y1:sem:index =Agt,
                Y1:sem:conj = Sem1-Sem2,
         X:cases = set([]),
         X:feats = Feats1,
                Feats1:vform = Vform,
```

```
         X:sem:index = SemInd,
         X:sem:conj = [Pred,role(SemInd,agt,Agt)|Sem1]-Sem2,
         X:sem:index = SemInd,
         X:sem:indices = [SemInd,Agt]
        ]).

% Intransitive Verbs - Type 2 (active, finite) (subj is nominative).
% Take 1 fixed preposition (eg wash up)
template(X,intrans_2_e,[Deriv,Prep1,Pers,Num,Vform,Pred,SemInd,Agt],
        [X:derivation = Deriv,
         X:cat = s,
         X:slash = [P1,Y1],
                P1:derivation = [lex,Prep1],
                P1:order = fwd,

                Y1:cat = np,
                Y1:slash = [],
                Y1:cases = set([Nom]),
                        Nom:agr:number = Num,
                        Nom:agr:person = Pers,
                        Nom:agr:case = c(nom,0,0,0,0),
                        Nom:index = Agt,
                Y1:order = back,
                Y1:sem:index =Agt,
                Y1:sem:conj = Sem1-Sem2,
         X:cases = set([]),
         X:feats = Feats1,
                Feats1:vform = Vform,
         X:sem:index = SemInd,
         X:sem:conj = [Pred,role(SemInd,agt,Agt)|Sem1]-Sem2,
         X:sem:index = SemInd,
         X:sem:indices = [SemInd,Agt]
        ]).

% Intransitive Verbs - Type 3 (active, non-finite) (subj is objective).
template(X,intrans_3_e,[Deriv,Pers,Num,Vform,Pred,SemInd,Agt],
        [X:derivation = Deriv,
         X:cat = s,
         X:slash = [Y1],
                Y1:cat = np,
                Y1:slash = [],
                Y1:cases = set([Nom]),
                        Nom:agr:number = Num,
                        Nom:agr:person = Pers,
                        Nom:agr:case = c(0,obj,0,0,0),
                        Nom:index = Agt,
```

```
                Y1:order = back,
                Y1:sem:index =Agt,
                Y1:sem:conj = Sem1-Sem2,
         X:cases = set([]),
         X:feats = Feats1,
                Feats1:vform = Vform,
         X:sem:index = SemInd,
         X:sem:conj = [Pred,role(SemInd,agt,Agt)|Sem1]-Sem2,
         X:sem:index = SemInd,
         X:sem:indices = [SemInd,Agt]
        ]).


%%%%%%%%%%%%%%%%%%%%%% Transitive Verbs %%%%%%%%%%%%%%%%%%%%%%%%%%

% Transitive Verbs - Type 1 (Active)
template(X,trans_1_e,[Deriv,Pers,Num,Vform,Pred,SemInd,Agt,Pat],
        [X:derivation = Deriv,
         X:cat = s,
         X:slash = [Y1,Y2],
                Y1:cat = np,
                Y1:slash = [],
                Y1:cases = set([Obj]),
                        Obj:agr:case = c(0,obj,0,0,0),
                        Obj:index = Pat,
                Y1:order = fwd,
                Y1:sem:index =Pat,
                Y1:sem:conj = Sem1-Sem2,

                Y2:cat = np,
                Y2:slash = [],
                Y2:cases = set([Nom]),
                        Nom:agr:number = Num,
                        Nom:agr:person = Pers,
                        Nom:agr:case = c(nom,0,0,0,0),
                        Nom:index = Agt,
                Y2:order = back,
                Y2:sem:index =Agt,
                Y2:sem:conj = Sem2-Sem3,

         X:cases = set([]),
         X:feats = Feats1,
                Feats1:vform = Vform,
         X:sem:index = SemInd,
         X:sem:conj = [Pred,role(E,agt,Agt),role(E,pat,Pat)|Sem1]-Sem3,
         X:sem:index = SemInd,
```

```
         X:sem:indices = [SemInd,Agt,Pat]
        ]).

% Transitive Verbs - Type 2 (Active) (take 2 fixed prepositions)
template(X,
        trans_2_e,
        [Deriv,Prep1,Prep2,Pers,Num,Vform,Pred,SemInd,Agt,Pat],
        [X:derivation = Deriv,
         X:cat = s,
         X:slash = [P1,P2,Y1,Y2],
                P1:derivation = [lex, Prep1],
                P1:order = fwd,
                P2:derivation = [lex, Prep2],
                P2:order = fwd,

                Y1:cat = np,
                Y1:slash = [],
                Y1:cases = set([Obj]),
                        Obj:agr:case = c(0,obj,0,0,0),
                        Obj:index = Pat,
                Y1:order = fwd,
                Y1:sem:index =Pat,
                Y1:sem:conj = Sem1-Sem2,

                Y2:cat = np,
                Y2:slash = [],
                Y2:cases = set([Nom]),
                        Nom:agr:number = Num,
                        Nom:agr:person = Pers,
                        Nom:agr:case = c(nom,0,0,0,0),
                        Nom:index = Agt,
                Y2:order = back,
                Y2:sem:index =Agt,
                Y2:sem:conj = Sem2-Sem3,

         X:cases = set([]),
         X:feats = Feats1,
                Feats1:vform = Vform,
         X:sem:index = SemInd,
         X:sem:conj = [Pred,role(E,agt,Agt),role(E,pat,Pat)|Sem1]-Sem3,
         X:sem:index = SemInd,
         X:sem:indices = [SemInd,Agt,Pat]
        ]).


%%%%%%%%%%%%%%%%%%%%% Ditransitive Verbs %%%%%%%%%%%%%%%%%%%%%%%%%
```

```
% Ditransitive Verbs - Type 1 (Active, Object + to)
template(X,ditrans_1_e,[Deriv,Pers,Num,Vform,Pred,SemInd,Agt,Pat,Goal],
        [X:derivation = Deriv,
         X:cat = s,
         X:slash = [Y1,Y2,Y3],
                Y1:cat = np,
                Y1:slash = [],
                Y1:cases = set([Obj]),
                        Obj:agr:case = c(0,obj,0,0,0),
                        Obj:index = Pat,
                Y1:order = fwd,
                Y1:sem:index =Pat,
                Y1:sem:conj = Sem1-Sem2,

                Y2:cat = pp,
                Y2:slash = [],
                Y2:cases = set([Dat]),
                        Dat:agr:case = c(0,0,0,prep,[to]),
                        Dat:index = Goal,
                Y2:order = fwd,
                Y2:sem:index =Goal,
                Y2:sem:conj = Sem2-Sem3,

                Y3:cat = np,
                Y3:slash = [],
                Y3:cases = set([Nom]),
                        Nom:agr:number = Num,
                        Nom:agr:person = Pers,
                        Nom:agr:case = c(nom,0,0,0,0),
                        Nom:index = Agt,
                Y3:order = back,
                Y3:sem:index =Agt,
                Y3:sem:conj = Sem3-Sem4,

         X:cases = set([]),
         X:feats = Feats1,
                Feats1:vform = Vform,
         X:sem:index = SemInd,
         X:sem:conj = [Pred,role(E,agt,Agt),
                        role(E,pat,Pat),
                        role(E,goal,Goal)|Sem1]-Sem4,
         X:sem:indices = [SemInd,Agt,Pat,Goal]
        ]).
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%% Volition Verbs %%%%%%%%%%%%%%%%%%%%%%%

% Volition Verbs - Type 1 (active, subcat for a "to" sentence,
% subject control) eg John wants to sing
template(X,vol_1_e,[Deriv,Pers,Num,Vform,Pred,SemInd,Agt,Wanted],
        [X:derivation = Deriv,
         X:cat = s,
         X:slash = [Y1,Y2],
                Y1:cat = s,
                Y1:slash = [Y3],       % subj control
                       Y3:cat = np,
                       Y3:slash = [],
                       Y3:order = back,
                       Y3:cases = set([Acc]),
                             Acc:agr:number = Num,
                             Acc:agr:person = Pers,
                             Acc:agr:case = c(0,obj,0,0,0),
                             Acc:index = Agt,
                       Y3:order = back,
                       Y3:sem:index = Agt,
                       Y3:sem:conj = Sem1-Sem1,

                Y1:cases = set([]),
                Y1:order = fwd,
                Y1:feats:vform = inf,
                Y1:sem:index =Wanted,
                Y1:sem:conj = Sem2-Sem3,

                Y2:cat = np,
                Y2:slash = [],
                Y2:order = back,
                Y2:cases = set([Nom]),
                        Nom:agr:number = Num,
                        Nom:agr:person = Pers,
                        Nom:agr:case = c(nom,0,0,0,0),
                        Nom:index = Agt,
                Y2:order = back,
                Y2:sem:index =Agt,
                Y2:sem:conj = Sem3-Sem4,

         X:cases = set([]),
         X:feats = Feats,
                Feats:vform = Vform,
         X:sem:index = SemInd,
         X:sem:conj = [Pred,
                       role(E,agt,Agt),
```

```
                  role(E,pat,Wanted)|Sem2]-Sem4,
         X:sem:index = SemInd,
         X:sem:indices = [SemInd,Agt,Wanted]
        ]).




%%%%%%%%%%%%%%%%%%%%%%%%%%%% Prepositions %%%%%%%%%%%%%%%%%%%%%%%%%%%%


% Prepositions - Type 1: giving PPs that may be subcategorized for.
% eg: "to Mary" with a ditransitive verb.
% These to not have semantic effects.
template(X,prep_1_e,[Deriv],
        [X:derivation = Deriv,
         X:cat = pp,
         X:slash = [Y],
                Y:cat = np,
                Y:slash = [],
                Y:order = fwd,
                Y:cases = set([Case1]),
                       Case1:agr:person = Pers,
                       Case1:agr:number = Num,
                       Case1:agr:gender = Gen,
                       Case1:agr:case = c(0,obj,0,0,0),
                       Case1:index = SemInd,
                Y:sem:conj = SemConj,
                Y:sem:index = SemInd,
         X:cases = set([Case]),
                Case:agr:person = Pers,
                Case:agr:number = Num,
                Case:agr:gender = Gen,
                Case:agr:case = c(0,0,0,prep,[to]),
                Case:index = SemInd,
         X:sem:conj = SemConj,
         X:sem:index = SemInd
        ]).

% Prepositions yielding sentential modifiers: (s/s)/np.
% These have semantic contents - eg: across
template(X,prep_2_e,[Deriv,Role,Pred,SemInd],
        [X:derivation = Deriv,
         X:cat = s,
         X:slash = [Y1, Y2],
```

```
               Y1:cat = np,
               Y1:slash = [],
               Y1:order = fwd,
               Y1:cases = set([Obj]),
                       Obj:agr:case = c(0,obj,0,0,0),
                       Obj:index = Role,
               Y1:order = fwd,
               Y1:sem:index = Role,
               Y1:sem:conj = Sem1-Sem2,

               Y2:cat = s,
               Y2:slash = [],
               Y2:order = back,
               Y2:cases = set(MoreCases),
               Y2:sem:index  = SemInd,
               Y2:sem:conj = Sem2-Sem3,
               Y2:sem:indices = [E|Indices],

        X:cases = set(MoreCases),
        X:sem:index = SemInd,
        X:sem:conj = [Pred|Sem1]-Sem3,
        X:sem:indices = [E,Role|Indices]
       ]).
```

C.2 Entries

```
% Title: lex_eng.pl
% Description: Lexical entries for an English grammar
% illustrating the contents of Chapter 4 of my thesis.
```

```
%
% This is meant to be used in conjuction with the English templates
% in temp_eng.pl

% Author: John Beaven
% Last Update: 4 Sept 1991

% Format:
% s(Sign,Language) --- Orthography, Goals
% where Sign is the representation of the attribute-feature structure,
% and Goals have to be satisfied.

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%  English Monolingual lexical entries
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%% Names %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

s(X,english) --- 'Mary',
   [template(X,name_e,[[lex,'Mary'],3,sg,fem,F3,name(F3,mary)],_),
    F3 = type(_Ind_F3,[fem,sing])
   ].


s(X,english) --- 'John',
   [template(X,name_e,[[lex,'John'],3,sg,masc,M3,name(M3,john)],_),
    M3 = type(_Ind_M3,[male,sing])
   ].


%%%%%%%%%%%%%%%%%%%%%%%%%%%% Nouns %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


s(X,english) --- book,
[template(X,noun_e,[[lex,book],3,sg,neut,X3,book(X3)],_),
 X3 = type(_Ind_X3,[inanimate,sing])
 ].

s(X,english) --- man,
[template(X,noun_e,[[lex,man],3,sg,neut,X3,man(X3)],_),
 X3 = type(_Ind_X3,[male,sing])
 ].

s(X,english) --- river,
[template(X,noun_e,[[lex,river],3,sg,neut,X3,river(X3)],_),
 X3 = type(_Ind_X3,[inanimate,sing])
```

```
 ].


%%%%%%%%%%%%%%%%%%%%%% Intransitive Verbs %%%%%%%%%%%%%%%%%%%%%%%%%%

% finite
s(X,english) --- sings,
   [template(X,intrans_1_e,[[lex, sings],3,sg,'3sg',sing(E),E,Agt],_),
    Agt= type(_Ind_Agt,[animate]),
    E= type(_Ind_E,[event])
   ].


s(X,english) --- swam,
   [template(X,intrans_1_e,[[lex, swam],3,sg,'3sg',swim(E),E,Agt],_),
    Agt= type(_Ind_Agt,[animate]),
    E= type(_Ind_E,[event])
   ].

% washes up
s(X,english) --- washes,
   [template(X,
             intrans_2_e,
             [[lex, washes],up,3,sg,'3sg',wash_up(E),E,Agt],
             _),
    Agt= type(_Ind_Agt,[animate]),
    E= type(_Ind_E,[event])
   ].

% base form
s(X,english) --- sing,
   [template(X,intrans_3_e,[[lex, sing],3,sg,base,sing(E),E,Agt],_),
    Agt= type(_Ind_Agt,[animate]),
    E= type(_Ind_E,[event])
   ].


%%%%%%%%%%%%%%%%%%%%%% Transitive Verbs %%%%%%%%%%%%%%%%%%%%%%%%%%

s(X,english) --- read,
   [template(X,trans_1_e,[[lex, read],3,sg,'3sg',read(E),E,Agt,Pat],_),
    Agt= type(_Ind_Agt,[human]),
    Pat= type(_Ind_Pat,[inanimate]),
    E= type(_Ind_E,[event])
   ].
```

```
s(X,english) --- saw,
   [template(X,trans_1_e,[[lex, saw],3,sg,'3sg',see(E),E,Agt,Pat],_),
    Agt= type(_Ind_Agt,[human]),
    Pat= type(_Ind_Pat,[entity]),
    E= type(_Ind_E,[event])
   ].

s(X,english) --- likes,
   [template(X,trans_1_e,[[lex, likes],3,sg,'3sg',like(E),E,Agt,Pat],_),
    Agt= type(_Ind_Agt,[human]),
    Pat= type(_Ind_Pat,[entity]),
    E= type(_Ind_E,[state])
   ].

% puts up with
s(X,english) --- puts,
   [template(X,trans_2_e,
    [[lex, puts],up,with,3,sg,'3sg',put_up_with(E),E,Agt,Pat],_),
    Agt= type(_Ind_Agt,[human]),
    Pat= type(_Ind_Pat,[entity]),
    E= type(_Ind_E,[event])
   ].

%%%%%%%%%%%%%%%%%%%%% Ditransitive Verbs %%%%%%%%%%%%%%%%%%%%%%%%%%

% Version 1: gave the book to mary

s(X,english) --- gave,
[template(X,
          ditrans_1_e,
          [[lex,gave],3,sg,'3sg',give(E),E,Agt,Pat,Goal],
          _),
    Agt= type(_Ind_Agt,[human]),
    Pat= type(_Ind_Pat,[inanimate]),
    Goal= type(_Ind_Goal,[human]),
    E= type(_Ind_E,[event])
   ].



%%%%%%%%%%%%%%%%%%%%%%%%%%%% Volition Verbs %%%%%%%%%%%%%%%%%%%%%%%

s(X,english) --- wants,                % subject control
   [template(X,vol_1_e,[[lex, wants],3,sg,'3sg',want(E),E,Agt,Pat],_),
    Agt= type(_Ind_Agt,[animate]),
    E= type(_Ind_E,[event]),
```

```
    Pat= type(_Ind_Pat,[event])
   ].

%%%%%%%%%%%%%%%%%%%%%%%%%%% Determiners %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

s(X,english) --- the,
[template(X,det_e,[[lex,the],_,X3,definite(X3)],_)].

%%%%%%%%%%%%%%%%%%%%%% Personal Pronouns %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

s(X,english) --- she,
   [template(X,pers_pron_e,[[lex,she],3,sg,fem,c(nom,0,0,0,0),F3],_),
    F3 = type(_Ind_F3,[fem,sing])
   ].

s(X,english) --- he,
   [template(X,pers_pron_e,[[lex,he],3,sg,masc,c(nom,0,0,0,0),M3],_),
    M3 = type(_Ind_M3,[male,sing])
   ].

s(X,english) --- her,
   [template(X,pers_pron_e,[[lex,her],3,sg,fem,c(0,obj,0,0,0),F3],_),
    F3 = type(_Ind_F3,[fem,sing])
   ].

s(X,english) --- him,
   [template(X,pers_pron_e,[[lex,him],3,sg,masc,c(0,obj,0,0,0),M3],_),
    M3 = type(_Ind_M3,[male,sing])
   ].

s(X,english) --- it,
   [template(X,pers_pron_e,[[lex,it],3,sg,neut,c(_Nom,_Obj,0,0,0),X3],_),
    X3 = type(_Ind_X3,[sing])
   ].


%%%%%%%%%%%%%%%%%%%%%%%%%%% Prepositions %%%%%%%%%%%%%%%%%%%%%%%%%%

s(X,english) --- across,
  [template(X,
            prep_2_e,
            [[lex,across],Crossed,role(E,across,Crossed),E],
            _),
   Crossed = type(_Ind_Crossed,[inanimate]),
   E = type(_Ind_E,[temporal])
  ].
```

```
s(X,english) --- up,
  [template(X,prep_2_e,[[lex,up],Up,role(E,up,Up),E],_),
   Up = type(_Ind_Up,[nontemporal]),
   E = type(_Ind_E,[temporal])
  ].

s(X,english) --- with,
  [template(X,prep_2_e,[[lex,with],With,role(E,with,With),E],_),
   With = type(_Ind_With,[nontemporal]),
   E = type(_Ind_E,[temporal])
  ].

% "to" used as a "dative" marker
s(X,english) --- to,
   [template(X,prep_1_e,[[lex,to_dat]],_)].

% "to" turning verb phrase from base form into an infinitive
s(X,english) --- to,
        [X:derivation = [lex, to_inf],
         X:cat = s,
         X:slash = [Y,NP],
                Y:cat = s,
                Y:slash = [NP],
                      NP:cat = np,
                Y:order = fwd,
                Y:feats:vform = base,
                Y:cases = Cases,
                Y:sem = Sem,
         X:feats:vform = inf,
         X:cases = Cases,
         X:sem = Sem
        ].
```

C.3 Lexical redundancy rules

```
% Title: lrul_eng.pl
% Description: lexical redundancy rules for English, as described
% in Chapter 4 of my thesis.
%
% Author: John Beaven
% Last Updated: 1 Sept 1991
%

% Format:
% lexrule(Language,In,Out,Goals).
% where Language is the language used, In is the input of the rule,
% Out is the Output, and Goals have to be satisfied.

% Dative movement rule
% Takes a ditransitive subcategorizing for Obj1 and to-marked PP
% and returns a ditransitive subcategorizing for Obj2 and Obj1.

lexrule(english,In,Out,
       [In:derivation = Deriv1,
        In:cat = s,
        In:slash = [Y1,Y2,Y3],
               Y1:cat = np,
               Y1:slash = [],
               Y1:cases = set([Obj]),
                       Obj:agr:case = c(0,obj,0,0,0),
                       Obj:index = Pat,
               Y1:order = fwd,
               Y1:sem:index =Pat,

               Y2:cat = pp,
               Y2:slash = [],
               Y2:cases = set([Dat]),
                       Dat:agr:case = c(0,0,0,prep,[to]),
                       Dat:index = Goal,
               Y2:order = fwd,
               Y2:sem:index =Goal,
               Y2:sem:conj = Sem2,

               Y3:cat = np,
               Y3:slash = [],
               Y3:cases = set([Nom]),
```

```
                Nom:agr:case = c(nom,0,0,0,0),
                Nom:index = Agt,
        Y3:order = back,
        Y3:sem:index =Agt,

  In:cases = set([]),
  In:feats = Feats1,
  In:sem = Sem,

  Out:derivation = [lrule_dat_shift,Deriv1],
  Out:cat = s,
  Out:slash = [Y4,Y1,Y3],
        Y4:cat = np,
        Y4:slash = [],
        Y4:cases = set([Obj2]),
               Obj2:agr:case = c(0,obj,0,0,0),
               Obj2:index = Goal,
        Y4:order = fwd,
        Y4:sem:index = Goal,
        Y4:sem:conj = Sem2,

  Out:cases = set([]),
  Out:feats = Feats1,
  Out:sem = Sem
 ]).
```

Appendix D

Bilingual entries

This appendix shows the file `lex_bi.pl`, which contains the bilingual lexicon, in the form of correspondences between monolingual entries and constraints for the correspondences to hold.

```
% Title: lex_bi.pl
% Description: Bilingual Lexicon that puts together the two
% monolingual ones
% Author: John Beaven
% Date Created: 25 Feb 1991
% Last Update: 4 Sept 1991
%
% This file contains some sample bilingual lexical entries, as described
% in Chapters 5 and 7 of my thesis.
% The general format is
%
% Spanish_Entries <---> English_Entries, Goals.
%
% where Spanish_Entries and English_Entries are lists (representing bags)
% of monolingual entries, and Goals is the list of goals that have to
% be satisfied for the bilingual correspondence to hold.

[entry(spanish,'Maria',XS)] <---> [entry(english,'Mary',XE)],
        [XS:sem:indices = XE:sem:indices].

[entry(spanish,'Juan',XS)] <---> [entry(english,'John',XE)],
        [XS:sem:indices = XE:sem:indices].

[entry(spanish,'libro',XS)] <---> [entry(english,'book',XE)],
        [XS:sem:indices = XE:sem:indices].

[entry(spanish,'rio',XS)] <---> [entry(english,'river',XE)],
```

```
        [XS:sem:indices = XE:sem:indices].

[entry(spanish,'hombre',XS)] <---> [entry(english,'man',XE)],
        [XS:sem:indices = XE:sem:indices].

[entry(spanish,'canta',XS)] <---> [entry(english,'sings',XE)],
        [XS:sem:indices = XE:sem:indices].

[entry(spanish,'lava',XS)] <---> [entry(english,'washes',XE),
                                  entry(english,'up',_)],
        [XS:sem:indices = XE:sem:indices].

[entry(spanish,'cantar',XS)] <---> [entry(english,'sing',XE)],
        [XS:sem:indices = XE:sem:indices].

[entry(spanish,'leyo',XS)] <---> [entry(english,'read',XE)],
        [XS:sem:indices = XE:sem:indices].

[entry(spanish,'vio',XS)] <---> [entry(english,'saw',XE)],
        [XS:sem:indices = XE:sem:indices].

[entry(spanish,'aguanta',XS)] <---> [entry(english,'puts',XE1),
                                     entry(english,'up',_XE2),
                                     entry(english,'with',_XE3)],
        [XS:sem:indices = XE1:sem:indices].

[entry(spanish,'cruzo',XS)] <---> [entry(english,'across',XE)],
        [XS:sem:indices = [Event,Crosser,Crossed|Rest],
         XE:sem:indices = [Event,Crossed,Crosser|Rest]].

% note the argument switch (compare with "leyo").
[entry(spanish,'gusta',XS)] <---> [entry(english,'likes',XE)],
        [XS:sem:indices = [Event,Liked,Liker|Rest],
         XE:sem:indices = [Event,Liker,Liked|Rest]].

[entry(spanish,'dio',XS)] <---> [entry(english,'gave',XE)],
        [XS:sem:indices = XE:sem:indices].

[entry(spanish,'quiere',XS)] <---> [entry(english,'wants',XE)],
        [XS:sem:indices = XE:sem:indices].

[entry(spanish,'nadando',XS)] <---> [entry(english,'swam',XE)],
        [XS:sem:indices = [Event,Swimmer|_],
         XE:sem:indices = [Event,Swimmer|_]].

[entry(spanish,'el',XS)] <---> [entry(english,'the',XE)],
```

```
        [XS:sem:indices = XE:sem:indices].

[entry(spanish,'lo',XS)] <---> [entry(english,'it',XE)],
        [XS:sem:indices = XE:sem:indices].

[entry(spanish,'lo',XS)] <---> [entry(english,'him',XE)],
        [XS:sem:indices = XE:sem:indices].

[entry(spanish,'le',XS)] <---> [entry(english,'it',XE)],
        [XS:sem:indices = XE:sem:indices].

[entry(spanish,'le',XS)] <---> [entry(english,'him',XE)],
        [XS:sem:indices = XE:sem:indices].

[entry(spanish,'le',XS)] <---> [entry(english,'her',XE)],
        [XS:sem:indices = XE:sem:indices].

[entry(spanish,'le',_XS)] <---> empty(english),
        [].

[entry(spanish,'a',_XS)] <---> empty(english),
        [].

empty(spanish) <---> [entry(english,'she',_XE)],
        [].

empty(spanish) <---> [entry(english,'he',_XE)],
        [].

empty(spanish) <---> [entry(english,'it',_XE)],
        [].

empty(spanish) <---> [entry(english,'to',_XE)],
        [].
```

Appendix E

Grammar rules

This final appendix contains the grammar rules. There are four of them: forward and backward function application, and forward and backward function composition.

```
% Title: grammar.pl
% Description: A simple categorial grammar in PRATTLE notation for
% Spanish & English with a 'neo-davidsonian' treatment of semantics,
% as presented in Chapters 3 & 4 of my thesis.
%
% Date Created: August 1989
% Last update: 28 March 1992

% Simple forward application

s(X,Lang) ---> s(Y,Lang),s(Z,Lang),
        [X:derivation = [fa, Y:derivation, Z:derivation],
         X:cat = Y:cat,
         Y:slash = [Z|Rest],
         Z:order = fwd,
         X:slash =  Rest,
         X:order = Y:order,
         X:feats = Y:feats,
         X:vfinal = Z:vfinal,     % inherited from right daughter
         X:lex = -,
         X:cases = Y:cases,
         X:sem = Y:sem
        ].



% simple backward application
```

```
s(X,Lang) ---> s(Z,Lang),s(Y,Lang),
        [X:derivation = [ba, Z:derivation, Y:derivation],
         X:cat = Y:cat,
         Y:slash = [Z|Rest],
         Z:order = back,
         X:slash =  Rest,
         X:order = Y:order,
         X:feats = Y:feats,
         X:vfinal = Y:vfinal,    % inherited from right daughter
         X:lex = -,
         X:cases = Y:cases,
         X:sem = Y:sem
         ].

% forward composition (Chapter 3 of thesis)

s(X,Lang) ---> s(Y,Lang),s(Z,Lang),
        [X:derivation = [fc, Y:derivation, Z:derivation],
         X:cat = Y:cat,
         Y:slash = [Z1],
                 Z1:cat = Z:cat,
                 Z1:slash = B,
                 Z1:order = fwd,
                 Z1:feats = Z:feats,
                 Z1:cases = Z:cases,
                 Z1:sem = Z:sem,
         Y:cases = X:cases,
         Z:slash = ZSlash,
         Z:order = fwd,
         prolog(append(C,B,ZSlash)),
         X:slash =  C,
         C = [_Head | _Tail],    % Make sure C is non-empty,
                                 % otherwise this duplicates FA
         X:order = Y:order,
         X:feats = Y:feats,
         X:lex = -,
         X:vfinal = Z:vfinal,    % inherited from right daughter
         X:sem = Y:sem
        ].

% backward composition (Chapter 3 of thesis)

s(X,Lang) ---> s(Z,Lang),s(Y,Lang),
        [X:derivation = [fc, Y:derivation, Z:derivation],
         X:cat = Y:cat,
         Y:slash = [Z1],
```

```
        Z1:cat = Z:cat,
        Z1:slash = B,
        Z1:order = back,
        Z1:feats = Z:feats,
        Z1:cases = Z:cases,
        Z1:sem = Z:sem,
 Y:cases = X:cases,
 Z:slash = ZSlash,
 Z:order = back,
 prolog(append(C,B,ZSlash)),
 X:slash =  C,
 C = [_Head | _Tail],    % Make sure C is non-empty,
                         % otherwise this duplicates BA
 X:order = Y:order,
 X:feats = Y:feats,
 X:lex = -,
 X:vfinal = Y:vfinal,    % inherited from right daughter
 X:sem = Y:sem
].
```