WRITING LARGE-SCALE PARALLEL GRAMMARS FOR ENGLISH, FRENCH, AND GERMAN

Miriam Butt, Stefanie Dipper, Anette Frank, Tracy Holloway King
University of Konstanz, IMS Stuttgart, XRCE, Xerox PARC

Proceedings of the LFG99 Conference

The University of Manchester

Miriam Butt and Tracy Holloway King (Editors)

1999

CSLI Publications
`http://www-csli.stanford.edu/publications/`

1 Introduction

This paper discusses issues relevant to writing large-scale parallel grammars.[1] It is a direct result of our experiences with ParGram, a parallel grammar project involving Xerox PARC (English), XRCE (French), IMS Stuttgart (German), and University of Bergen (Norwegian). The basic goal of the ParGram project is to write large-scale LFG grammars with parallel analyses. In this introduction, we define what we mean by parallel analyses and by large scale, and briefly discuss the system which we use. There are three basic aspects to parallel grammars:

Similar analyses for similar phenomena
Same basic coverage
Common features, values, node names, etc.

Section 1.1 discusses the first of these, namely what it means to have parallel analyses. The second issue is covered in section 1.2. The third point, that the grammars have common features, values, and node names, is not

[1]Each section of this paper was presented and then written up by a different author, although the overall content of the paper was created jointly. M. Butt wrote section 3.2 on underspecification; S. Dipper wrote section 2 on how a grammar is written; A. Frank wrote section 4 on machine translation; and T. H. King wrote the introduction and section 3.1 on morphosyntactic structure. The entire paper benefitted greatly from input from Jonas Kuhn.

discussed here other than to note that such conventions make parallelism more transparent to the user.

1.1 Parallel Analysis

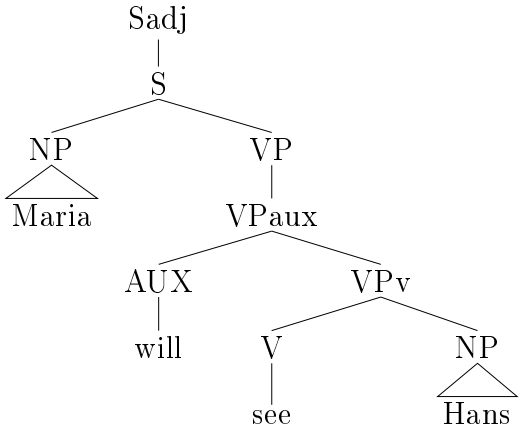
The basic idea behind parallel analysis is that, when linguistically justifiable, similar analyses are given to similar phenomena across languages. As such, a linguistically unjustified analysis is never forced on a language. However, if more than one analysis is possible, then the one that can be used in all the languages is chosen. Here we consider the representation of tense in English, French, and German. Consider the sentences in (1), which are translations of one another.

(1) Maria *will see* Hans. (English)
Maria *verra* Hans. (French)
Maria *wird* Hans *sehen*. (German)

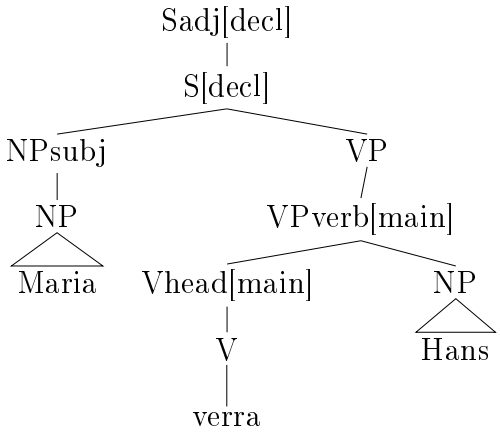
Although the basic meaning of the three sentences in (1) is identical, their morphosyntactic manifestation is different in all three languages. French uses just one word *verra* to represent the future tense, while English and German use two, namely an auxiliary and a main verb. English and German differ in that the auxiliary *will* is adjacent to the main verb *see* in English, whereas in German the auxiliary *wird* is in second position while the main verb *sehen* is in final position.

Given these differences in morphosyntactic representation, the constituent-structures for the sentences in each language differ on linguistically well-motivated grounds:

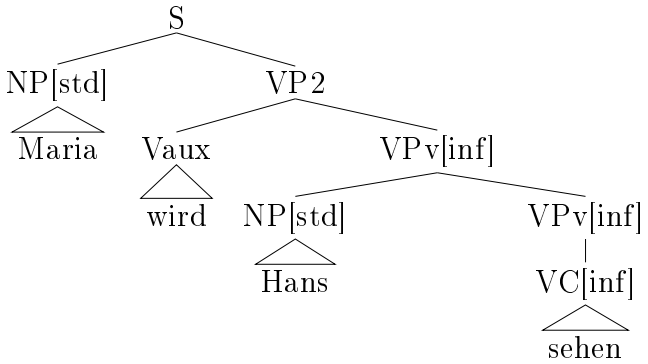
- English



- French



- German



Given these differences in the c-structure shown above, one might ask where the parallelism of the analysis comes in. The answer is in the f-structure. Since these sentences have similar meanings and especially similar syntactic behavior, we assign them similar f-structures, differing only in the PRED values. The main thing to notice about this f-structure is that the main verb is the top level predicate for all three languages. That is, the auxiliary in English and German does not provide a PRED feature. In addition, the TENSE feature of the f-structure is FUT for all three languages.

$$\begin{bmatrix}
\text{PRED} & \text{'see/voir/sehen<(}\uparrow \text{ SUBJ),(}\uparrow \text{ OBJ>'} \\
\text{TENSE} & \text{FUT} \\
\text{SUBJ} & \begin{bmatrix}
\text{PRED} & \text{'Maria'} \\
\text{NTYPE} & \begin{bmatrix}\text{PROPER} & \text{NAME}\end{bmatrix} \\
\text{PERS} & 3 \\
\text{GEND} & \text{FEM} \\
\text{NUM} & \text{SG} \\
\text{CASE} & \text{NOM}
\end{bmatrix} \\
\text{OBJ} & \begin{bmatrix}
\text{PRED} & \text{'Hans'} \\
\text{NTYPE} & \begin{bmatrix}\text{PROPER} & \text{NAME}\end{bmatrix} \\
\text{PERS} & 3 \\
\text{GEND} & \text{MASC} \\
\text{NUM} & \text{SG} \\
\text{CASE} & \text{ACC}
\end{bmatrix} \\
\text{PASSIVE} & - \\
\text{STMT-TYPE} & \text{DECLARATIVE} \\
\text{VTYPE} & \text{MAIN}
\end{bmatrix}$$

Thus, a parallel analysis for the German, French, and English tense system provides for remarkably similar f-structures for the three languages, while allowing for linguistically motivated c-structure variability. It is important to remember that this parallelism is only exploited when linguistically justifiable, as in the representation of morphosyntactic tense shown above.

1.2 Phenomena Considered

In order to be large scale, a grammar must cover a significant portion of the constructions in a language. In addition, the parallel grammars cover roughly the same constructions in each language (modulo the fact that some constructions only exist in some of the languages, e.g., French does not have particle verbs). A sample of the phenomena covered by the ParGram grammars includes:

declaratives, interrogatives, imperatives
embedded clauses, clausal adjuncts

subcategorization, auxiliaries, modals, particle verbs, predicatives
noun phrases, pronouns, compounds, relative clauses
determiners, adjectives
adverbs, negation, prepositional phrases
coordination

Each of these in turn involves a number of constructions which must be incorporated into the grammar. Consider the case of clausal adjuncts. An analysis of this construction must take into account the fact that they can occur (1) with or without a subordinating conjunction and can be (2) finite, infinitive, or participial (passive or progressive). Some instances of this are seen below for English:

When the light is red, push the button.

To start the engine, turn the key.

After closing the door, lock it carefully.

Having turned off the lights, stop the engine.

Implementing large-scale parallel grammars gives rise to a number of interesting theoretical questions due to a number of factors. First, implemented grammars require the grammar writer to be very explicit and hence it is impossible to gloss over "irrelevant" details of the analysis. Second, covering a large number of phenomena gives rise to interactions which otherwise remain unnoticed. Third, the parallel aspect of the grammars forces the grammar writer to consider why a particular analysis is chosen over another one and more generally to focus on the linguistic justifiability of any given analysis. Interesting issues of theoretical linguistic import which the project has encountered include: copular constructions, adjectival subjects, m(orphosyntactic)-structure, and the interaction of Optimality Theory and LFG.

1.3 Modularity in the System

Writing and maintaining large-scale grammars is made possible by modularity in the grammar implementation. Without this modularity, it would be extremely difficult to have a grammar which covered a significant portion of the linguistic constructions in a given language. In this section, we briefly

present the system used in the ParGram project as the backbone to parallel grammar writing, the Xerox Linguistic Environment (XLE).

The grammars comprise four basic components: a morphological analyzer, lexical entries, rules, and templates. The morphological analyzers take surface forms of words and analyze them as stem forms plus a series of tags which provide information about part of speech and other linguistically relevant factors. An example of this is seen below for the word 'sees', which is analyzed as the stem 'see' and three tags, one indicating it is a verb, one indicating it is present tense, and one indicating that it is third singular. (Note that some words may be assigned more than one morphological analysis, e.g., 'hit' is both a noun and a verb.) The morphological analyzers are developed completely independently of the grammar writing activity. As such, in the ParGram project we build on morphological analyzers that have already been developed for other uses.

(2) Morphological analyzer: sees $\longrightarrow$ see +Verb +Pres +3sg

To write a large-scale grammar, it is necessary to have a large lexicon. For words which have no subcategorization frames, such as most nouns, adjectives, and adverbs, and for ones which have predictable subcategorization frames, such as comparative adjectives taking 'than' clauses, it is possible to use the morphological analyzer to increase the available lexical items without writing explicit lexical entries for each item. However, for words like verbs which have variable subcategorization frames, it is necessary to have explicit lexical entries, as in (3). Fortunately, the modularity of the system allows us to incorporate large verb lexicons that have been compiled from other sources. Such sources include electronic dictionaries and corpora-derived entries. As such, it is possible to incorporate thousands of verbs into a lexicon without having to hand-code them, and such lexicons can be compiled by someone who is not working on the grammar itself. Similar methods can be used to compose lexicons of other types of items, such as nouns subcategorizing for 'that' clauses.

(3) Lexical entries:

```
see V XLE   { @(V-SUBJ-OBJ see)
            |@(V-SUBJ-COMP see)
            |...} .
```

The core of the grammar writing activities in ParGram focus on the grammar rules. These take the form of standard LFG rules with a few minor changes to allow for the ASCII format required by the XLE parser. A sample rule is seen in (4) for measure phrases like 'a three meter cord'. (4) states that a MEASUREP can be composed of either a number phrase preceding the head noun, with an optional hyphen, or a coordinated MEASUREP. (The default annotation of ↑=↓ is supplied by the parser.)

(4) Rules:

MEASUREP ⟶	{ NUMBERP:	(↑ SPEC)=↓
		(↓ NUMBER-TYPE)=c card;
	(HYPHEN)	
	N:	(↑ NUM)=c sg;
	| @(SCCOORD MEASUREP MEASUREP)}.	

In addition to rules, large-scale grammars make use of templates to allow for greater generalization. In particular, templates allow a complex set of information, such a rule annotations, to be given a name which can be invoked whenever that complex set of information is needed. As such, whenever a change is required, it only needs to be made to the template. One typical use of templates is for verb subcategorization frames, as in (5). (5) states that the template V-SUBJ-OBJ, the standard transitive verb template, takes one argument P. This argument becomes the PRED value of the verb and is given a subject and object argument. In addition, the template calls another template PASS which allows for passivization of the form in addition to the active variant (this can be thought of as capturing the fact that the active and passive forms are related). The PASS template can be called by any number of subcategorization frames.

(5) Templates:
V-SUBJ-OBJ(P) = @(PASS (↑ PRED)='P<(↑ SUBJ)(↑ OBJ)>').

A sample input and output of the system is shown below. The initial input is a string of words to be parsed:

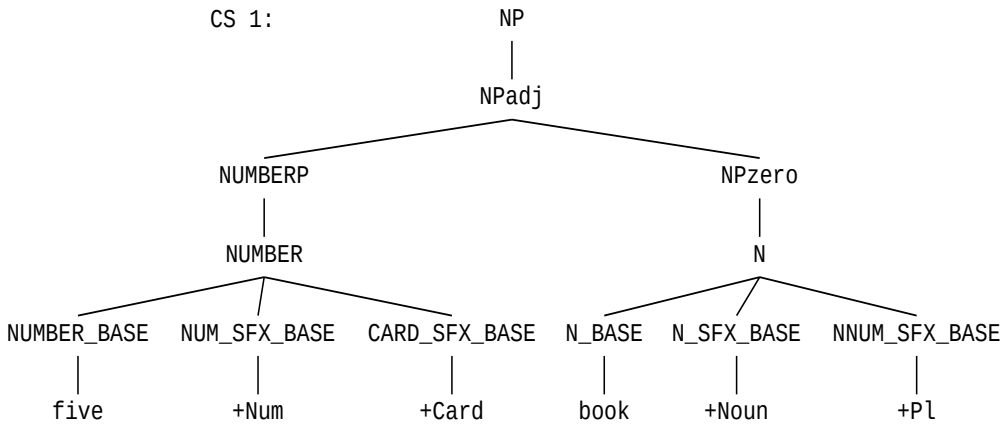
(6) parse "NP: five books"

This string is first given to the morphological analyzer (after having been broken into the appropriate tokens by a tokenizer) and gives a new string:

(7) five +Num +Card book +Noun +Pl

This new string, including the tags, is parsed by the grammar. The tags are treated like any other lexical item in that they are assigned a part of speech which the grammar recognizes. This sublexical information is normally hidden so that the linguist only sees the standard NUMBER and N leaves of the tree. For completeness the sublexical information is shown in (8).

(8)



The annotations on these rules in conjunction with the lexical entries result in the f-structure below:

(9)

```
"five book"s

    [PRED     'book'                                                    ]
    [NTYPE    [GRAIN count]                                             ]
 16 [                                                                    ]
210 [ADJUNCT {  0[PRED 'five'                                           ]}
226 [        {321[NUMBER-TYPE  card, NUM pl, ADJUNCT-TYPE  quant        ]}
287 [        {324[                                                      ]}
194 [ANIM -, PERS 3, NUM pl                                             ]
```

Thus, the modularity of the system allows for large-scale grammar writing since many of the components can be build independently of one another: the morphological analyzer, the lexicons, and the rules and templates.

1.4 Summary

This introduction discussed the ideas behind the ParGram project, namely what it means to write large-scale parallel grammars, and briefly described the system used. The remainder of this paper is structured as follows. Section 2 discusses how a large-scale grammar is written, focussing on basic steps in grammar writing and how to balance broad coverage with linguistically motivated analyses. Section 3 discusses two theoretical implications that have arisen from the ParGram project, namely the positing of morpho-syntactic structure and issues with the definition of underspecification. Finally, section 4 presents a multilingual NLP application of our parallel grammar development effort, a recently evolving translation project which was briefly introduced and accompanied by a demonstration of the translation prototype as part of the workshop.

2 How a Grammar is Written – a Case Study in German Compound Nouns

In this section we illustrate what 'real life' grammar writing might look like.[2] Generally, whenever the grammar writer is confronted with a type of construction not yet covered by the grammar, she/he has to take into account the following aspects:

- which data are to be covered, i.e.
  - what types of data are instances of the construction in question?
  - how frequently does each type occur in corpora?
- which theoretical analyses are proposed in the literature?
- what are the alternative ways of modelling these analyses?

[2]S. Dipper would like to thank the other authors of this paper, Judith Berman, Steve Berman, Jonas Kuhn, and Sabine Schulte im Walde for helpful comments on this section. The work reported in this section has been partially funded by the *Deutsche Forschungsgemeinschaft* within the *Sonderforschungsbereich* 340, project B12 (Methods for extending, maintaining and optimising a comprehensive grammar of German).
Unless indicated otherwise, all examples in this section are taken from the *Huge German Corpus*, cf. fn. 6.

- which factors determine the choice between the alternatives?

Decisions within the last area mainly depend on the project's objectives such as (i) broad coverage, (ii) linguistically motivated analyses, (iii) efficient parsing.

Obviously, these objectives often are in conflict with each other: in order to enlarge the coverage of a grammar the grammar writer might add special rules for a frequently occurring construction. On the other hand, aiming at linguistically motivated analyses means seeking a general solution that covers all instances of a certain phenomenon. In the latter case, there is no difference between instances that occur frequently and others that are rare. In both cases interaction between the rules of the grammar may become more complex, which will have bad impact on efficiency.

In other words, the grammar writer has to find a compromise between these objectives. In the following subsections, we will have a closer look at German compound nouns and coordination in order to see what such a compromise may look like. In section 2.1 we present the data, followed by a theoretical analysis in section 2.2. The implementation is the topic of section 2.3.

2.1 The Data

It is well known that German compound nouns can be complex in structure. Section 2.1.1 presents basic data illustrating the structure of compound nouns; section 2.1.2 considers more complex data involving coordination.

2.1.1 Basic Data

Some examples of basic compound nouns are given in (10). Without going into much detail, let us note the structural properties that are relevant for our discussion: the head of a compound is the rightmost element and, among other things, determines gender, number, and case of the compound (e.g. *Union* in (10a)). The other constituents of the compound function as modifiers of the head and may consist of simple words (*Währung* in (10a); *Bund* and *innen* in (10b)), frequently followed by a so-called linking morpheme ("Fugenmorphem") like *-s-* or *-es-*. But often, the modifying constituents are compounds themselves (*Landtag* in (10c)). The internal structure of (10b,c) is indicated by brackets in (11a,b), respectively.

(10) a. Währungsunion
economy union
'economic union'

b. Bundesinnenminister
federation interior minister
'Federal Minister of the Interior'

c. Landtagsabgeordneter
state council representative
'member of the Landtag'

(11) a. [federation [interior minister]]

b. [[state council] representative]

When parsing a German noun, information about the noun's gender and declension class is looked up in an on-line dictionary. Since many compounds are not lexicalised, they are not listed as such in the dictionary. The compounds therefore have to be decomposed to obtain the relevant information about their head.

Decomposing a compound also has advantages for transfer based on f-structure. Other languages have different means expressing modification, e.g. by use of PPs or APs. That means that a compound cannot be translated literally but translation starts out from the compound's constituents.

In fact, decomposition should not only enumerate all basic constituents but also represent the internal structure, as shown in (11). However, we are faced with the problem that for many compounds, detailed semantic or contextual information is necessary for disambiguation, cf. (12). In addition to these ambiguous cases, there are unambiguous compounds whose internal bracketing is nevertheless difficult to determine, cf. (13a) and its potential structures in (13b).

(12) Kindergartenfest
child garden party
[child [garden party]] – 'garden party for children'
[[child garden] party] – 'party in the kindergarten'

(13) a. Landschaftsschutzgebiet
landscape conservation area
'nature reserve'

b. [landscape [conservation area]]
[[landscape conservation] area]

As a solution to both problems, we represent all modifying compound constituents as members of a set-valued feature *MOD* at f-structure and thus leave the internal bracketing underspecified. We only keep track of the constituent's relative surface order by a precedence relation $<_{prec}$, cf. the partial f-structure for (12) in (14).[3]

(14)
$$\left[\begin{array}{ll} \text{PRED} & \text{'Fest'} \\ \text{MOD} & \left\{\begin{array}{l} \left[\begin{array}{ll} \text{PRED} & \text{'Kind'} \\ <_{prec} & \end{array}\right] \\ \left[\text{PRED} \quad \text{'Garten'}\right] \end{array}\right\} \end{array}\right]$$

2.1.2 Data Involving Coordination

When compounds are coordinated, they may be "elliptical", i.e., some part may be missing. Roughly, this happens whenever the coordinated compounds have some part in common. The identical part is then omitted in one of the compounds.[4] Let us have a look at some examples.[5]

The ellipsis may consist of one or more constituents. It may be located on the compound's right edge (i.e., it contains the head as in (15)) or on the left edge as in (16), or even on both the right and left edge simultaneously as in (17).

[3]The motivation for the precedence relation is to represent the linear order of the constituents at f-structure so that it can be exploited easily e.g. for transfer. Alternatively, this could be done by encoding the modifying constituents in a list.

[4]This type of ellipsis is not limited to compound nouns but occurs with complex verbs and adjectives as well; cf. also fn. 10.

[5]For better legibility, the part of the unreduced compound that corresponds to the ellipsis is set in italics. Furthermore, the location of the ellipsis is indicated by a hyphen according to German spelling rules.

(15) a. Wirtschafts- und Währungs*union*
economy and currency union
'economic and monetary union'

b. Bundes- und Land*tagsabgeordneter*
federation and state council representative
'member of the Bundestag and Landtag'

(16) a. *Fahr*lehrer und -schüler
drive instructor and pupil
'driving instructor and learner driver'

b. *Kraftfahrzeugsteuer*befreiung oder -ermäßigung
power vehicle tax exemption or reduction
'exemption or allowance of motor vehicle tax'

(17) a. *Daten*erfassungs- und -auswerte*station*
data recording and evaluate station
'station for data recording and data evaluation'

b. *Frauen*forschungs-, -bildungs- und -informations*zentrum*
woman research education and information centre
'centre for research, education and information concerning women'

In the remainder of this section, we only consider data of the type of "Right Periphery Ellipsis" as in (15) because they are far more frequent than data involving "Left Periphery Ellipsis" as in (16) and (17).[6]

2.2 The Analysis

Let us turn to the theoretical analysis now. To facilitate the representation, we introduce a special category *Nmod* for modifying constituents in compound nouns. *Nmod* is not part of syntax proper since composition is

[6] The *Huge German Corpus*, a collection mainly of newspaper texts, contains about 12 million sentences with 45 million nouns. 1/3 of the nouns are compounds. Among them, there are approximately 420,000 elliptical compound nouns of the following types:

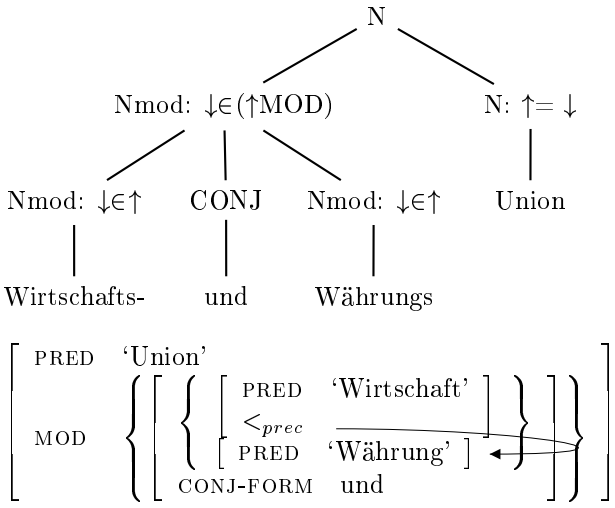
Right Periphery Ellipsis:	395,000
Left Periphery Ellipsis:	25,000
Right and Left Periphery Ellipsis mixed:	500

In fact, it has been argued that Left Periphery Ellipsis is not just the "mirror" of Right Periphery Ellipsis but represents a clearly different construction (Neijt 1987, Höhle 1991).

a process applying at the level of morphology. Nevertheless *Nmod* will be represented at c-structure to facilitate the representation of the basic idea underlying the different analyses.
At first glance, there are two ways of analysing the constructions:[7]

1. Compound constituents may be coordinated (base generation hypothesis).
   At the level of morphology, a coordination rule applies to *Nmod*; the coordinated *Nmod* categories in turn combine with a head noun to form a compound. The c- and f-structures for (15a) are sketched in (18).[8]

   (18)



2. Elliptical compounds result from a deletion process (deletion hypothesis).[9]

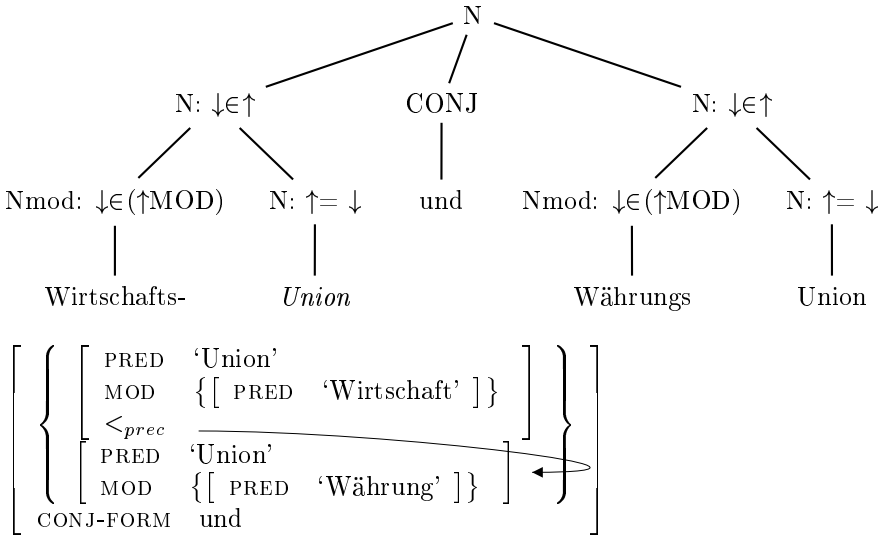
[7]Another analysis is proposed by Maxwell and Manning (1996). According to them, Right Periphery Ellipsis results from a special way of expanding the right hand side of a rule during parsing. In contrast to the analyses sketched in the text, the analysis by Maxwell and Manning (1996) cannot be modelled in XLE (the Xerox Linguistic Environment) since the special expanding mechanism is not implemented.

[8]A coordination rule applying at the level of morphology is independently motivated in German, cf. appendix 2.2.1.

[9]The term "deletion hypothesis" is borrowed from the work cited below. As a more theory independent term, it could be replaced by the term "gapping hypothesis".

According to this analysis, the c- and f-structures look as if the elliptical compound was unreduced, cf. (19). (The head missing on surface structure is set in italics at c-structure.)

(19)



There are various arguments in favour of the deletion hypothesis (Booij 1985, Neijt 1987, Höhle 1991):

1. Elliptical compounds are interpreted as if they were unreduced. This is easy to see with examples involving idiosyncratic compounds: *groß* normally means 'big, great' but in connection with kinship terms, it has the idiosyncratic meaning of 'one generation further'. These compounds are listed as such in the lexicon; nevertheless, the head may be missing as in (20).

   (20) Groß- und Urgroß*väter*
   big and original big fathers
   'grandfathers and great grandfathers'

2. In many compounds, there is a so-called linking morpheme ("Fugenmorphem") between two constituents, e.g. *-es-* and *-s-* in *Bund**es**tag**s**abgeordneter*. If the base generation hypothesis was correct, there should be no linking morpheme on the right edge of the elliptical compound, contrary to the facts illustrated by (21b).

(21) a. Bund und Land
'federation and state'

b. Bund**es**- und Land*tagsabgeordneter*
federation and state council representative
'member of the Bundestag and Landtag'

3. The putative conjuncts may consist of different categories. Instead of an *Nmod* modifier, an attributive *AP* might modify the head noun, cf. (22a). Likewise an *AP* by itself may represent the elliptical conjunct as in (22b). In both cases, the base generation hypothesis would require special coordination rules.

(22) a. im Verwaltungs- und technischen *Bereich*
in-the administration and technical sector
'in the administrative and technical sector'

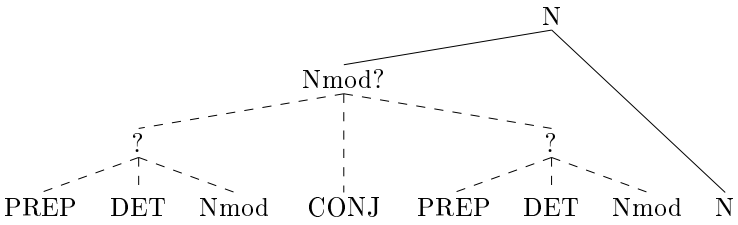
b. professionelle und Laien*künstler*
'professional and amateur artists'

4. Besides the coordinating conjunction there may be other material between the putative conjuncts: in (23a), the cardinal *14* intervenes between *Tages*, the conjunction *und*, and *Wochen*; in (23b), the preposition plus determiner *in der* intervene between *Markt*, the conjunction *als auch*, and *Plan*. To explain these patterns with the base generation hypothesis, one is forced to assume that syntactic categories like cardinals, prepositions, or determiners can form a constituent with morphological categories like *Nmod*, cf. the putative c-structure of (23b) in (23c). However, in terms of the deletion hypothesis, the elliptical and the unreduced compound simply may be embedded in other constituents which are standardly coordinated.

(23) a. 4 Tages- und 14 Wochen*zeitungen*
4 day and 14 week newspapers
'4 daily papers and 14 weekly papers'

b. in der Markt- als auch in der Plan*wirtschaft*
in the market and in the plan economy
'in the market economy and in the planned economy'

c.



To sum up, for each of the examples given above, the base generation hypothesis would have to stipulate special rules whereas under the assumption of the deletion hypothesis, the examples can be explained in a straightforward way.

To complete the picture, we finish this subsection by stating the conditions that have to be fulfilled for the deletion process to apply (following Booij 1985 and Höhle 1991):[10]

(24) Right Periphery Ellipsis:
A string s may be deleted if

- s is a sequence of one or more phonological words;[11]
- s is left-adjacent to a conjunction;[12]
- s is identical in sense and phonology to a string at the right periphery of the final conjunct;

[10] As it is formulated, (24) is not restricted to complex words but may apply to any string. This reflects the insight by Höhle (1991) that compound ellipsis is just a special instance of a more general deletion process. Another instance of this process would be (i) (= (13a) in Höhle 1991) where the verb *füttern* is deleted in the first conjunct (indicated by *e*).

(i) Heinz sollte den Hund *e* und Karl sollte den Kater füttern.
H. should the dog and K. should the cat feed
'Heinz should feed the dog, and Karl should feed the cat.'

At present, we only consider compound ellipsis for efficiency reasons: since in German, compound ellipses are indicated by a word-final hyphen, admitting ellipses can be restricted to words ending with a hyphen. Likewise we do not consider examples as in (22).

[11] A "phonological word" is either a word or a constituent of a complex word flanked by strong morpheme boundaries (Höhle 1982).

[12] In multiple coordination as in (17b), s is left-adjacent to a comma or a conjunction. Note that there are examples where the ellipsis is not left-adjacent to the conjunction, cf. (i). Furthermore, ellipses also occur in contexts without coordination, cf. (ii), (iii). We do not know of any discussion of these constructions in the literature. Probably some sort

- there is a remnant that, like its counterpart, can function as focus constituent (to give an example: in (15a), the remnant is *Wirtschafts* and its counterpart is *Währungs*).

Since the conditions refer to both syntactic as well as phonological structure, the cited authors – working in the framework of GB – conclude that Right Periphery Ellipsis presumably is a process in the PF component, relating S-structure and surface structure. In section 2.3 we will see how the properties listed in (24) can be captured in the framework of LFG.

Note, however, that there are data which cannot be subsumed under the deletion hypothesis, although they seem, at first sight, very similar to Right Periphery Ellipsis, cf. the following appendix.

2.2.1 Appendix

There are clearly base-generated data which seem very similar to Right Periphery Ellipsis, cf. (Toman 1985). Examples are given in (25) (= (7a)/(8a), respectively, in Toman 1985). In contrast to elliptical compounds, the examples in (25):

- do not have an unreduced counterpart that is semantically equivalent;
- usually do not have an explicit coordinating conjunction;
- have dashes between all constituents according to German spelling rules.

---

of parallelism requirement is at work.

i. die Stamm- mit neun und die Vorzugs*aktien* mit zehn *Mark* bedienen
the regulars with nine and the preference shares with ten mark serve
'to distribute an amount of nine marks for the ordinary shares and of ten marks for the preference shares'

ii. von Miet- in Eigentums*wohnungen*
from rent in property flats
'from rented flats into privately owned flats'

iii. Dreifelder- ersetzt die Zweifelder*wirtschaft*
three field replaces the two field cultivation
'three field system replaces two field system'

(25) a. Katz-und-Maus-Spiel
'cat and mouse game' ≠ 'cat game and mouse game'

b. Hals-Nasen-Ohren-Klinik
throat nose ear clinic
'ear, nose and throat clinic'

In the same way, example (26a) cannot be an instance of Right Periphery Ellipsis because there is no counterpart *∗Aufbewegung* in German (neither is there a noun *∗Abbewegung*). But note that *Auf und Ab* can be used as a noun, cf. (26b). So possibly (26a) is also an instance of a base-generated compound and simply violates spelling conventions – (26c) would then be the correct spelling.

(26) a. die Auf- und Abbewegung
the up and down movement
'the moving up and down'

b. ein Auf und Ab der Zinsen
an up and down of-the interests
'a going up and down of the interests'

c. die Auf-und-Ab-Bewegung

2.3 Implementation

We now consider implementation. We will start with some basic comments in section 2.3.1. In section 2.3.2 we will first sketch an implementation of the analysis based on the deletion hypothesis. However as we will see, this implementation has certain disadvantages. In a second step, we therefore sketch an implementation of the base-generated analysis. We will finish by comparing both solutions.

2.3.1 Basic Comments

Let us first recall the conditions that are to be captured by our implementation (based on (24)):

- compounds must be decomposed;
- the elliptical compound is left-adjacent to a conjunction, or put in other words: a hyphenated word must be followed by a conjunction;

- the unreduced conjunct must be a compound (this condition is based on the last point in (24));[13]
- at f-structure the elliptical compound "copies" parts of the f-structure from the unreduced compound.

The decomposition of compounds is done by a morphological module. A morphological analysis of (12) is shown in (27a), where the compound's constituents are separated by #. Hyphenated compounds are analysed as in (27b) (cf. (17b)). All constituents marked by *+Trunc* will be associated with the category *Nmod*.

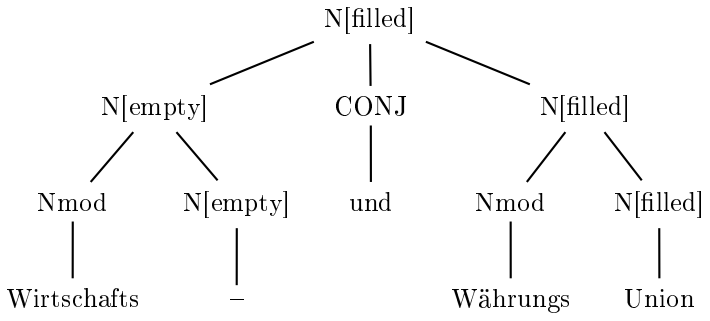
(27) a. Kindergartenfest
Kind+Noun+Trunc#Garten+Noun+Trunc#Fest+Noun+Common+...

b. Frauenforschungs-
Frau+Noun+Trunc#Forschung+Noun+Trunc#−+Hyphen

2.3.2 Writing Rules

For implementing the deletion analysis, we simply assume that a noun consists of arbitrarily many *Nmod* constituents plus either a head noun or a hyphen. Instead of representing this difference by a feature at f-structure, we use special c-structure categories provided by the XLE formalism. These categories consist of complex symbols containing parameters; the head status of nouns (*filled* for nouns containing a head, or *empty* for elliptical compounds) can be represented by specifying the parameter accordingly, as in (28).[14]

(28)

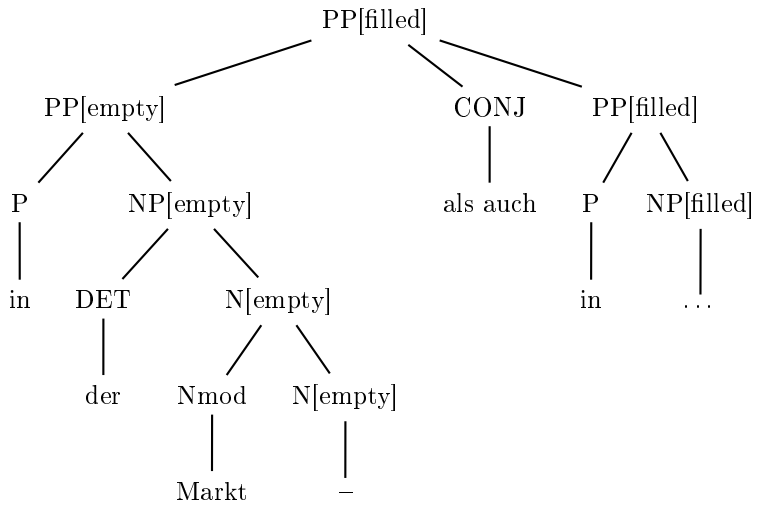


[13]We do not treat examples as in (22), cf. fn. 10.

[14]For a short introduction to complex categories cf. (Kuhn 1999, section 4.1).

Recall that the elliptical compound and its unreduced counterpart may be embedded in other constituents as in (23). In these cases, a chain of *empty*-marked categories dominates the hyphen, cf. the partial c-structure for (23b) in (29).

(29)



It is an important feature of the implementation sketched here that the head status of the compound is represented at c-structure. Otherwise, it would be difficult to formulate the adjacency condition, namely that hyphenated words (= $N[empty]$) must be followed by a conjunction – obviously a condition that has nothing to do with f-structure.
To encode the adjacency condition, the right hand side of all rules that possibly contain a category $X[empty]$ (like $N[empty]$, $PP[empty]$) is intersected with a regular expression which filters out all expansions of the rules containing $X[empty]$ followed by another constituent.[15]
Furthermore, the unreduced conjunct must be a compound. To check this condition, the f-structure projected by the head of the unreduced compound has to be located. Once it has been found its status as a compound can be checked by the existential constraint ($\downarrow MOD$). Finally, the head's *PRED* value has to be "copied" to the elliptical compound's f-structure.

Note that locating the head's f-structure is not a trivial task:

- The elliptical and the unreduced compound may be embedded by arbitrarily many constituents, cf. (23) and (30). The search mechanism

[15]Compare Kuhn's (1999, section 4.1) discussion of rule generalization by intersecting regular expressions – what he calls the "description-based approach".

therefore has to proceed via relatively unrestricted functional uncertainty paths.

(30) nicht aus dem Etat des Umwelt-, sondern aus dem des Entwicklungs*ministers*
not from the budget of-the environment but from that of-the development minister
'not from the budget of the Minister of Environment, but from the budget of the Minister of Development'

- The elliptical and the unreduced compound may occupy structurally different positions, cf. (31) (= (42) in Toman 1985).

(31) die Wiederaufnahme der Inlands- und des größten Teils der Auslands*flüge*
the resuming of-the internal and of-the the largest part of-the foreign flights
'the re-opening of internal flights and of the larger part of foreign flights'

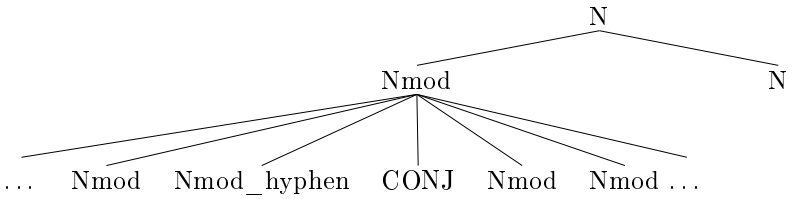
- There is no right-periphery restriction on the unreduced compound; i.e., it may be followed by constituents modifying the head, as in (32).

(32) die Jungmann- und die Autogen*straße* zwischen Omegabrücke und Jungmannstraße
the J. and the A. street between O. bridge and J. street
'Jungmann Street and Autogen Street between Omega bridge and Jungmann Street'

Thus the locating mechanism has to check first for information about gender, number, and case of the elliptical compound, supplied e.g. by a determiner. Then it has to look for any compound contained in the following conjunct. Finally, these compounds are checked for agreement in gender, number, and case with the elliptical compound.
Concluding this discussion, this implementation arrives at an analysis close to linguistic intuitions. However, it is based on rather complex rules and may be computationally expensive, since it involves relatively unrestricted functional uncertainty.

An alternative implementation is based on the base-generation hypothesis. Obviously with this implementation, not all of the instances are covered, cf. the discussion in section 2.2. Examples with intervening elements between conjunction and compounds as in (23) and (30) do not get an analysis. However, the simplest type of instances is captured, namely all examples without intervening elements. Their basic structure is sketched in (33).

(33)



For an assessment of this analysis, the following aspects have to be taken into account :

- without employing any additional mechanisms, all of the conditions stated in section 2.3.1 are fulfilled; i.e., there is no need of using regular expressions nor of checking and copying features via functional uncertainty. The conditions are fulfilled in the following way (for an illustrative example cf. the c- and f-structure in (18)):
  - a hyphenated word must be followed by a conjunction (adjacency condition);
    since in this analysis the hyphenated word and the conjunction are sister constituents, this condition can be encoded easily in the *Nmod* coordination rule:
    Nmod → *Nmod* Nmod_ hyphen CONJ Nmod+.*
  - the unreduced conjunct must be a compound;
    the only way to introduce hyphenated words is by the *Nmod* coordination rule sketched above. The *Nmod+* part constitutes the modifying constituents of the unreduced compound, hence it is automatically a compound.
  - the elliptical compound "copies" the head's *PRED* value;
    in this analysis all *Nmod* categories form one coordinated *Nmod* constituent, which is sister to their joint head constituent *N*. They therefore all share the same head.
- since functional uncertainty plays no role, the analysis is more efficient;

- as already mentioned, not all instances are covered;
- however, more than 95% of 395.000 instances occurring in the *Huge German Corpus* (cf. fn. 6) are instances of the simplest type, i.e., less than 5% of the data are not covered by this analysis.

Let us summarise the findings of this section:
Both analyses presented here have advantages and disadvantages. It depends on the grammar writer's objectives which implementation is to be preferred. Those who are interested in modelling linguistic insights as faithfully as possible, will stick to the first analysis. On the other hand, for those interested in parsing large corpora efficiently, the second analysis probably offers a good compromise.

3 Theoretical Implications

Implementing large scale grammars uncovers a number of issues that are of direct relevance for theoretical linguistics. We discuss two of the theoretical issues here which are a direct result of the ParGram project: the place of morphosyntactic information and the interpretation of underspecification.

3.1 Morphosyntactic Structure

This section discusses a proposal outlined in Butt, Niño, and Segond 1996 that a new level of grammatical representation represent morphosyntactic information in an attribute-value matrix structure, parallel to the f-structure.[16] This is referred to as the m-structure proposal.

In Butt et al. and here, the focus of the proposal is on how best to represent morphosyntactic tense in French, English, and German. Cross-linguistically, tense may be expressed morphologically by tense inflection and/or compositionally in syntax across languages and within languages. In this section we first discuss some basic properties of the English, French, and German tense systems and then propose an analysis to capture these properties.

[16]T. H. King would like to thank the other authors of this paper, Mary Dalrymple, Jonas Kuhn, and the audience of LFG99 for comments on this section.

3.1.1 Tense formation: synthetic vs. analytic

Two distinct types of verb/tense formation may contribute the same functional/semantic information to the f-structure. This is seen in (34) for French in which a synthetic form *parla* and an analytic form *a parlé* provide the same tense information. (The forms differ in style only.) Since the tense information is identical for the two forms and there are no other syntactically relevant differences, the f-structure should be similar. This is indeed the case, as seen below.

(34) a. *Il parla.* (he spoke-passé simple)

b. *Il a parlé.* (he spoke-passé composé)

c. $\begin{bmatrix} \text{PRED} & \text{'parler<(}\uparrow \text{ SUBJ)>'} \\ \text{TENSE} & \text{PAST} \end{bmatrix}$

In addition to differences within a language in the representation of a given tense, there are differences across languages between morphological and syntactic tense formation. That is, the same basic tense can be represented morphosyntactically in a number of ways. (35) shows the future tense of a typical transitive verb in English, German, and French. In French, there is a single verb form *tournera* which indicates both the main verb and the future tense. German and English both use an auxiliary to mark the future tense, in addition to the main verb. However, they differ in that in English the auxiliary immediately precedes the main verb, whereas in German the auxiliary is in second position and the main verb is in clause final position.

(35) a. *The driver will turn the lever.*

b. *Der Fahrer wird den Hebel drehen.* (German)

c. *Le conducteur tournera le levier.* (French)

Thus, the same tense can be represented both synthetically and analytically both within a language and cross-linguistically. Given that the functional information with respect to tense is identical regardless of the morphosyntactic representation, the question is what the f-structure corresponding to these forms should be.

3.1.2 Tense formation: well-formedness constraints

Next consider the tpes of constraints placed on the analytic formation of tense.[17] The occurence of auxiliaries are moderated by some constraints on the form and order in which auxiliaries in these languages can appear. Any analysis of tense must take these restrictions into account.

First consider restrictions on form, as in (36). In each case, each auxiliary specifies the form of the following auxiliary or verb. For example, in (36a) the modal *will* is followed by the base form of the auxiliary *have*. The auxiliary *have* in turn requires the perfect participle of the following verb *turned*. As seen in (36a), any change in these forms results in ungrammaticality. The same holds for German and French.

(36) a. *The driver will have turned the lever.*
**The driver will has turn the lever.*

b. *Der Fahrer wird den Hebel gedreht haben.* (German)
**Der Fahrer wird den Hebel drehen haben.* (German)

c. *Le conducteur aura tourné le levier.* (French)
**Le conducteur aura tourner le levier.* (French)

In addition to restrictions on form of the auxiliaries, there are also restrictions on their order, as in (37). As seen in (37a) the order in English is *modal-have-verb*. All other orders are ungrammatical, regardless of what form the auxiliaries and verb appear in. The same holds for German and French.

(37) a. **The driver have will turned the lever.*
**The driver has will turned the lever.*

b. **Der Fahrer wird den Hebel haben gedreht.* (German)
**Der Fahrer hat den Hebel gedreht werden.* (German)

c. **Le conducteur tourné aura le levier.* (French)
**Le conducteur tournera eu le levier.* (French)

[17] Here we do not consider the synthetic formulation of tense since that is part of the morphology proper and hence will not be relevant to the syntax, i.e., to the c-structure and f-structure.

The question is then how to account for these restrictions.[18] The classic LFG analysis of auxiliaries is to treat them like raising verbs, positing a PRED for each auxiliary which takes an XCOMP and a nonthematic subject (Falk 1984). This analysis is shown in (38) for the given English sentence.

(38)
$$\begin{bmatrix} \text{PRED} & \text{'have<(}\uparrow\text{ xcomp)>(}\uparrow\text{ subj)'} \\ \text{SUBJ} & \boxed{1}\ \begin{bmatrix} \end{bmatrix} \\ \text{XCOMP} & \begin{bmatrix} \text{PRED} & \text{'be<(}\uparrow\text{ xcomp)>(}\uparrow\text{ subj)'} \\ \text{SUBJ} & \boxed{2}\ \begin{bmatrix} \end{bmatrix} \\ \text{XCOMP} & \begin{bmatrix} \text{PRED} & \text{'flash<(}\uparrow\text{ subj)>'} \\ \text{SUBJ} & \boxed{3}\ \begin{bmatrix} \text{PRED} & \text{'pro'} \end{bmatrix} \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

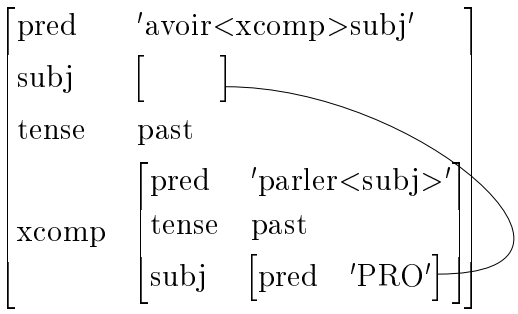
This type of analysis makes it relatively simple to state well-formedness constraints on both order and form of the auxiliaries since each auxiliary corresponds to a distinct f-structure. In (38) the well-formedness information is indicated by having a VFORM feature for each f-structure corresponding to an auxiliary or verb. However, this approach suffers from two main drawbacks. First, it requires a VFORM feature to appear in the f-structure despite such a feature not being relevant to the syntax other than for well-formedness reasons. That is, there is nothing else which depends on theses features since they are orthogonal to the tense aspect information. Second, and more importantly, each auxiliary has its own PRED. This means that the top level PRED is not that of the main verb and that the identity of structures for similar tenses within languages and across languages is lost. For example, the French examples in (39) have the same meaning, but would have two different f-structures under this type of analysis.

(39) a. Il *parla.* (he spoke-passé simple)

$$\begin{bmatrix} \text{pred} & \text{'parler<subj>'} \\ \text{tense} & \text{past} \\ \text{subj} & \begin{bmatrix} \text{pred} & \text{'PRO'} \end{bmatrix} \end{bmatrix}$$

[18] In this paper we are not concerned with the motivation behind these restrictions, just as we are not concerned with the exact morphological form of the synthetic tenses.

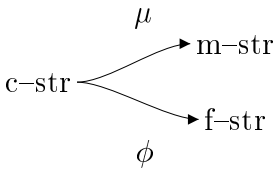
b. Il *a parlé*. (he spoke-passé composé)



We propose that these differences in morphosyntactic form should not be reflected in f-structure as they do not bear functional information or represent functional distinctions. However, this leaves us with the question of how to define a uniform (parallel) f-structure representation.

3.1.3 The m-structure proposal

In order to capture both the form and order restrictions without having XCOMPs in the f-structure, a new projection has been proposed: m(orpho-syntactic)-structure. M-structure is projected directly off the c-structure, in parallel to the f-structure. The basic idea is that auxiliaries will have a nested structure in the m-structure, but not in the f-structure.[19] Under this analysis tense auxiliaries are non-PRED-bearing elements (Bresnan 1999, King 1995, Schwarze 1996), and the distinction between analytic and synthetic tense formation is not reflected in functional terms (e.g., auxiliaries are not raising verbs and hence do not take XCOMPs).



Using the m-structure analysis, English, French, and German will have structurally identical f–structures for similar sentences, like those in (40). This "flat" f-structure is shown in (40d). In (40d) the main verb is the PRED of the top level f-structure and TENSE is indicated at that level.

[19]Modals are still analyzed as having PREDs and taking XCOMPs because they are assumed to have semantic content other than just tense and aspect information.

(40) a. *The driver will have turned the lever.*

b. *Der Fahrer wird den Hebel gedreht haben.* (German)

c. *Le conducteur aura tourné le levier.* (French)

d.
$$\begin{bmatrix} \text{PRED} & \text{'turn/drehen/tourner<(}\uparrow\text{SUBJ),(}\uparrow\text{ OBJ)>'} \\ \text{TENSE} & \text{FUTPERF} \\ \text{SUBJ} & \begin{bmatrix} \text{PRED} & \text{'driver/Fahrer/conducteur'} \end{bmatrix} \\ \text{OBJ} & \begin{bmatrix} \text{PRED} & \text{'lever/Hebel/levier'} \end{bmatrix} \end{bmatrix}$$

The morphosyntactic differences between the representation of tense in the languages are found in the m-structure. For the sentences in (40a) and (40b) English and German have a triply nested m-structure, with each auxiliary having a DEP(endent) feature. The VFORM features which were necessary for the raising verb analysis of auxiliaries are now placed in m-structure as features irrelevant for the f-structure syntax.

$$\begin{bmatrix} \text{FIN} & + \\ \text{AUX} & + \\ \text{DEP} & \begin{bmatrix} \text{AUX} & + \\ \text{VFORM} & \text{BASE} \\ \text{DEP} & \begin{bmatrix} \text{AUX} & - \\ \text{VFORM} & \text{PERFP} \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

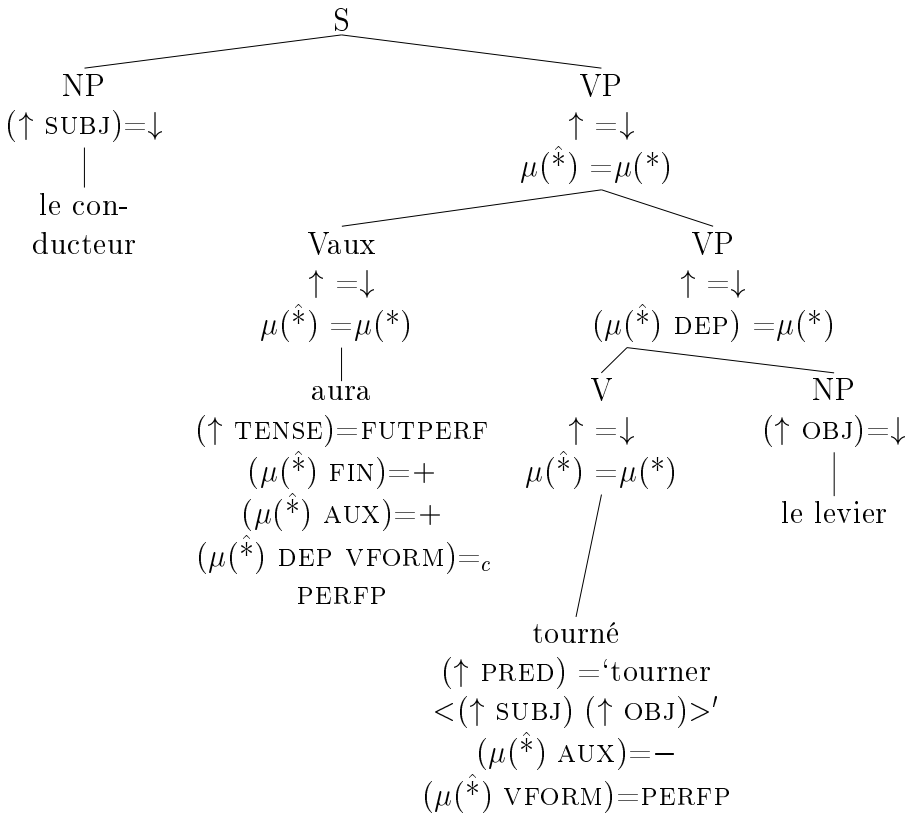
In contrast, French only has a doubly nested f-structure for (40c), reflecting the fact that French uses one less auxiliary to convey the same tense.

$$\begin{bmatrix} \text{FIN} & + \\ \text{AUX} & + \\ \text{DEP} & \begin{bmatrix} \text{AUX} & - \\ \text{VFORM} & \text{PERFP} \end{bmatrix} \end{bmatrix}$$

The details of tense formation in a parallel μ-projection are shown below for the French sentence (40c) *Le conducteur aura tourné le levier.* The basic c-structure involves an S composed of a subject NP and a VP. This VP comprises the auxiliary and another VP which in turn contains the main V and the NP object. The f-structure annotations use the familiary $\uparrow$ (up arrow) and $\downarrow$ (down arrow) annotations. The subject and object are mapped

to SUBJ and OBJ respectively. All of the VP and V nodes are marked $\uparrow = \downarrow$ indicating that all the information is relevant to the head f-structure.

The projection of the μ-structure is crucially not identical to that of the f-structure. This projection is marked by $\mu(\hat{*})$ to indicate the μ-projection of the mother node (cf. the up arrow for the f-structure) and $\mu(*)$ to indicate the μ-projection of the node itself (cf. the down arrow for the f-structure). The top level VP and first auxiliary are labelled $\mu(\hat{*}) = \mu(*)$ since they head the μ-structure for the sentence. However, the second VP is annotated $(\mu(\hat{*})$ DEP$) = \mu(*)$. This creates the dependent μ-structure seen above and allows for constraints on form to be included. These constraints are seen in the lexical entries for the auxiliary and main verb, namely that the VFORM of the auxiliary's DEP must be PERFP.



3.1.4 Summary

In this section we have presented the m-structure proposal introduced by Butt, Niño, and Segond 1996 which involves a m(orphosyntactic)-structure projected off the c-structure in parallel to the f-structure. The proposal of a further projection which contains morphosyntactic information irrelevant for f-structure, but necessary for language internal well-formedness is clearly a contribution of a theoretic nature within LFG. However, it grew directly out of a computational processing issue which first arose with the use of functional uncertainty (XCOMPs) for multiply embedded German auxiliaries. The issue came up with respect to German because German is a langauge with fairly free word order in which functional uncertainty is made use of more heavily than in English. As such, the processing issue associated with the interaction of the XCOMP treatment of auxiliaries and the flexible word order properties in German was the one to prompt a fresh look at the treatment of auxiliaries crosslinguistically.

However, some open questions and problems do remain with the m-structure proposal. One of these is what criteria distinguish between f-structure and m-structure features. Another concerns problems of long distance dependencies in a parallel architecture. Some of these are discussed in Frank and Zaenen 1998 who propose that m-structure not be projected off the c-structure, but off the f-structure. Finally, another paper which addresses the status of m-structure for the representation of tense is that of Dyvik 1999 in this volume.

3.2 Underspecification

Another area in which computational considerations have given rise to a reexamination of theoretical issues is the topic of underspecification. Underspecification is at the heart of much of linguistic thinking, particularly in the areas of phonology and morphology (see Ghini 1998 for an overview and argumentation on underspecification in phonological theories),[20] and psycholinguistic evidence makes a strong case for underspecification as a mechanism of representation in the mental lexicon. Consider the priming experiment conducted by Lahiri and van Coillie (1998), for example, which shows that while /m/ must be specified for place ([+labial]), /n/ can only be underspecified for place. Both the German word *Bahn* 'rail' and the non-word *Bahm* prime the

[20] One recent exception is the advent of Optimality Theory (Prince and Smolensky 1993)

semantically related word *Zug* 'train', showing that underlying *Bahn* cannot be specified for place (coronality, in this case). The word *Arm* 'arm', on the other hand, is the only one that can prime a semantically related word like *Bein* 'leg', indicating that /m/ must be underlyingly specified for place, blocking other possibilities.

(41)

Prime	**Target**	*Priming Effect*
Bahn 'rail'	Zug 'train'	+
*Bahm	Zug 'train'	+
*Arn	Bein 'leg'	−
Arm 'arm'	Bein 'leg'	+

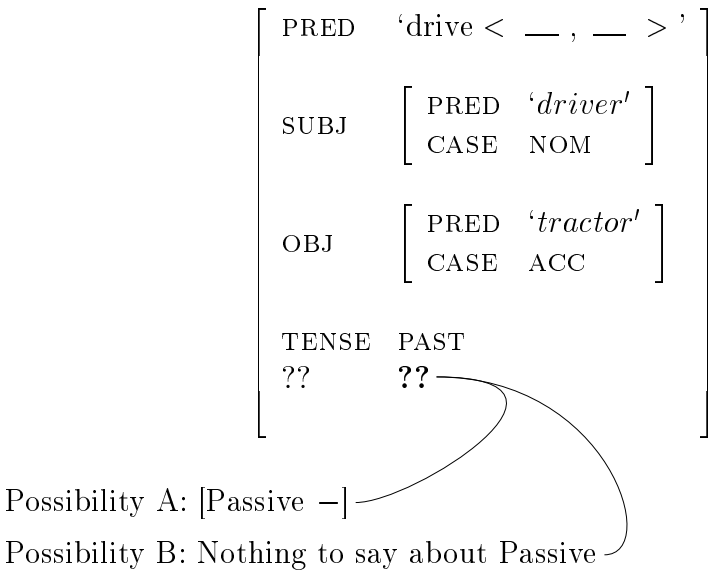
3.2.1 The Issue of Representation

Underspecification in LFG is interwoven with a theory of markedness in that the marked case is explicitly specified with a given feature, while the unmarked case can be left underspecified. The possibility of including underspecified representations would appear to be a point in favor for the LFG architecture, as it allows for a potential realization of the psycholinguistic insights. However, the formal tools available for the expression of underspecification actually give rise to an ambiguity.

Consider the f-structure in (42). The absence of the feature PASSIVE, for example, could either mean that the attribute-value matrix (AVM) is underspecified for this feature, or that the AVM should in fact be considered to be negatively specified for this feature.[21] That is, the f-structure in (42) could in principle either be interpreted as being not passive ([PASSIVE −]), or as leaving that option open: we simply do not know if the f-structure is passive or not and that information is also irrelevant for our current purposes.

[21]This point first came up in a discussion with Ron Kaplan in the spring of 1997.

(42)

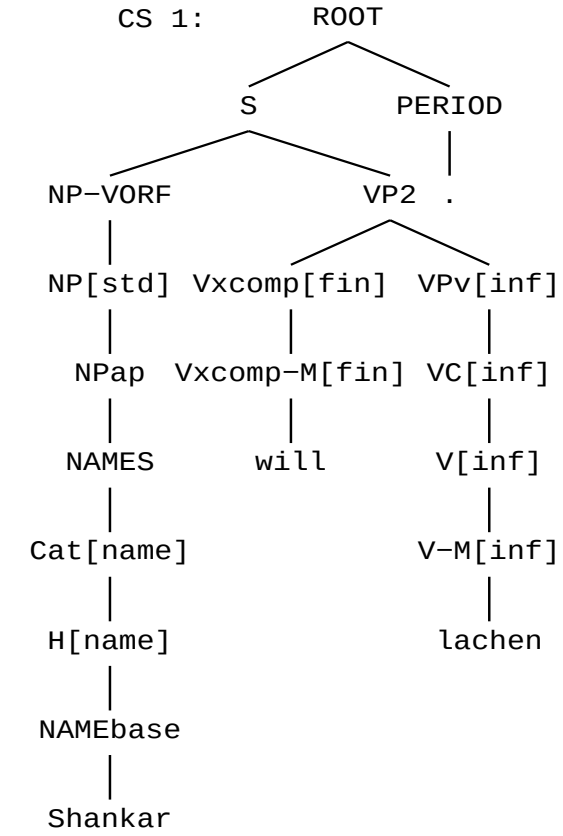


3.2.2 Actual Grammar Writing Practice

The problem, if one does indeed want to view it as such, is that the space of possibilities for specification are not as well understood and therefore not as circumscribed as in phonological analyses. The effect this has on actual grammar writing practice is that while the ParGram grammars make extensive use of macros and templates to express generalizations, there is also quite a bit of redundancy in coding and representations are generally overspecified for information. That is, feature specifications and constraints are often coded at multiple places or in multiple ways in the grammar.

As an example consider finiteness and tense in the German grammar. The c-structure and f-structure analyses currently assigned to the example in (43) are shown below.

(43) Shankar will lachen.
Shankar want.3.Sg.Pres laugh.Inf
'Shankar wants to laugh.'



"Shankar will lachen."

$$
\begin{bmatrix}
\text{PRED} & \text{'wollen<[54:lachen]>[0:Shankar]'} \\
\text{TNS-ASP} & [\text{MOOD indicative, TENSE pres}] \\
\text{VMORPH} & [\text{MODAL +}] \\
\text{SUBJ} & \begin{bmatrix} \text{PRED} & \text{'Shankar'} \\ \text{NTYPE} & [\text{PROPER name, NAME-TYPE unknown}] \\ 0 & \text{PERS 3, CASE nom, NUM sg} \end{bmatrix} \\
\text{XCOMP} & \begin{bmatrix} \text{PRED} & \text{'lachen<[0:Shankar]>'} \\ \text{PASSIVE} & - \\ \text{VMORPH} & [\text{FIN -, VFORM base}] \\ 54 & \text{SUBJ} \quad [0:\text{Shankar}] \end{bmatrix} \\
22 & \text{FIN +, STMT-TYPE declarative}
\end{bmatrix}
$$

As can be seen, the distinction between finite and infinite verbal forms is encoded at both levels of representation. At the c-structure, the appearance of the FIN vs. INF features is a direct consequence of the use of complex categories, which allow a parametrization of one and the same set of rules. At the f-structure, the FIN +/- features are used both for checking on wellformedness conditions, and for providing functional information as to finiteness. Additionally, the feature f-structure TENSE is linked to finiteness.

This type of redundancy in coding appears unattractive and avoidable. In fact, it is avoidable and the redundancy could well be eliminated in an overhaul of the grammar. Such overhauls have been conducted several times over the years in an effort to make the grammar more elegant. However, experience has shown that the elimination of redundancy makes the grammar less robust. One way of eliminating redundancy is to make one feature play more than one role in the grammar and to ensure that this feature appears only at one level of representation. While such elimination of redundancy would appear to be more elegant, it also tends to render the grammar less transparent as the role and function of a given feature is not immediately apparent any more (clever tricks are by definition not obvious). Furthermore, if one feature becomes "overloaded" in the sense that it is expected to play a number of interacting roles, the grammar is liable to break more easily when changes are introduced.

Thus, the experience gleaned over some years of grammar writing practice suggests that coding redundancy could actually be viewed as good grammar writing because it ensures greater robustness as grammars are continually changed and expanded.[22] Within ParGram, it has turned out that grammar writers have not made use of underspecification in order to represent (un)markedness though that option would have in principle been open to them. As shown below, in the f-structure analyses features like **passive** are always marked as either positive or negative in order to avoid the possibilities of overgeneration stemming from the ambiguity inherent in underspecified representations.[23]

[22]An interesting side question is whether coding redundancy may not also play a role in human parsing/generation. As we are able to recognize and parse speech from less than perfect input and under less than perfect conditions by using information stored in our mental lexicon as well as our world knowledge and expectations as to what is going to be said, it would seem that we do indeed rely on several different sources of information that might in fact contain overlapping types of information.

[23]Note that the ParGram grammars do make use of optimality marks via the o-

"Die Frau trinkt den Kaffee."

```
    [PRED    'trinken<[1:Frau], [186:Kaffee]>'    ]
    [TNS-ASP [MOOD indicative, TENSE pres]        ]
            [PRED   'Frau'                        ]
            [NMORPH [ADJ-AGR w]                   ]
    [SUBJ   [NTYPE  [GRAIN count]                 ]
            [SPEC   [SPEC-TYPE def, SPEC-FORM die]]
          1 [PERS 3, GEND fem, CASE nom, NUM sg   ]
            [PRED   'Kaffee'                      ]
            [NMORPH [ADJ-AGR w]                   ]
    [OBJ    [NTYPE  [GRAIN mass]                  ]
            [SPEC   [SPEC-TYPE def, SPEC-FORM die]]
        186 [PERS 3, CASE acc, GEND masc, NUM sg  ]
138 [FIN +, PASSIVE -, STMT-TYPE declarative      ]
```

"Der Kaffee wird getrunken."

```
    [PRED    'trinken<NULL, [1:Kaffee]>'          ]
    [TNS-ASP [MOOD indicative, TENSE pres]        ]
    [VMORPH  [AUX 'fin', FIN -, VFORM perfp]      ]
            [PRED   'Kaffee'                      ]
            [NMORPH [ADJ-AGR w]                   ]
    [SUBJ   [NTYPE  [GRAIN mass]                  ]
            [SPEC   [SPEC-TYPE def, SPEC-FORM die]]
          1 [PERS 3, GEND masc, CASE nom, NUM sg  ]
198 [PASSIVE +, FIN +, STMT-TYPE declarative      ]
```

Notwithstanding the above conclusion that redundancy may actually be desirable from a grammar writer's point of view, from a theoretical point of view the notion of underspecification still remains desirable. As such, the theoretical implication that can be drawn from the above discussion is that if the ambiguity inherent in the representation of underspecification gives rise to overgeneration and makes wellformedness checking difficult for the grammar writer, it would be nice if there were a formal notation that allowed

projection (Frank, King, Kuhn, and Maxwell 1998) to identify differing analyses as more and less marked.

for the representation of underspecification but did not result in unwanted ambiguities.

3.2.3 A Possible New Notation

One possibility for such a new notation might be to allow the AVMs to contain features that are not necessarily specified for a value. The f-structures in (44) and (45) show two different ways of allowing for this. In (44) the TENSE feature simply contains no value. In (45) the value of the TENSE feature is ANY. ANY in turn is a variable which stands for one of a clearly defined range of values. In the case of (45), ANY has been defined to only have the possible values of PRES, PAST or FUT.

(44)

$$\begin{bmatrix} \text{PRED} & \text{`drive} < \_\_ , \_\_ > \text{'} \\ \text{TENSE} & \end{bmatrix}$$

(45)

$$\begin{bmatrix} \text{PRED} & \text{`drive} < \_\_ , \_\_ > \text{'} \\ & \\ \text{TENSE} & \text{ANY} \end{bmatrix}$$

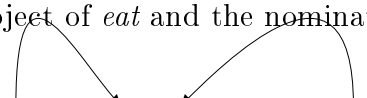
ANY = { pres | past | fut }

This notation differs from the current notation in that it actually allows for underspecification: in the current notation the markedness is represented by a presence of information, while unmarkedness is represented by the absence of information. No specfication (the absence of information), however, crucially differs from the notion of underspecification, in which only a constrained range of values could be used for specification. And it is precisely the fact that there is *some* information available in the f-structure, as opposed to none, that allows the grammar writer to be able to have a better chance of avoiding the kind of ambiguity and overgeneration problem sketched at the beginning of this section, while still being able to employ the theoretically desirably notion of underspecification.[24]

[24]Note that this notation is reminiscent of the use of typed feature structures. The notation as proposed here, however, is by no means intended to be as powerful as typed feature structures.

3.2.4 Feature Indeterminacy vs. Underspecification

The LFG formalism has recently been expanded to include the notion of feature indeterminacy (Dalrymple and Kaplan 1998). This was prompted by examples as in (46), in which the German pronoun *was* must simultaneously serve as the accusative object of *eat* and the nominative subject of *be*.



(46)	Ich	habe	gegessen	was	übrig	war.	(German)
	I.Nom	have	eaten	what	left	was	
			OBJ=ACC	?		SUBJ=NOM	

'I ate what was left.

The idea behind feature indeterminacy mainly consists of allowing the value of a feature to be a set with atomic values, rather than restricting the value of a feature to be a simple atomic value, or another AVM, as was the case in the past. The case specification of the German pronoun *was* 'what' can then be stated as the complex bundle of features shown in (47).

(47) was: $(\uparrow \text{CASE}) = \{\text{NOM,ACC}\}$

Checking for wellformedness now involves looking for the presence of an element in the set, as shown in the entry for the past participle of *eat* shown in (48).

(48) essen: $\text{ACC} \in (\uparrow \text{OBJ CASE})$

The introduction of feature indeterminacy at first glance appears as if it might also be the solution to the problem of representing underspecification more adequately. However, under the feature indeterminacy proposal the set of atomic values must be interpreted as representing a complex value. Under the underspecification proposal of the previous section, in contrast, only one atomic value can serve as the specification of the feature, not a complex value. Thus, the notion of underspecification must be very clearly differentiated from the notion of feature indeterminacy.

3.2.5 Summary

The notion of underspecification in LFG is interwoven with a notion of markedness that turns out to be inadequate in the light of experiences made

in the course of large-scale grammar writing. Rather than taking advantage of the theoretical notion of underspecification, the last few years of the ParGram project have shown that grammar writers instead tend towards a redundancy in both coding and representation in order to avoid unwanted ambiguities and overgeneration.

This section has tried to suggest that since the notion of underspecification is clearly theoretically desirable, the consequences from the computational experiences should be drawn and a new type of notation should be introduced into the formalism which would allow for the representation of underspecification while avoiding the problem of unwanted ambiguities and overgeneration.

4 From Parallel Grammar Development to Machine Translation

4.1 Introduction

One of several multilingual NLP applications that naturally emerge from the ParGram LFG Grammar Development Project is Machine Translation.[25] Most recently, Xerox PARC and XRCE Grenoble initiated a joint research project in Machine Translation, which builds on the linguistic and computational resources of the ParGram project.

This research approach towards machine translation focuses on innovative computational technologies which lead to a flexible translation architecture. Efficient processing of "packed" ambiguities in transfer – so-called *Chart Translation* (Kay 1999) – not only enables ambiguity preserving transfer. As opposed to standard processing schemes, which resort to early pruning of ambiguities for reasons of computational complexity, efficient processing of *packed ambiguities* in all modules of the translation chain – parsing, transfer and generation – allows for a flexible architectural design, open for various extensions which take the right decisions at the right time.

[25]The present section gives a summarization of the research conducted by the project members Marc Dymetman, Andreas Eisele, Anette Frank, Ron Kaplan, Martin Kay, John Maxwell, Paula Newman, Hadar Shemtov, Annie Zaenen, and the grammar writer team. A. Frank is grateful for helpful comments and suggestions from her colleagues. Remaining errors are her own.

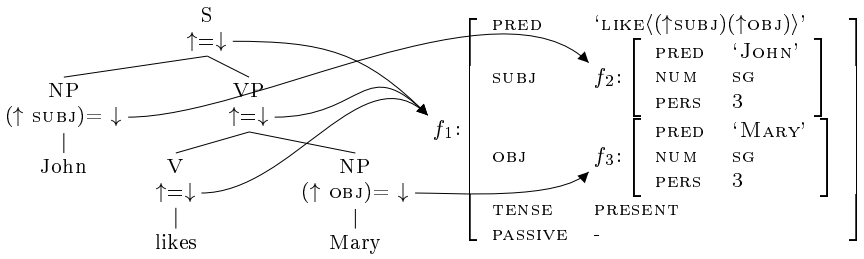


Figure 1: LFG projection architecture

In the present section we give a short overview of this approach, referring the reader to two recent publications, (Kay 1999) and (Frank 1999).[26] These papers provide more detailed discussion of the conceptual approach, its motivations and potential for high-quality translation, of the underlying computational technology, as well as the technical details of the translation component realized in an experimental translation prototype.

4.2 Parallel Grammar Development for Multilingual NLP

Lexical-Functional Grammar (Bresnan 1982) is particularly well suited for high-level syntactic analysis in multilingual NLP tasks. The LFG formalism assigns natural language sentences two levels of linguistic representation – a constituent phrase structure (c-structure) and a functional structure (f-structure) – which are related in terms of a functional projection, or correspondence function (ϕ–projection). The c-structure is a classical phrase structure tree that encodes constituency (dominance) and surface order (precedence). The f-structure is an attribute-value representation which encodes syntactic information in terms of morphosyntactic features (NUM, GEND, TENSE, etc.) as well as functional syntactic relations between predicates and their arguments or adjuncts. The two levels of representation are related via the correspondence function ϕ, which maps partial c-structures to partial f-structures (see Fig.1).

The separation between the surface oriented c-structure and the more abstract representation of functional syntactic properties makes it possible to provide syntactic descriptions for typologically diverse languages which may differ radically in terms of their c-structure properties (free word order, agglutinative languages, etc.), while relating them – via the ϕ-projection –

[26]The present section is a shortened version of (Frank 1999).

to the level of functional representation, which encodes functional syntactic properties that are largely shared across typologically distinct languages. This makes the f-structure representation provided by LFG-based analysis attractive for multilingual NLP tasks, such as Machine Translation.[27]

The ParGram project explores this potential of LFG as a framework for "parallel" syntactic description of various languages for multilingual NLP tasks. Large-scale LFG grammars have been developed for English, French and German, both under an engineering perspective (grammar engineering techniques for large-scale grammar development) and a linguistic research perspective (the development of principles for parallel f-structure representation across languages).[28] The aim of "parallel" grammar development is to provide common f-structure descriptions for similar constructions across distinct languages, by using a common description language, i.e. a common feature inventory. Due to this parallelism in the abstract f-structure representation, these "parallel" large-scale grammars provide important linguistic resources for the recently emerging project towards Machine Translation.

4.3 Computational technology for LFG-based NLP applications

Along with the ParGram project, Xerox PARC has developed the XLE (Xerox Linguistic Environment) system, a platform for large-scale LFG grammar development.

4.3.1 XLE as a grammar development platform

XLE as a grammar development platform comes with an interface to finite-state transducers for tokenization and morphological analysis (Kaplan and Newman 1997). A cascade of tokenizers and normalizers segments the input string into tokens, which are then "looked up" in finite-state morphological transducers. The integration of morphological analysis allows to automatically generate large LFG lexica for open class categories like nouns, adjectives, adverbs, etc. They are created by generic LFG lexicon entries which

[27]Together with the fact that LFG grammars are *declarative*, such that one and the same grammar can be used in analysis and generation.

[28]Both aspects are documented in (Butt, King, Niño, and Segond 1999) with further references on special issues in both areas.

specify f-structure annotations for morphological and lexical information provided by the morphology. While each grammar comes with hand-coded core LFG lexica for closed class "syntactic" lexical items, XLE supports integration and processing of large-size subcategorization lexica, which are extracted and converted from machine-readable dictionaries (Brazil 1997), or obtained by use of corpus analysis tools (Kuhn, Eckle-Kohler, and Rohrer 1998). Finally, a constraint ranking mechanism provided by XLE filters syntactic and lexical ambiguities (Frank, King, Kuhn, and Maxwell 1998). Distinct constraint ranking hierarchies can be specified for analysis vs. generation mode for a single LFG grammar. This allows us to account for a wide variety of constructions in analysis, while restricting generation from f-structures to default, or "unmarked" surface realizations.

4.3.2 Algorithms and architectures for high-performance unification-based grammar processing

The XLE platform integrates an efficient parser and generator for LFG grammars. The parsing and generation algorithms are based on insights from research into efficient processing algorithms for parsing and generation with unification-based grammars, in particular (Maxwell and Kaplan 1989), (Maxwell and Kaplan 1993), (Maxwell and Kaplan 1996) and (Shemtov 1997).

While context-free phrase structure grammars allow for parsing in polynomial time, grammar formalisms that in addition specify feature constraints can be NP-complete or undecidable, and parse in worst-case exponential or infinite time. However, the unification algorithm realized in XLE, described in (Maxwell and Kaplan 1996), automatically takes advantage of simple context-free equivalence in the feature space. As a result, sentences parse in cubic time in the typical case, while still being exponential in the worst case.

4.3.3 Contexted constraint satisfaction for processing of packed ambiguities

Contexted constraint satisfaction, a method for processing ambiguities efficiently in a "packed", chart-like representation, is of particular importance for the approach towards translation advocated in (Kay 1999) (described below in Section 4.4).

A major source of computational complexity with higher-level fine-grained syntactic analyses in general is the high potential for ambiguities, in particular with large-coverage grammars, where rule interaction plays an important role.

While disjunctive statements of linguistic constraints allow for a transparent and modular specification of linguistic generalizations, the resolution of disjunctive feature constraint systems is expensive, in the worst case exponential. Conjunctive constraint systems, on the other hand, can be solved by standard unification algorithms which do not present a computational problem.

In standard approaches to disjunctive constraint satisfaction, disjunctive formulas are therefore converted to disjunctive normal form (DNF). Conjunctive constraint solving is then applied to each of the resulting conjunctive subformulas. However, the possibly exponential number of such subformulas results in an overall worst-case exponential process. It is important to note that by conversion to DNF individual facts are replicated in several distinct conjunctive subformulas. This means that they have to be recomputed many times.

$$(a \vee b) \wedge x \wedge (c \vee d) \quad \begin{array}{c} \text{DNF} \\ \Rightarrow \end{array} \quad \begin{array}{l} (a \wedge x \wedge c) \\ \vee\ (a \wedge x \wedge d) \\ \vee\ (b \wedge x \wedge c) \\ \vee\ (b \wedge x \wedge d) \end{array}$$

(Maxwell and Kaplan 1989) observe that – though the number of disjunctions to process grows in rough proportion to the number of words in a sentence – most disjunctions are independent of each other. The general pattern is that disjunctions that arise from distinct parts of the sentence do not interact, as they are embedded within distinct parts of the f-structure. If disjunctions are independent, they conclude, it is in fact not necessary to explore all combinations of disjuncts as they are rendered in DNF, in order to determine the satisfiability of the entire constraint system.

On the basis of these observations, (Maxwell and Kaplan 1989) devise an algorithm for contexted constraint satisfaction – realized in the XLE parsing and generation algorithms[29] – that reduces the problem of *disjunctive* constraint solving to the computationally much cheaper problem of *conjunctive*

[29]For generation this holds for a new generation algorithm, designed by John Maxwell and Hadar Shemtov.

contexted constraint solving. The disjunctive constraint system is converted to a contexted conjunctive form (CF), a flat conjunction of implicational (contexted) facts, where each fact (a, b, x, ...) is labeled with a *propositional (context) variable* p, q or its negation,

$$(a \vee b) \wedge x \wedge (c \vee d) \quad \overset{\text{CF}}{\Rightarrow} \quad (p \rightarrow a) \wedge (\neg p \rightarrow b) \wedge x \wedge (q \rightarrow c) \wedge (\neg q \rightarrow d)$$

based on the **Lemma:**

$\phi_1 \vee \phi_2$ is satisfiable iff $(p \rightarrow \phi_1) \wedge (\neg p \rightarrow \phi_2)$ is satisfiable, where p is a new propositional variable.

Context variables p and their negations are thus used to specifiy the requirement that for a disjunction of facts $\phi_1 \vee \phi_2$ at least one of the disjuncts is true.

As can be seen in the above example, conversion to CF has the advantage that each fact appears only once, and thus will be processed only once. The resulting formula is a flat conjunction of implicational facts, which forms a boolean constraint system that can be solved efficiently, based on mathematically well-understood, general and simple principles (see (Maxwell and Kaplan 1989) for more detail).

After resolving the conjunctive implicational constraint system, the satisfiable constraints are kept in conjunctive contexted form, i.e. in a *packed* representation format, where disjunctive facts are not compiled out and duplicated. In the packed f-structure representation local disjunctions are directly accessible through their context variables. This is illustrated in Fig.2, the *packed f-structure chart* for the ambiguous sentence *Unplug the power cord from the wall outlet.* The PP-attachment ambiguity is spelled out in the corresponding *unpacked* c- and f-structure pairs of Fig.3.

The ambiguity resides in the attachment of the PP as a VP- or NP-adjunct. While this ambiguity affects the entire c-to-f-structure mapping down from the level of VP, it is captured in terms of the local disjunctive contexts a_1 and a_2 in Fig.2. All remaining f-structure constraints are conjoined in the TRUE context.

$$\begin{array}{ll} a_1 & \rightarrow (f_2 \ \text{ADJUNCT} \ \in) = f_{64} \\ a_2 & \rightarrow (f_{15} \ \text{ADJUNCT} \ \in) = f_{64} \\ TRUE & \rightarrow (f_2 \ \text{PRED}) = \text{`}unplug\langle(f_2 \ \text{SUBJ})(f_2 \ \text{OBJ})\rangle' \ldots \\ TRUE & \rightarrow (f_2 \ \text{OBJ}) = f_{15} \ldots \end{array}$$

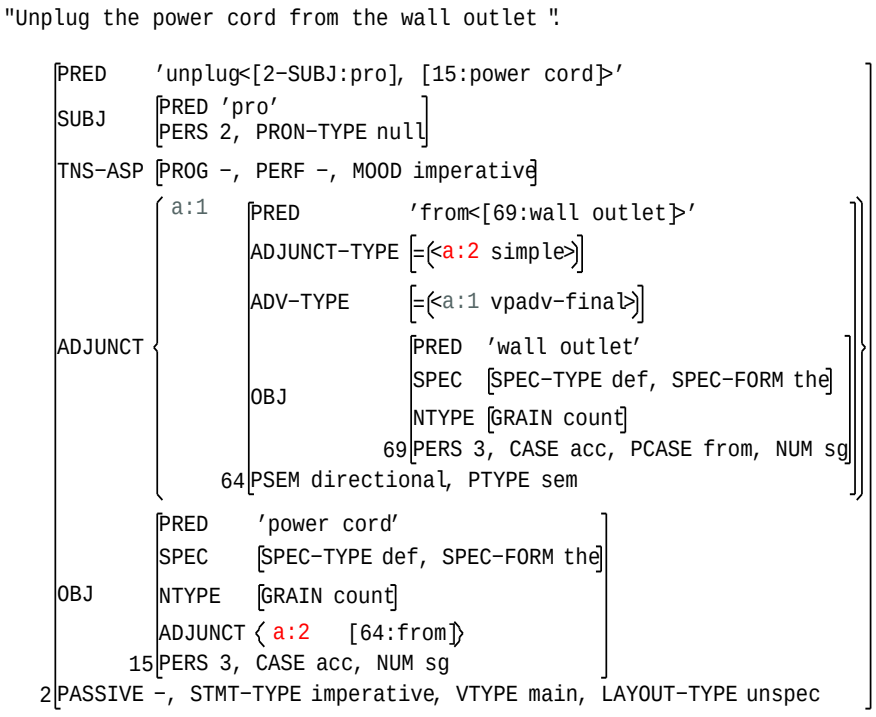


Figure 2: F-structure chart with disjunctive contexts $a1, a2$ for *Unplug the power cord from the wall outlet.*

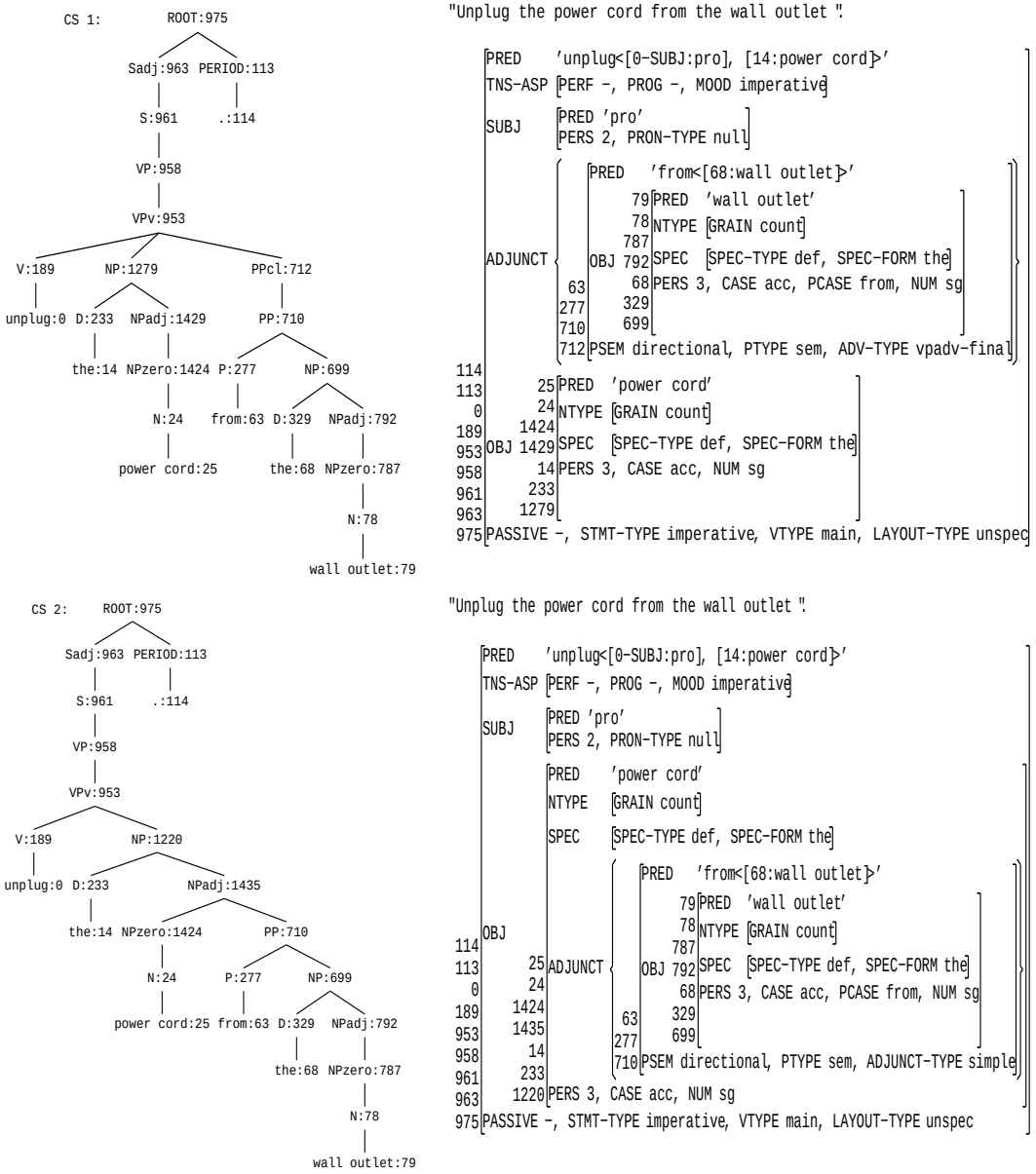


Figure 3: C-structure/f-structure ambiguity for *Unplug the power cord from the wall outlet.*

To sum up, disjunctive contexted constraint satisfaction allows for efficient computation of ambiguities in a "packed" representation format where local disjunctive facts are indexed with context variables. The resolution of conjunctive implicational constraints systems is mathematically simple and general, and can be computed efficiently. Ambiguous f-structures can be represented in one single *packed* representation, with local ambiguities indexed by their context variables.

4.4 An Innovative Translation Architecture

As we have seen, contexted constraint processing severely reduces the computational complexity in parsing and generation.[30] The conceptual approach towards translation taken in (Kay 1999) is therefore to take advantage of efficient processing of ambiguities also in translation, by generalizing contexted constraint processing and packed representation of ambiguities to all modules and interfaces of the translation chain. That is, processing of contexted disjunctive constraints is extended to the transfer module, which operates on contexted, i.e. packed representations.[31]

As a major advantage of this processing scheme, ambiguities which arise in each of the processing modules – parsing, transfer, and generation – can be efficiently propagated forward within the translation chain, as opposed to conventional translation architectures, where heuristic filters are applied early and throughout the translation chain to reduce the computational complexity arising from these multiplied ambiguities – yet at the risk of pruning correct solutions too early, on the basis of poor evidence.

A translation model that allows for efficient processing of ambiguities in packed representations is clearly in line with the conception of Machine Translation advocated early in (Kay 1980). Machine Translation being a highly complex and poorly understood problem, a translation system mustn't take decisions which it is not well-prepared to take. The overall value of automatic translation is enhanced if such alternatives are left undecided. Ambiguities can be propagated towards the end of the translation chain,

[30] In the current XLE implementation, generation still requires unpacking of f-structures. See however (Shemtov 1997) for a sound generation algorithm for efficient generation from packed structures. Generation from packed f-structures is currently being implemented in XLE.

[31] See (Dymetman and Tendeau 1998) for a variant of this approach. See also (Emele and Dorna 1998).

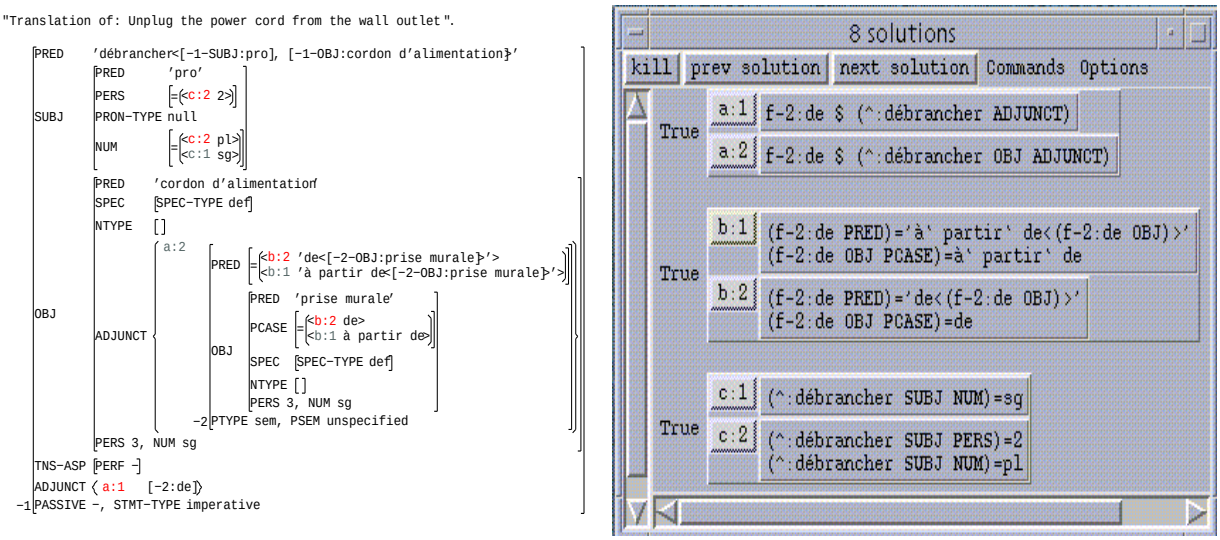


Figure 4: Ambiguity preserving translation – representation in chart and indexed by context variables

where examination on the output can provide useful hints for disambiguation. Moreover, the system must be designed in a flexible way, so as to allow for interactive guidance by a human. Interactive disambiguation can improve translation quality by avoiding chains of misguided decisions. Memory-based learning techniques can propagate human decisions for subsequent, similar decision problems.

The translation architecture that is grounded on efficient processing of ambiguities in all modules of the translation chain makes it possible to realize this conceptual approach: Since ambiguities can be carried along without harm, selections can be made flexibly, at various stages in the processing chain, whenever choices can be made on a justified basis, and with good evidence.

Moreover, transfer on packed representations of source language ambiguities allows for *ambiguity preserving* translation (Kay 1980), (Kay 1997) and (Shemtov 1997). Often an ambiguous sentence translates to a target sentence that displays the same ambiguity that is present in the source. As an example, reconsider Fig.2. The English sentence *Unplug the power cord from the wall outlet* can be translated into French as *Débranchez le cordon d'alimentation de la prise murale*, which displays the very same PP-attachment ambiguity that is present in the English sentence (see Fig.4). Even though

transfer introduces additional ambiguities for preposition choice (*de/à partir de*) and two morphological variants of the imperative (*Débranchez ...* vs. *Débrancher...*), it is possible – with packed ambiguity processing in parsing, transfer and generation – to carry over the PP attachment ambiguity to the target without unfolding. The additional ambiguities can be locally resolved after the transfer phase, or filtered from the set of generated target strings.[32]

4.5 The Transfer Component

The XLE system was extended to XTE (the Xerox Translation Environment) by addition of a transfer component that realizes *packed*, or *Chart Translation* (Kay 1999) in terms of contexted constraint processing.[33]

The transfer component is a fairly general rewrite system that works on unordered sets of terms. In our application the terms represent f-structures, but the system lends itself to processing any kind of semantic (term) representation.[34]

In our translation scenario, the transfer component takes a packed f-structure from source string analysis as input, and delivers a packed f-structure as output. The attribute-value representation from source string analysis is first converted to a flat unordered set of f-structure terms. F-structure attributes with atomic values (f_1ATTR)=VAL are rewritten as `attr(var(1), val)`; attributes which take an f-structure node as value (f_1ATTR)=f_2 are rewritten as `attr(var(1), var(2))`. The f-structure terms are internally associated with their respective context variables.

The unordered set of terms from source string analysis is input to a cascade of rewrite rules that continuously rewrite subsets of (source language) f-structure terms into (target language) f-structure terms. The order in which the transfer rewrite rules are stated is crucial. Each rewrite rule applies to the current input set of terms, and yields an output set of terms. The output set constitutes the input for the next rewrite rule. A rule cannot reapply to its own output, but it applies to each distinct instantiation of the specified

[32]Fig.4 shows a display where local ambiguities are indexed by their respective context variables. In interactive mode, some of these ambiguities (e.g. preposition choice) can be resolved by choosing among the disjunctive contexts. See Section 4.7 for various other disambiguation strategies.

[33]See (Kay 1999) for more detail.

[34]In much the same way as (Emele and Dorna 1996)'s relational transfer system, as shown in (Dorna, Frank, van Genabith, and Emele 1998).

left-hand side terms that occur in the input set.

The left-hand side of rewrite rules specifies a set of terms `p`. If all these terms match a term in the input set, the matched terms are eliminated from the input, and the terms specified on the right-hand side of the rule are added to the input set. The left-hand side of a rule may contain positive `+p` and negative `-p` terms. A rule that specifies a positive constraint only applies if this term matches some term in the input. A rule that specifies a negative constraint only applies if the term doesn't match any term in the input. Positive terms are not eliminated from the input.

There are obligatory (`==>`) and optional (`?=>`) rules. Stated in an informal way, an obligatory rule that matches the input rewrites the left-hand side terms into the right-hand side terms. An optional rule that matches the input creates two output sets: in one output set the left-hand side terms are rewritten into the right-hand side terms, as in the application of an obligatory rule; the second output set is identical to the input set. Subsequent rules consider all alternative output sets created by preceding optional rules.

The transfer component comes with a formalism that allows for a modular and generalized description of transfer patterns.
Macros and templates provide means for stating hierarchies of recurring patterns of terms or rules. They can be (recursively) referenced in the definition of transfer rules.
Templates define shorthands for optional, obligatory or unioned rewrite rules.

```
template_name(par1,par2)::  lhs {==>|?=>} rhs.
```

Macros define shorthands for sets of terms and can be referenced in left- or right-hand sides of transfer rules, rule templates or in other macros.

```
null_pron(A):= pred(A,pro), pron_type(A,null).
```

A union operator (`&&`) allows for union of two or more rewrite rules (or rule templates). A set of individual, modular rewrite rules can thus be flexibly combined, or unioned, to account for complex transfer patterns. If one of the unioned rules is an optional rule, the union will be an optional rule. If all of the rules are obligatory rules, the union is an obligatory rule.
Finally, left- or right-hand sides of transfer rules may state the empty set `0`. A rule `p ==> 0` with nonempty `p` deletes `p` from the input without introducing new terms in the output. Transfer rules with empty left-hand sides can be

used in conjunction with rule unioning and redefinition of rule templates, which allows for a compact definition of sequences of transfer rules.[35]

In the example below, we first define two vacuous rule templates `restriction` and `opt`, the latter being optional. By union (`&&`) with these rule templates, the main template for verb transfer `v2v` is turned into an optional rule, which is called by the entry for *open*, to define transfer to *soulever*. Subsequent redefinition of `opt` as a vacuous obligatory rule effectively *redefines* the `v2v` template – with which `opt` is unioned – as an obligatory rule for subsequent template calls. *open* is thus alternatively transferred to French *ouvrir*. Finally, `restriction` – and thus `v2v` – is redefined to apply only in the absence of the term `obj`, in which case a macro for reflexive marking is called on the right-hand side. In this way we correctly transfer *appear* to French *s'afficher*.

```
restriction(A) ::  0 ==> 0.

opt::  0 ?=> 0.

v2v(S,T)::  pred(A,S), +vtype(A,_) ==> pred(A,T)
&& opt
&& restriction(A).

v2v(open,soulever).

opt ::  0 ==> 0.

v2v(open,ouvrir).
v2v(unplug,débrancher).

restriction(A) ::  -obj(A,_) ==> refl(A).

v2v(appear,afficher).
```

4.6 A Transfer Grammar

With the extension of XLE to XTE, we built an experimental Translation Prototype that covers the entire translation chain, as a feasibility study for the newly designed transfer architecture, without aiming for large-scale coverage. A transfer grammar has been created for f-structure based transfer from English to French. As a corpus for translation we chose a text from

[35]Templates and macros can be redefined at any point in the grammar. A new definition takes effect as soon as it is encountered. When a redefinition takes place, this causes an implicit redefinition of any other template or macro in whose definition it partakes, directly or indirectly.

a technical domain, the user manual for the Xerox HomeCentre device. An arbitrary contiguous section of 99 sentences was selected from the corpus, the rationale being to ensure that a realistic collection of transfer problems would be encountered. We obtained correct translations for 94 sentences. Some example translations are given in the Appendix.[36]

The prospects of parallel grammar development were confirmed in that the definition of transfer is clearly facilitated for many linguistic constructions, due to structural parallelism at the level of f-structure. The definition of transfer for standard syntactic constructions involving adverbials, negation, conjunctions, prepositions, adjectives, relative clauses, comparative clauses, etc. could be reduced to simple lexical transfer rules, while the (possibly complex) syntactic feature structures are left untouched as long as parallelism is preserved. This is illustrated by the following transfer rules. Due to uniform f-structure encodings for the specification of mood, sentence type, coordination, adjunct structures, etc., transfer of negation and conjunctions is covered by simple lexical rules that apply irrespective of the syntactic (f-structure) context, i.e. whether the material appears in declarative or imperative sentences, in relative clauses or conditional sentences. The adjective transfer rule, e.g., covers transfer of adjectives irrespective of their degree of comparison, which is specified by additional features that can be carried over to the target without changes. Even complex relative clauses can be transferred by simple lexical transfer rules for relative pronouns, the f-structures for relative clauses being specified in parallel across the grammars.

```
adv2adv(S,T)::  pred(A,S) ==> pred(A,T).
   adv2adv(carefully,soigneusement).
   adv2adv(not, ne pas).

coord2coord(S,T)::  coord_form(A,S) ==> coord_form(A,T).
   coord2coord(and, et).
   coord2coord(then, puis).

conj2conj(S,T)::  conj_form(A,S) ==> conj_form(A,T).
   conj2conj(that, que).
   conj2conj(if, si).
```

[36]The transfer grammar currently consists of 171 structural transfer rules and 76 lexicalized transfer rule templates with approximately 5 entries per template. The transfer lexicon is restricted to the chosen corpus.

```
adj2adj(S,T)::  pred(A,S) ==> pred(A,T).
   a2a(good, bon).

+pron_type(A,rel), pron_form(A,that) ==> pron_form(A,qui).
```

There are of course transfer phenomena where source and target language exhibit distinct syntactic structures. Differences in argument structure or contextual restrictions on transfer are defined in a straightforward way in terms of lexicalized rule templates.

More complex structural changes can be stated in a fairly modular way by exploiting a specific characteristics of the underlying transfer algorithm, the strictly ordered application of transfer rules which operate directly on the output of previous rule applications, as opposed to a rewrite scenario where the input set of terms is continuously rewritten into a *distinct* output set.[37] This characteristics allows us to split up complex transfer definitions into a sequence of subsequent rules which define modular partial transformations in a stepwise fashion. Below we state the rule complex that defines nominalization, as in *removing the print head – retrait de la tête d'impression.*

The first rule performs lexical transfer of a verbal to a nominal predicate, jointly with a unioned (`&&`) rewrite operation that eliminates verbal and introduces appropriate nominal features (`nominal_to_verbal`). We further introduce the term `nominalized(A)`, as a handle, or trigger for the subsequent rules that will complete the nominalization transfer.

```
nominalization(SourceV,TargetN) ::
    pred(A,SourceV)
==> pred(A,TargetN), nominalized(A)
&&  verbal_to_nominal(A).

nominalization(clean, nettoyage).
nominalization(remove, retrait).
```

After lexical transfer, the argument structure of the originally verbal predicate is still unchanged. In nominalization, various kinds of relation changes occur, depending on the argument structure of the verb. A set of subsequent alternative transfer rules defines these various relation changes. Below we state the rule for active transitive verbs, where the object of the lexical head is rewritten into a prepositional adjunct; the non-overt subject argument is

[37]The latter conception is realized in the VerbMobil transfer component (Emele and Dorna 1996).

deleted. The rules for relation changes are restricted to nominalization contexts by the constraint `+nominalized(A)`. In a subsequent, final rule this predicate is deleted from the set of terms.

```
    +nominalized(A), passive(A,-),
    obj_arg(A,B), subj_arg(A,C), null_pron(C)
==> adjunct_x(A,D), prepsem(de,D,B).
```

The transfer algorithm realized in XTE's transfer component provides for a flexible way of encoding even complex structural changes in a modular and general way. Yet, the fact that any rule application changes the input for subsequent transfer rules requires a thorough organization of the transfer grammar.[38]

4.7 Future Directions and Conclusion

The translation architecture and processing techniques realized in XTE constitute only a first, basic step towards a Machine Translation system. However, the system is designed in such a way as to allow for innovative extensions. The way in which extensions will be integrated into the overall system design, the way in which the system's characteristics are further exploited will be decisive for its overall value.

Ambiguity Preservation and Disambiguation: XTE's system architecture allows for a flexible design for ambiguity handling. Ambiguity preserving translation is inherently supported by the translation architecture. Propagation of ambiguities without filtering can be exploited in multilingual translation by triangulation (Kay 1980), (Shemtov 1997) and for various techniques of ambiguity management (Shemtov 1997). Interfaces can be designed to allow for a flexible mixture of stochastic and interactive disambiguation, depending on specific applications and user needs. In the XTE prototype, a stochastic disambiguation model (Eisele 1999) assigns probabilistic weights to ambiguities present in source (and/or target) f-structures. Thresholds can be set for non-interactive n-best propagation of ambiguities. In interactive mode, probabilistically ranked structures can be inspected by the user, to select f-structures for further processing. Ranked alternatives can be selected by reference to local ambiguities, indexed by their context variables

[38] Obvious complications like translation cycles are dealt with in a straightforward way, by source and target language marking of lexical predicates.

(see Fig.4). This interactive model can be extended in various ways, e.g. to trigger user-intervention for predefined decision problems (which may be presented in terms of stochastic ranking), and by integration of learning and propagation techniques for human-approved ambiguity resolution.

Acquisition of Transfer Knowledge: Techniques for automatic acquisition of transfer lexica from bilingual corpora were proposed e.g. by (Turcato 1998). His approach can be generalized to packed f-structure processing, and seamlessly integrated within the XTE translation architecture. Extensions towards alignment models as proposed by (Grishman 1994) can be exploited for automatic acquisition of transfer rules.

Statistical Methods and Robustness: Further extensions are required for transforming an MT prototype into a powerful and robust large-scale MT system. Knowledge-, or rule-based systems are not well-prepared to process unseen data not captured by grammar or transfer rules. Interfacing statistical processing models with rule-based systems is a challenge worth to explore. Corpus-driven stochastic parsing models in the LFG framework (Bod and Kaplan 1998), with possible extensions towards transfer architectures (Way 1998) take first steps into this direction.

Appendix: Some (Disambiguated) Example Translations

To keep your HomeCentre in good operating condition, you need to perform periodic maintenance tasks.
Pour assurer le bon fonctionnement de votre HomeCentre, vous devez effectuer périodiquement des tâches d'entretien.

Removing and Replacing the Paper Cassette
Retrait et mise en place de la cassette papier.

Before you add paper, make sure that the paper matches the paper size settings in Windows.
Avant d'ajouter du papier, assurez-vous que le papier correspond au format de papier sélectionné dans Windows.

Keep in mind that you can't use paper that is wider than 8 inches in the HomeCentre.
N'oubliez pas que vous ne pouvez pas utiliser du papier qui dépasse 8 inches

dans le HomeCentre.

Fan the paper and put up to 125 sheets into the paper tray.
Ventilez le papier et placez jusqu'à 125 feuilles dans le plateau de départ papier.

Make sure that the green carriage lock lever is still moved all the way forward before you reinstall the print head.
Assurez-vous que le levier vert de verrouillage du chariot est toujours repoussé complètement vers l'avant avant de remettre la tête d'impression en place.

You can clean the print head only when the green LED is lit or while printing.
Vous ne pouvez nettoyer la tête d'impression que lorsque le voyant vert est allumé ou pendant l'impression.

Calibrating the scanner restores a sharp image quality and helps the scanner capture clear images and text.
L'étalonnage du scanner rétablit une bonne qualité d'image et permet au scanner de produire des images et des textes nets.

You should print a test page each time you move the HomeCentre or replace an ink cartridge.
Il est conseillé d'imprimer une page de test chaque fois que vous déplacez le HomeCentre ou que vous remplacez une cartouche d'encre.

References

Bod, R. and R. Kaplan (1998). A probabilistic corpus-driven model for lexical-functional analysis. In *Proceedings of COLING/ACL 98, Canada*.

Booij, G. E. (1985). Coordination reduction in complex words: a case for prosodic phonology. In H. van der Hulst and N. Smith (Eds.), *Advances in Nonlinear Phonology*, pp. 143–160. Dordrecht: Foris.

Brazil, K. (1997). Building subcategorisation lexica for an LFG grammar of French. Technical report, Xerox Research Centre Europe, Grenoble. Summer Internship Report.

Bresnan, J. (Ed.) (1982). *The Mental Representation of Grammatical Relations*. Cambridge, MA: MIT Press.

Bresnan, J. (1999). Lexical-functional syntax. Ms., Stanford University. To appear, Blackwell.

Butt, M., T. H. King, M.-E. Niño, and F. Segond (1999). *A Grammar Writer's Cookbook*. Stanford: CSLI.

Butt, M., M.-E. Niño, and F. Segond (1996). Multilingual processing of auxiliaries within LFG. In *Proceedings of KONVENS 96, Bielefeld*, pp. 111–122. Mouton de Gruyter.

Dalrymple, M. and R. Kaplan (1998). Feature indeterminacy and feature resolution in description-based syntax. Ms., Xerox PARC.

Dorna, M., A. Frank, J. van Genabith, and M. C. Emele (1998). Syntactic and semantic transfer with f-structures. In *Proceedings of COLING 98*.

Dymetman, M. and F. Tendeau (1998). An algorithm for the transfer of packed linguistic structures. Unreleased internal Xerox publication.

Dyvik, H. (1999). The case of modals. In M. Butt and T. H. King (Eds.), *Proceedings of LFG 99*. CSLI Online Publications. `http://csli-publications.stanford.edu/LFG4/`.

Eisele, A. (1999). *Representation and stochastic resolution of ambiguity in constraint-based parsing*. Ph. D. thesis, University of Stuttgart.

Emele, M. C. and M. Dorna (1996). Efficient implementation of a semantic-based transfer approach. In *Proceedings of ECAI 96, Budapest, Hungary*.

Emele, M. C. and M. Dorna (1998). Ambiguity preserving machine translation using packed representations. In *Proceedings of COLING 98, Canada*.

Falk, Y. (1984). The english auxiliary system: A lexical-functional analysis. *Language 60*(3), 483–509.

Frank, A. (1999). From parallel grammar development towards machine translation. In *Proceedings of MT Summit VII. "MT in the Great Translation Era", Kent Ridge Digital Labs, Singapore*, pp. 134–142.

Frank, A., T. H. King, J. Kuhn, and J. Maxwell (1998). Optimality theory style constraint ranking in large-scale LFG grammars. In M. Butt and T. H. King (Eds.), *Proceedings of LFG 98*. CSLI Online Publications. `http://csli-publications.stanford.edu/LFG3/`.

Frank, A. and A. Zaenen (1998). Tense in LFG: Syntax and morphology. Ms., Xerox Research Centre Europe, Grenoble.

Ghini, M. (1998). *Aymmetries in the Phonology of Miogliola*. Ph. D. thesis, University of Konstanz.

Grishman, R. (1994). Iterative alignment of syntactic structures for a bilingual corpus. Workshop on Very Large Corpora 1994.

Höhle, T. N. (1982). Über Komposition und Derivation: zur Konstituentenstruktur von Wortbildungsprodukten im Deutschen. *Zeitschrift für Sprachwissenschaft 1*, 76–112.

Höhle, T. N. (1991). On reconstruction and coordination. In H. Haider and K. Netter (Eds.), *Representation and Derivation in the Theory of Grammar*, Volume 22 of *Studies in Natural Language and Linguistic Theory*, pp. 139–197. Dordrecht/Boston/London: Kluwer.

Kaplan, R. and P. Newman (1997). Lexical resource reconciliation in the Xerox linguistic environment. In D. Estival, A. Lavelli, K. Netter, and F. Pianesi (Eds.), *Computational environments for grammar development and linguistic engineering. Proceedings of a workshop sponsored by ACL, Madrid, Spain*, pp. 54–61.

Kay, M. (1980). The proper place of men and machines in language translation. *Machine Translation, Kluwer, Netherlands 12*, 3–23. Xerox PARC Working Paper, 1980.

Kay, M. (1997). It's still the proper place. *Machine Translation, Kluwer, Netherlands 12*, 35–38.

Kay, M. (1999). Chart translation. In *Proceedings of MT Summit VII. "MT in the Great Translation Era", Kent Ridge Digital Labs, Singapore*, pp. 9–14.

King, T. H. (1995). *Configuring Topic and Focus in Russian*. CSLI Publications.

Kuhn, J. (1999). Towards a simple architecture for the structure-function mapping. In M. Butt and T. H. King (Eds.), *Proceedings of LFG 99*. CSLI Online Publications. `http://csli-publications.stanford.edu/LFG4/`.

Kuhn, J., J. Eckle-Kohler, and C. Rohrer (1998). Lexicon acquisition with and for symbolic NLP-systems – a bootstrapping approach. In *Proceedings of the First International Conference on Language Resources and Evaluation (LREC98), Granada, Spain*, pp. 89–95.

Lahiri, A. and S. van Coillie (1998). Sandhi — psycholinguistic evidence for underspecification. Ms., University of Konstanz.

Maxwell, J. T. and R. M. Kaplan (1989). An overview of disjunctive constraint satisfaction. In *Proceedings of the International Workshop on Parsing Technologies*, pp. 18–27.

Maxwell, J. T. and R. M. Kaplan (1993). The interface between phrasal and functional constraints. *Computational Linguistics 19*(4), 571–590.

Maxwell, J. T. and R. M. Kaplan (1996). Unification-based parsers that automatically take advantage of context freeness. Paper presented at the LFG 96 Conference, Grenoble, France. Ms. Xerox PARC.

Maxwell, J. T. and C. D. Manning (1996). A theory of non-constituent coordination based on finite-state rules. In M. Butt and T. H. King (Eds.), *Proceedings of LFG 96*. CSLI Online Publications. `http://csli-publications.stanford.edu/LFG1/`.

Neijt, A. (1987). Coordination reduction in dutch morphology. *Grazer Linguistische Studien 28*, 91–101.

Prince, A. and P. Smolensky (1993). Optimality: Constraint interaction in generative grammar. Ms., Rutgers University. To appear, MIT Press.

Schwarze, C. (1996). The syntax of romance auxiliaries. In M. Butt and T. H. King (Eds.), *Proceedings of LFG 96*. CSLI Online Publications. `http://csli-publications.stanford.edu/LFG1/`.

Shemtov, H. (1997). *Ambiguity Management in Natural Language Generation*. Ph. D. thesis, Stanford University.

Toman, J. (1985). A discussion of coordination and word-syntax. In J. Toman (Ed.), *Studies in German Grammar*, Volume 21 of *Studies in Generative Grammar*, pp. 407–432. Dordrecht: Foris.

Turcato, D. (1998). Automatically creating bilingual lexicons for machine translation from bilingual text. In *Proceedings of COLING 98*.

Way, A. (1998). A hybrid architecture for robust MT using LFG-DOP. *Journal of Experimental and Theoretical Artificial Intelligence*. To appear.

Miriam Butt
University of Konstanz
FG Sprachwissenschaft
Postfach 5560 <D186>
D-78457 Konstanz
Germany
http://www.ling.uni-konstanz.de
/pages/home/butt/
miriam.butt@uni-konstanz.de

Stefanie Dipper
University of Stuttgart
IMS
Azenbergstr. 12
D-70174 Stuttgart
Germany
http://www.ims.uni-stuttgart.de/~dipper/
dipper@ims.uni-stuttgart.de

Anette Frank
Xerox Research Centre Europe
6 chemin de Maupertuis
F-38240 Meylan
France
http://www.xrce.xerox.com
/people/frank
Anette.Frank@xrce.xerox.com

Tracy Holloway King
NLTT/ISTL, Xerox PARC
3333 Coyote Hill Road
Palo Alto, CA 94304
USA
http://www.parc.xerox.com
/istl/members/thking/
thking@parc.xerox.com