Towards a simple architecture for the structure-function mapping

Jonas Kuhn

Institut für maschinelle Sprachverarbeitung
Universität Stuttgart

Proceedings of the LFG99 Conference

University of Manchester, UK

Miriam Butt and Tracy Holloway King (Editors)

1999
CSLI Publications
http://www-csli.stanford.edu/publications/

Abstract

Recent theoretical work in LFG proposes a principle-based account of the mapping from c-structure to f-structure. This paper discusses how this account can be applied in a computational context. The assumption of a level of abstraction over rules is required. There are at least two ways of introducing such abstractions without having to alter the architecture of LFG much.

Although the generality of the principle-based account is highly desirable – not just from the theoretical, but also from an engineering point of view –, a naive implementation of the principles will run into complexity problems. However, for most principles, a formulation is possible that avoids these problem without giving up generality of specification.

1 Introduction

This paper is primarily an investigation into how recent developments in the theoretical LFG literature – endocentric principles of c-structure and the mapping to f-structure (cf. e.g. Bresnan 1998a) – could be exploited in ongoing efforts of implementing sizeable grammars on the computer (like the parallel grammar project ParGram, cf. Butt et al. 1999b; Butt et al. 1999a). At the same time however, more general conceptual issues regarding the architecture underlying the theory are raised.

The main points of this paper – besides a remark on the Off-line Parsability restriction – are: (i) There are at least two ways of introducing a level of abstraction over rules for expressing principles, without having to alter the architecture of LFG much. (ii) To a certain degree, a computationally harmful proliferation of c-structure analyses can be practically avoided without losing generality – by expressing restrictive principles at the level of c-structure rules (which need not necessarily be confined to purely categorial information). For some principles, the rule-level may indeed be the theoretically adequate level of application. (iii) The checking of economy principles requires an architecture allowing for comparison of alternative analyses, similar as in Optimality Theory (OT). When a restriction on the domain of comparison is assumed however, a precompilation with a similar effect as the strategy under (ii) becomes possible.

In sec. 2, I try to motivate why recent insights of the theoretical LFG literature may indeed solve a problem of large-scale grammar development, illustrating the issue and the relevant parts of the theory with an example from German – the mixed category analysis of adjectival participles (Bresnan 1997). Sec. 3 addresses the formulation of the Off-line Parsability restriction, which guarantees decidability; the mixed category analysis reveals the need for a slight modification of the standard formulation of the restriction – adding a criterion that is not c-structure internal but talks about the mapping to f-structure also. In sec. 4, I discuss ways of rendering the general principles of the theory in a computational implementation, working with an appropriate level of abstraction. I furthermore point out a complexity problem due to the great number of c-structures generated, which can however be reduced by reformulating principles as c-structure based restrictions. Sec. 5 finally states the different nature of the complications arising in the checking of comparison-based principles like Economy of Expression. Pointing out similarities with OT, I briefly discuss how cross-analysis comparison could be implemented computationally (for a more extensive discussion, the reader is referred to Kuhn 1999b).

2 Motivation

An important goal of the theoretical LFG literature of recent years has been to develop a principled account of the c-structure/f-structure mapping (cf. the initialization in Kroeger 1993 and the overview in Bresnan 1998a). Restrictions of what shape c-structure rules and their functional annotations can take are clearly required in an explanatory account of the human language faculty. The formalism itself would license linguistically absurd rules like (1).

(1) *Linguistically absurd rule*

DP	→	V′	A*	(PP)
		↓ ∈ (↑COMP ADJUNCT)	(↑SUBJ)=↓	↑=↓

The import of such meta-level rule restrictions for efforts in large-scale grammar writing may be less clear. The traditional approach of a rule-by-rule grammar specification has been applied very successfully to implement sizeable fragments (cf. e.g. the grammars of the ParGram project, Butt et al. 1999b; Butt et al. 1999a). Rule specification in such a framework is not at all arbitrary, but is subject to background conventions. Nevertheless, higher-level generalizations are mostly inexplicit in the grammar code, i.e. rules R_1 to R_n are all stated in accordance with a particular generalization G, but there is no unique location in the grammar code where G is specified (in sec. 2.1, a concrete example will be given).[1]

This can become problematic as the grammars get more and more extensive, in particular if the agenda of grammar extension is determined by the need to cover constructions as they occur in text corpora. Such extensions require the modification of previously specified rules, and typically, a relatively small set of frequently used rules will undergo a lot of changes, whereas other rules remain more or less untouched. Now, if certain generalizations are inexplicit they are very likely to get corrupted in this process: for instance, say that rule R_1 is extended to deal with certain data not covered so far; although conceptually all phenomena underlying the generalization G are affected by this change, the grammar writer fails to modify *all* of the rules R_2 to R_n (the reasons for skipping a rule may vary: some rule may be overlooked by chance, or the new effect of G on a particular rule may be considered irrelevant for practical application of the grammar – an unnecessary increase of the search space is often avoided). Since in a real system, very many rules and generalizations are involved, the overall behaviour can get hard to predict and it will become increasingly expensive to make even small modifications.

Given this situation, it seems that grammar writing could benefit a lot from an explicit account of linguistic principles underlying the c-structure/f-structure mapping – even from a purely software engineering point of view. However, since grammar writing projects are usually long-term efforts involving considerable resources, it is clear that a fundamental redesign would take a relatively long time, so there have to be good reasons to expect major improvements.

In this light, the generalized structure-function mapping account is problematic since its straightforward implementation is inefficient when compared to the traditional rule-by-rule specification. This is unfortunate since one of the main attractions of the LFG framework in language processing is the availability of highly developed algorithms that make the processing very fast (given the expressiveness of the formalism). In this paper, I will illustrate how the inefficiency arises and discuss ways of avoiding it without giving up the goal of expressing generalizations more explicitly.

[1] It should be noted that practical grammar writing environments like the Grammar Writer's Workbench (Kaplan and Maxwell 1996) have always provided various means of abstraction (like templates and macros) to make generalizations explicit; however the relation of these mechanism to linguistic theory has been somewhat unclear.

2.1 Generalizations in the structure-function mapping theory

An example for a generalization that is rather hard to make explicit in terms of traditional rules is the parallel behaviour of different categories with respect to their complements, as observed for so-called "mixed categories" (Bresnan 1997). I will illustrate the problem with some German data and go on with a sketch of Bresnan's (1997) principle-based analysis with the concept of extended c-structure heads, thereby introducing the relevant concepts of the theory. (Another phenomenon of German that could have been mentioned as benefiting a lot from extended head theory is of course the positioning of the finite verb: verb last vs. verb second.[2])

Although adjectival participles in German carry agreement information like ordinary adjectives, and their phrase projection is distributed like normal attributive adjective phrases (compare (2a) and (2b)), the phrase internal behaviour of the participles resembles the behaviour of verbs in VPs, i.e., the participles take the same types of complements and modifiers as verbs (compare (2a) and (2c)).

(2) a. ein mehrere Sprachen sprechender Mann
a.M.SG.NOM *several languages speaking*.M.SG.NOM *man*.M.SG
'a man speaking several languages' (Bresnan 1998a, 187)

b. ein alter Mann
a.M.SG.NOM *old*.M.SG.NOM *man*.M.SG

c. weil er mehrere Sprachen spricht
because he several languages speaks
'because he speaks several languages'

Example (3a) – taken from a newspaper corpus – shows that even complicated examples with the participle of modal verbs occur in real text, which means that the underlying generalization is in fact relevant for language processing ((3b) again gives a parallel verb phrase example).

(3) a. undurchsichtig bleiben wollenden Kunden
devious stay.INF *wanting*.PL.DAT *customer*.M.PL.DAT
'customers wanting to stay devious' (German newspaper corpus)

b. weil die Kunden undurchsichtig bleiben wollen
because the customers devious stay.INF *want*
'because the customers want to stay devious'

In a rule-by-rule account, there will normally be two rules – or rule complexes – for VPs and APs. The part of the AP rule complex that takes care of participles will be very similar to the VP rule complex, but there is no guarantee that all extensions (or modifications/corrections) made in the VP rules will take effect on the AP rules.[3]

Now, a principle-based account of the structure-function mapping based on Bresnan's (1997) mixed category analysis will avoid exactly this problem. Working with very general X-bar-type principles on endocentric c-structure configurations[4] and their legal f-annotations, the same restrictions on

[2] See (Bresnan 1998a, 7.2) for the analysis of head mobility in Welsh.

[3] As mentioned in fn. 1, one could use a macro for verb complements as a part of both rules. However, without a theory of such abstractions, it is likely that they will be omitted in many places.

[4] I will not address exocentric rules here.

verb complements are at work, both for the VP and for the AP examples. On the structure-function pairings, further well-formedness conditions apply: these are – apart from the classical Completeness and Coherence condition – "Fully extended" coherence, Endocentricity, Economy of expression.

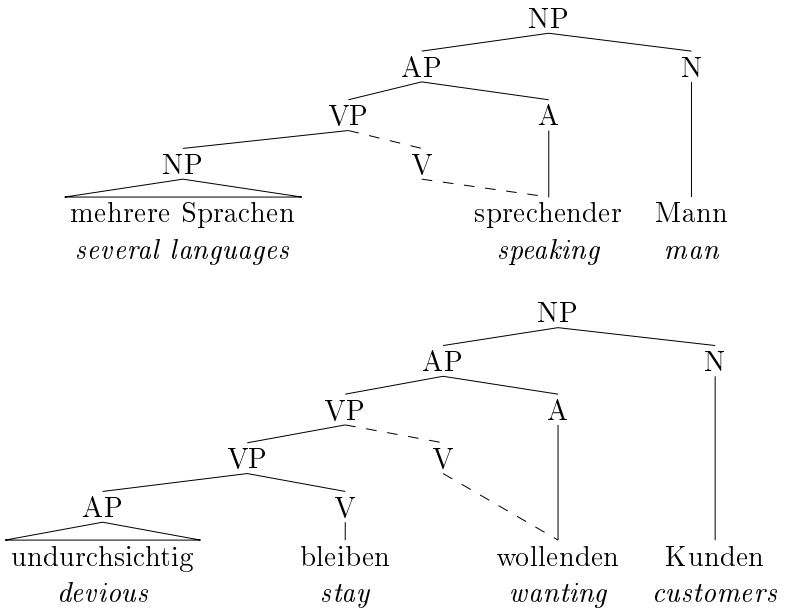


Figure 1: Extended head analysis of adjectival participles

Before addressing the principles in turn, here's the general idea behind the mixed-category analysis of adjectival participles: Their c-structure will look as shown in fig. 1. The dashed lines are not part of the actual c-structure. They indicate the intuition that the participle acts as a head for both the AP and the VP. In the actual analysis, there is no X-bar categorial head of VP (the V node doesn't appear) – it is legal for categories to appear without a head if there is another category (called the *extended head*) that projects to the same f-structure, like the A node does in this case.

With this conception at hand, the principles regulating VP-internal complementation automatically carry over to adjectival participles, since the analysis of these participles involves a VP. At the same time, the participles' external behaviour as adjectives is explained.

2.2 Background: Extended Head Theory

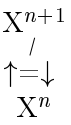
In this subsection, the individual principles making up the Extended Head Theory are addressed in turn. C-structure is organized according to an X-bar theory, where all categories involved appear optional (4) (and in any order, as suggested by the comma).

(4) **X′ theory**

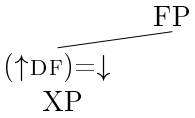
XP	→	(X'),	(YP)	*Head, Specifier*
X'	→	(X^0),	(YP)	*Head, Complement*
X^n	→	(X^n),	(YP)	*Adjunction*

In addition to the bar-level distinction, the categories are classified in terms of *kinds* of categories (V, P, N, or A), and along the lexical/functional distinction (with I and C being functional categories of the verbal kind, and D the nominal functional category). The grammatical functions are classified into grammaticalized discourse functions (TOPIC, FOCUS, and SUBJ), argument functions (SUBJ, OBJ etc.), and non-argument functions (TOPIC, FOCUS, and ADJUNCT).

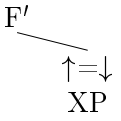
a. C-structure heads are f-structure heads.



b. Specifiers of functional categories are the grammaticalized discourse functions.

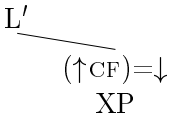


c. Complements of functional categories are f-structure co-heads.



(*To cover mixed categories, this is extended to complements of lexical categories, as an alternative to clause d.: Bresnan 1998a, 7.4, cf. discussion below.*)

d. Complements of lexical categories are the non-discourse argument functions.



e. Constituents adjoined to phrasal constituents are non-argument functions.

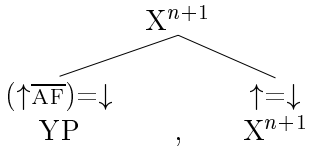


Figure 2: Annotation principles

Based on these concepts, Bresnan (1998a, 131) defines the annotation principles regulating the mapping from c-structure to f-structure in fig. 2. Clause c. (in combination with clause a.) gives rise to configurations where both the c-structure head and its complement are projected to the same f-structure – they are *co-heads*. This makes up an important part of the theory of functional categories: as these categories go together with maximal projections of their lexical counterparts (e.g., a determiner and its NP), the functional category will only contribute certain features, it will not introduce a new level of f-structure embedding. The functional projection and the lexical projection together form an *Extended Projection* (Grimshaw 1991). Since all categories in the X-bar schemata are optional, the lexical projection need not necessarily contain its own c-structure head – provided the information required to make the f-structure complete comes from another source, i.e., the functional head category in this case (an example would be the occurrence of finite verbs in the I position in "verb movement" languages).

Apart from the classical Completeness and Coherence conditions, further well-formedness principles apply, restricting the space of possibilities generated by the X-bar and annotation principles. The *Endocentricity* principle (6) transfers the original intuition of endocentricity in a classical X-bar theory – that every category must contain an X^0 head – to the setting given in a theory that works with extended projections and co-projection of an f-structure from both functional heads and their

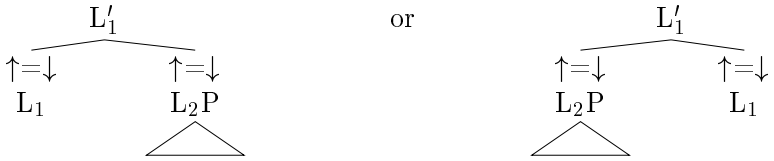
lexical complements. The notion of an *extended head* (5) accomodates for the situation where a category has no X-bar categorial head, but there is a co-projecting category (typically the head of a functional projection) higher up in the tree, in a c-commanding position. The ordinary categorial head is also subsumed under the definition of *extended head*, such that both options are available to satisfy Endocentricity.

(5) *Extended head:* (Bresnan 1999, ch. 7, p. 162)
Given a c-structure containing nodes $\mathcal{N}, \mathcal{C}$, and c- to f-structure correspondence mapping ϕ, $\mathcal{N}$ is an **extended head** of $\mathcal{C}$ if $\mathcal{N}$ is the minimal node in $\phi^{-1}(\phi(C))$ that c-commands $\mathcal{C}$ without dominating $\mathcal{C}$.

(6) **Endocentricity:** (Bresnan 1998a, 182)
Every lexical category has an extended head.

In her analysis of mixed categories, Bresnan (1997) extends the co-head conception (clause c. in fig. 2) – originally devised only for complements of functional heads – to lexical categories:

(7)



(Note that if lexical X^0 categories always introduce a PRED value, a clash will occur if both the L_1 and the L_2 head are present. Furthermore, Endocentricity enforces that the higher head is there: else L'_1 would fail to have an extended head – unless there are further co-projecting categories higher up than L_1P of course.)

This opens the possibility that the VP in the analysis of adjectival participles (fig. 1 and (10) below) acts as an f-structure cohead of A. Note that the A also qualifies as an extended head of the VP (which doesn't contain its own X-bar categorial head V′), thus Endocentricity is satisfied.

The possibility of forming co-heads of lexical categories is certainly highly restricted: in the case of participles it is due to the fact that the adjectival participle is derived from a verb – and not even all morphological derivations will allow for this transparency (cf. Bresnan 1998a, 194 on agentive nominalizations in Kikuyu vs. English). Bresnan (1998a, 184ff) proposes to assume a typing of the PRED value that reflects which types of complements – and features in general – a given item can combine with:

(8) *Typing of semantic forms:*

a. *house* V (↑PRED)='house⟨(↑SUBJ)⟩$_v$'
b. *house* N (↑PRED)='house⟨⟩$_n$'
c. *sprechender* A (↑PRED)='speak⟨⟨x, (↑OBJ)⟩$_v$⟩$_a$'

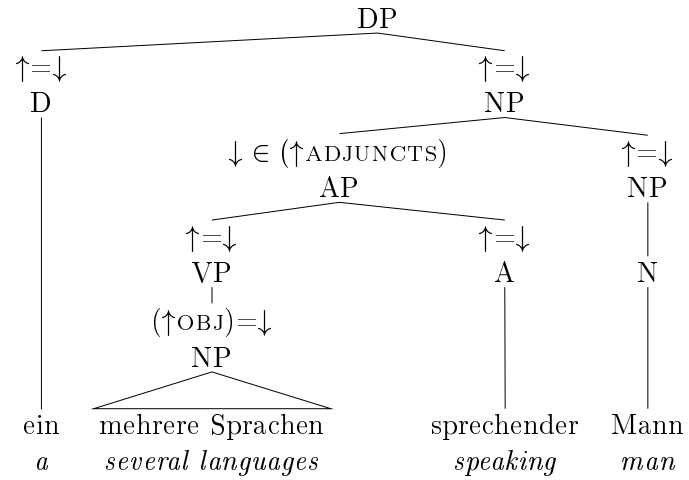
Transparent derived items, like German adjectival participles (8c), will have more than one type in their PRED value (here v and a), thus they are compatible with features arising in both categories' lexical projections – which makes it possible that the lexical co-head configuration can arise.

The following condition excludes f-structures in which the typing is not obeyed:

(9) **Fully extended coherence condition:**
(Bresnan 1998a, 185)
All f-structure attributes must be integrated into their f-structure. [...] Features (symbol-valued attributes) are integrated into their f-structure f when (f PRED) exists and matches the feature in type.

Having introduced X-bar and annotation principles, Endocentricity, and the Fully extended coherence condition, we can see how the analysis for (3a) arises (note that both the adjectival features – CASE, DECL-CL etc. – and the verbal feature OBJ in the f-structure of *speak* are licensed due to the twofold typing of the PRED value):

(10)



$$\begin{bmatrix} \text{PRED} & \text{'man'} \\ \text{SPEC} & \text{INDEF} \\ \text{ADJUNCTS} & \left\{ \begin{bmatrix} \text{PRED} & \text{'speak}\langle\langle x,(\uparrow\text{OBJ})\rangle_v\rangle_a\text{'} \\ \text{CASE} & \text{NOM} \\ \text{NUM} & \text{SG} \\ \text{GEN} & \text{MASC} \\ \text{DECL-CL} & \text{STRONG} \\ \text{OBJ} & \begin{bmatrix} \text{PRED} & \text{'language'} \\ \text{ADJUNCTS} & \left\{ \begin{bmatrix} \text{PRED} & \text{'several'} \end{bmatrix} \right\} \end{bmatrix} \end{bmatrix} \right\} \end{bmatrix}$$

There is one more principle that is very important in restricting the space of possibilities generated by the other principles:

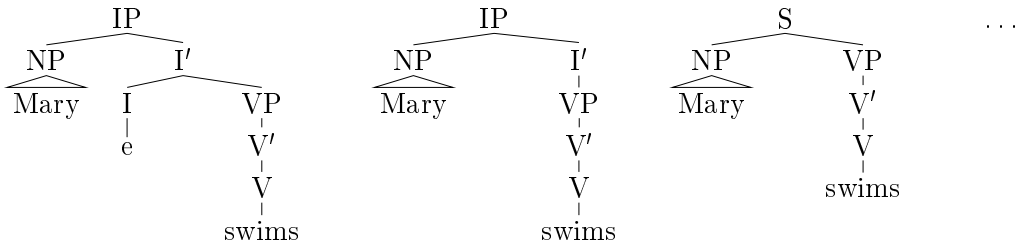
(11) **Economy of expression:** (Bresnan 1998a, 115)
All syntactic phrase structure nodes are optional and are not used unless required by independent principles (completeness, coherence, semantic expressivity).

The assumption of this principle is quite crucial for keeping the overall system of restrictions compact and highly general. The other principles can generate a great number of possibilities (e.g., the structures in (12)[5]), but Economy of expression will pick the smallest of these structures. Without

[5]The third structure involves the non-endocentric category S.

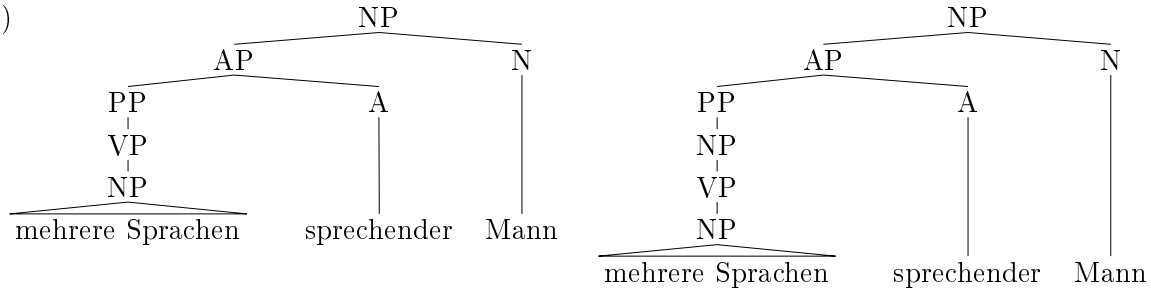
such a principle, every part the generative mechanism (i.e., the rules) would have to be restricted individually to exclude unwanted structures. This would involve a large number of rather complicated case distinctions, while the formulation of Economy of expression simply makes all nodes optional.

(12)



Economy of expression is also involved in the mixed category analysis, excluding larger structures that are otherwise compatible with the mentioned principles:

(13)



The highest explanatory impact of Economy of expression arises however if there are realization alternatives for a given underlying form that will result in different surface forms. A relevant phenomenon is e.g. the realization of pronominal objects in a language that allows object clitics, like French. The structure-function mapping principles allow the realization of such a pronominal object either as a full NP (14a) or as a clitic (14b).

(14) a. *J' ai vu elle
I have seen her

b. Je l' ai vu
*I her.*CLITIC *have seen*

Here, Economy of expression will favour the clitic variant since it involves less structure (assuming that semantic expressivity doesn't make a difference, i.e., there is no special emphasis involved which can only be realized in the full NP variant).[6]

[6]See (Bresnan 1998a, 210) on how clitic doubling fits into the picture – one might expect that Economy of expression should exclude any kind of doubling. The explanation is basically that languages that allow clitic doubling reanalyze the clitic as part of the verbal host.

3 Extended Head Theory and Off-line parsability

Before discussing how the principles of the theory just reviewed can be implemented in a grammar development environment, a point about a formal property of the structures postulated by the theory should be made. Some of the structures that have to be assumed seem to violate a formal restriction imposed on Lexical-Functional grammars: Off-line parsability. It turns out however that this is only true for a particular formulation of this condition, as it is found in published literature.[7]

The Off-line parsability condition is a restriction on allowable c-structures excluding that for a given string, infinitely many c-structure analyses are possible. This would make the parsing problem undecidable. However, only a finite number of c-structures are linguistically plausible, so it makes sense to restrict the formalism to cover only these, thus ensuring decidability of the parsing problem.

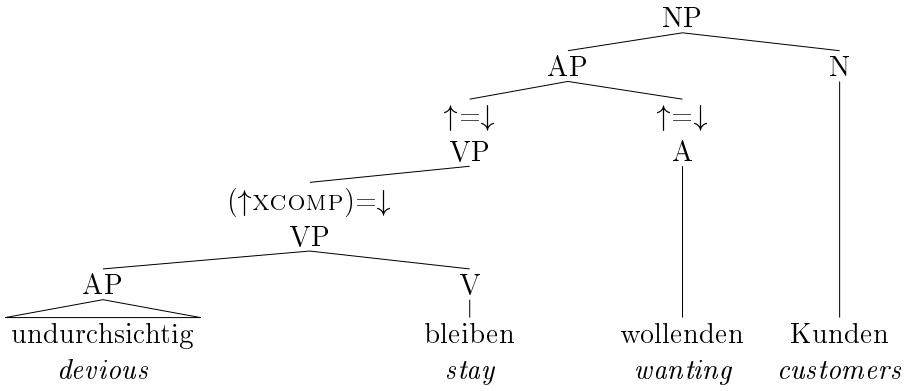
The standard formulation of Off-line parsability (15) excludes non-branching c-structure dominance chains with more than one occurrence of the same category (cf. the example in (16)) – this chain could be passed over and over again, without any change in coverage of overt material:

(15) **Valid Derivation** (*known as Off-line parsability, JK*) (Kaplan and Bresnan 1982, 266)
A c-structure derivation is valid if and only if no category appears twice in a nonbranching dominance chain [...]

(16) * X
|
Y
|
X

Now, the mixed category analysis predicts c-structure configurations that do contain nonbranching dominance chains with two instances of the same category: in (17), the analysis of (3a), the participle is a modal verb. According to the co-head analysis the adjective will take a VP as its complement, which doesn't contain an X-bar categorial head (the A is the VP's extended head). Being a modal verb, the participle now furthermore takes a VP as its complement. So, we have a nonbranching dominance chain with two VPs.

(17)

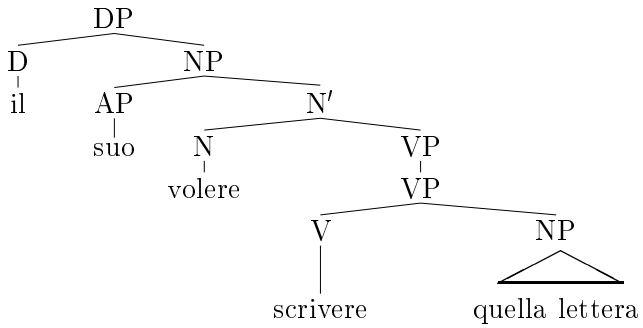


A similar example can be constructed for nominalized infinitives in Italian (see (19) – a modification of example (18)). Analysis (20) furthermore shows that the same apparent problem arises for a head mobility analysis based on Extended Head Theory (Bresnan 1998a, sec. 7.1).

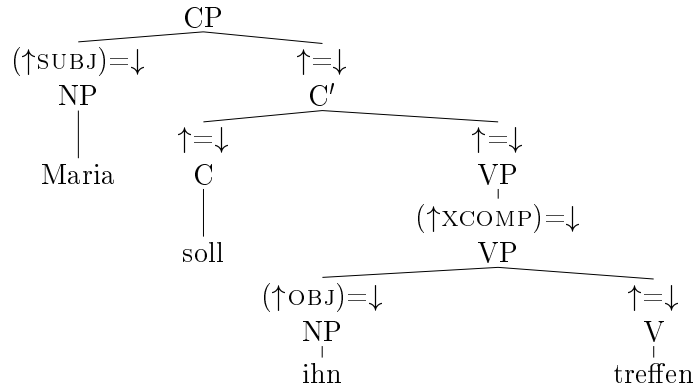
[7] According to a reviewer, Lori Levin pointed out already in the early 80's that the formulation is too strict. In implemented systems, another formulation is used.

(18) il suo scrivere quella lettera improvvisamente
*the his write.*INF *that letter suddenly*
'his suddenly writing that letter' (Bresnan 1998a, 186)

(19) il suo volere scrivere quella lettera
*the his want.*INF *write.*INF *that letter*



(20) Maria soll ihn treffen
M. shall him meet



Does this mean that Extended Head Theory falls outside the class of grammatical systems for which Off-line parsability condition ensures decidability of the parsing problem? Fortunately the answer is no. Under a slightly different formulation of the condition, the intended effect (restricting the set of possible c-structures to be finite) can be reached without excluding the structures just illustrated:

(21) **Revised Off-line Parsability condition**
A c-structure derivation is valid if and only if no category appears twice in a nonbranching dominance chain *with the same f-annotation.*

Note that in the above examples, the re-occurring category has different f-annotations in the two instances, so they do pass the revised condition (21).

Interestingly, the information that has to be taken into account to restrict c-structure in terms of its technical role in parsing is no longer purely categorial, but concerns the structure-function mapping also. Complexity considerations below will also lead to the inclusion of non-categorial information in the process of c-structure (context-free) parsing.

4 Aspects of a computational implementation

In sec. 2, I argued that it would be desirable for language processing efforts to work with a principle-based grammars like the system outlined in sec. 2.2, since by doing so some significant engineering problems of a traditional rule-by-rule account could be avoided. Here, I will discuss how the underlying formalism could accomodate for the formulation of the relevant principles (4.1). In 4.2, I will address a complexity problem caused by this approach, and in 4.3 I will discuss a possible solutions.

Leaving aside the well-formedness conditions (Endocentricity, Fully extended coherence, Economy of expressions) for the moment, we have to find a way of expressing the underlying X-bar and the annotation principles. Note that translating the effect of these principles into ordinary annotated context-free rules is not an option (although if correctly done, the resulting grammar would behave in the desired way) – the goal is to keep the underlying generalizations as explicit as possible to keep the code of a larger grammar manageable over time.

We need a more abstract level to talk about different dimensions of information encoded in one rule. There are basically two ways of going about to facilitate a more explicit statement of generalizations over c-structure rules, which we might call the "representation-based" and the "description-based" approach.

4.1 Representation-based vs. description-based formulation of structure-function mapping principles

In the **representation-based approach**, the representation of the expressions that we are working with is altered. For the purpose of generalizations over c-structural aspects of the rules, this means in particular that c-structure categories are no longer viewed as unanalyzable symbols; rather, category labels are assumed to have some internal structure (cf. also Bresnan 1998a, 125): lexical kind, X-bar level, status of being functional or lexical. Then rules and principles can apply selectively to a certain dimension of information.[8]

The LFG system Xerox Linguistic Environment (XLE) provides a mechanism for using a single rule specification to create a set of effective LFG rules, which will differ in terms of the internal structure of the complex category symbols mentioned in the rule. The notation of a complex category uses an identifier, followed by one or more parameters in square brackets (e.g., `ZP[f,g]`). The original intention was to provide a mechanism for transforming certain feature distinctions into c-structure distinctions (the processing advantages of doing so will be discussed in sec. 4.3, cf. also Maxwell and Kaplan 1995). Thus, a simple example for a rule specification using complex categories (ignoring f-annotations) is the following:

```
PP[_var] --> { P      NP[_var]
             | NP[_var] Ppost   }.
```

The NP and VP categories in this example are viewed as further subclassified, thus their category symbol is complex. Rule specification makes use of the formal parameter `_var`, which will be instantiated to the different actual parameters that can appear in this slot.

[8] One could of course go further and assume generalizations over rules to be regulated by features that undergo the full unification mechanism; however, this would be against the basic conception of LFG's c-structure as a backbone of limited expressiveness, controlling the more powerful f-structure part of the formalism.

Assuming that the grammar works with the PP variants `PP[-wh]` and `PP[+wh]`, the above specification will be expanded to the following effective rules:

```
PP[-wh] --> { P      NP[-wh]
            | NP[-wh] Ppost   }.
PP[+wh] --> { P      NP[+wh]
            | NP[+wh] Ppost   }.
```

The engineering advantage of the explicit generalization is obvious: when the specification is altered, all effective rules will change accordingly.

Now, this mechanism can be used to render the X-bar and annotation principles in a generalized way. The following rules show the basic idea in terms of a rule parametrization along the category kind dimension (in the XLE notation, `^` replaces the ↑, and `!` the ↓; the f-annotations follow the category symbol after a colon and end with a semicolon):

```
XP[_C]    -->  YP: (^DF)=!;   Xbar[_C]: ^=!.
Xbar[_C]  -->  X[_C]: ^=!;    YP: (^GF)=!;*.
```

With further parameters for the lexical/functional distinction (`-f`/`+f`) and bar-level (`0`,`1`,`2`), the following specification will capture the principles discussed in sec. 2.2.[9] (Note that unless ideas of the description-based approach discussed below are integrated, this specification has certain limitations in terms of generality: only one dimension of generalization is fully explicit – some of the basic X-bar principles are still encoded redundantly in the various rule disjuncts[10]).

```
X[_F,_C,_B]  -->
     { e: _F = +f                    " FP --> (GP) (F') "
          _B = 2;
       (X[+f,any,2]: (^DF)=!;)       "specifier"
       (X[_F,_C,1]:  ^=!;)           "head"

     | e: _F = -f                    " L' --> (L) (FP) "
          _B = 1;
       (X[_F,_C,0]:  ^=!;)           "head"
       (X[+f,any,2]: (^GF)=!;)       "complement"

     | e: _B = 1;                    " X' --> (X) (LP) "

       (X[_F,_C,0]:  ^=!;)           "head"
       (X[-f,_C,2]:  ^=!;)           "cohead"

     |                               " Xn --> FP Xn "

        X[+f,any,2]:  !$(^ADJUNCTS); "adjoined category"
       (X[_F,_C,_B]:  ^=!;)          "head"                    }.
```

[9] Constraints on the mother node are encoded using the epsilon expression `e` – this is simply to have a place where to attach the equations. Where `any` is used as a parameter, an expansion to all possible fillings is caused.

[10] However, unless a principle-based grammar is started from scratch, fully general principles will be hard to incorporate into a given large grammar. So a certain limitation of the degree of generality may have practical advantages.

For expressing generalizations in a **description-based approach** it is not necessary to assume changes in the underlying representation (although it is certainly possible). Instead, along the relevant dimensions of generalization, different meta-descriptions are defined. Each such meta-description will describe a certain set of primitive expressions; for instance, a meta-category like `LexCat` could be defined to refer to any lexical category (NP, N′, N^0, VP, V′, etc.), `MaxCat` for all maximal categories (NP, DP, VP, IP etc.). Since these meta-descriptions are regular expressions, they can undergo the operations of concatenation, intersection (`&`), and many other regular language operations. Thus, `[ LexCat & MaxCat ]` will refer to the maximal lexical categories.

Category descriptions can be used to compose phrase descriptions (which again may be used to formulate principles over right-hand sides of rules): the expression
`[ (MaxCat: (^DF)=!;), ([ FuncCat & XbarCat ]) ]`
may, e.g., be used to render clause b. of fig. 2 ("Specifiers of functional categories are the grammaticalized discourse functions"). Several such principles can be intersected in a single LFG rule. This allows one to keep the different dimensions of generalization separate.

The use of regular language descriptions over category strings doesn't mean an extension of the formalism, since the right-hand sides of LFG rules have always been regular expressions.[11]

The description-based approach plays an important role in the considerations about the formulation of comparison-based principles like Economy of expression or OT principles (cf. sec. 5.1 and Kuhn 1999b);[12] however, the representation-based approach is currently more relevant to practical grammar writing. The c-structure level ambiguity problem that I will address in the next section is rather independent of the approach adopted, but I will base considerations on the representation-based approach using complex categories.

4.2 The problem of massive c-structure ambiguity

Having indicated how the X-bar and annotation principles can be encoded, we can come to the additional well-formedness principles. In the theory they are formulated as separate conditions on the structures generated by the underlying rules. If this conception is taken over literally in a computational account, a complexity problem arises. This is due to the fact that while the theory assumes very little restriction at the level of c-structure, c-structure plays a central role in limiting the search space in parsing. Although the Off-line parsability condition guarantees decidability, because the number of c-structure analyses will always be finite (sec. 3), the high generality of the X-bar rules and the wide-spread optionality leads to a combinatorial explosion of alternative c-structure readings that have to be checked. (Even over a single word, a multitude of c-structures are licensed: due to the optionality of the X-bar categorial head in the rules, a maximal category can feed into the head-less variant of the X′ rule as the complement, etc., forming all possible non-branching dominance chains up to the limit that Off-line parsability allows.)

As already noted in (Kaplan and Bresnan 1982, 272), if the number of c-structures ambiguities grows exponentially (with respect to the length of the input string), it takes exponential time to

[11] For very high-level constraints, it does become necessary however to introduce marker symbols (denoting the empty string). This is required, e.g., to ensure that the head of a phrase is unique across different principles referring to it. Such marker symbols have a slightly different status from the symbols appearing in the regular expressions of standard LFG.

[12] See also Butt, Dipper, Frank, and King 1999a, sec. 2.3.2, for an application of the description-based approach: here, a regular expression is used to ensure that a certain type of category is at the periphery of the phrase.

check the f-structures.[13] With the combinatorial explosion occurring with the highly unrestricted c-structure of a general X-bar scheme, parsing of longer sentences would be intractable.

Of course the theory doesn't claim to reflect a processing model in this literal way – and the simplicity and generality of the resulting system is enough justification for the theory. The question that I'd like to address in the remainder of this paper is what the consequence from the viewpoint of language processing should be: do the complexity problems arising in a naive implementation mean that the theory cannot be used in large-scale grammar-writing efforts (despite its obvious engineering advantages thanks to generality)?

I'd like to argue that there are ways of modifiying the naive implementation such that the complexity problem can be avoided without giving up much of the generality; this means that at least analyses inspired by extended-head theory can be integrated in existing larger grammars.

4.3 The reflex of Endocentricity and Fully extended coherence on c-structure

The key idea for avoiding the observed complexity problem is quite simple and not new (cf. Maxwell and Kaplan 1995): additional information is included at the (technical) level of c-structure, such that ill-formed analyses are excluded during c-structure (context-free) parsing already – reducing the search space for the later parsing steps.[14] It is important to note the difference between the view of c-structure as a level of linguistic representation (accomodating categorial information), and the view of it as the backbone in processing, controlling the search space. As discussed for Off-line parsability in sec. 3, the latter view has to be based on more than purely categorial information.

Clearly, it is neither desirable nor possible to transfer all f-structure information into the c-structural part of the rules – else the generative capacity would be that of context-free grammars. It turns out however that Endocentricity (6) (and likewise, a version of the relevant part of Fully extended coherence) can be expressed rather straightforwardly in this way. (The situation with Economy of expression is different: sec. 5.)

Endocentricity demands that every lexical category have an extended head. The definition of the extended head (5) recurs to the ϕ projection and its inverse, but all relevant f-structure information (namely what are c-structure nodes that are projected to the same f-structure) can be read off the f-annotations, i.e., it is available before f-structure construction. Recall that the extended head is either the X-bar categorial head or the closest co-projecting node higher up in the tree.

Having a way of expressing generalizations over rules available anyway (representation-based or description-based – I will show it for the former), we can quite simply restrict the rules in such a way that only structures satisfying Endocentricity are generated: in addition to the previously introduced dimensions of classifying categories we need an additional one to distinguish ones with an X-bar categorial head from ones without. The latter will only be acceptable in a context where there's a head higher up in the extended projection.

[13] In the discussion of the exponential processing complexity of LFG Kaplan and Bresnan (1982, 272) say: "For our formal system, this processing complexity is not the result of a lengthy search along erroneous paths of computation. Rather, it comes about only when the c-structure grammar assigns an exponential number of c-structures ambiguities to a string. To the extent that c-structure is a psychologically real level of representation, it seems plausible that ambiguities at that level will be associated with increased cognitive load."

[14] There may be different ways of reaching a similar effect, involving an adjustment of the parsing algorithm, but the advantage of a grammar-based solution is that the grammar writer can take care of it herself/himself, using a standard parsing system. In practical grammar writing, she/he may occasionally have to do a manual fine-tuning to real-life corpora anyway.

The following rules concentrate on the relevant aspects:

```
YP = XP[+h].

XP[_H]    -->  (YP:          (^DF)=!;)      "XP   --> (YP) (Xbar)"
               { Xbar[_H]:   ^=!                  "specifier head"
               | e:          _H = -h  }.

Xbar[_H]  -->  { { X:        _H = +h        "Xbar --> (X) (YP)"
                             ^=!;                "head complement"
                 | e:        _H = -h; }
                 (YP:        (^GF)=!;)

               | { X:        _H = +h        "Xbar --> (X) (YP)"
                             ^=!;                "head co-head"
                 | e:        _H = -h; }
                 ({ XP[-h]: ^=!
                  | XP[+h]: ^=! })      }.
```

Optionality of the X-bar categorial heads is marked here throughout by an explicit disjunction of the head or epsilon (`e`). When the latter option is chosen, the mother node will be marked as `-h` (for: "lacking an X-bar categorial head"). The distribution of `XP[-h]` categories is very limited: they can only be used as f-structure co-heads in the second disjunct of the `Xbar` rule. In any other usage of an XP category, this has to be one marked `+h`.[15] This ensures that although a `-h` marking can percolate up in the tree, an extended projection that doesn't contain an extended head at some point higher up cannot be used in a well-formed analysis.[16]

With this mechanism the number of c-structure hypotheses can already be reduced considerably. Quite similarly, encoding the effect of the Fully extended coherence condition (9) (which checks whether all features in an f-structure are compatible with the typing of the PRED value) into the rules would have an enormous effect. However, unlike with Endocentricity, there is nothing in the definition of Fully extended coherence that ensures tree-geometrical connectedness: in principle, the feature/PRED-value-type correspondence could be mediated through a non-local dependency. So, there cannot be an exactly equivalent rendering of this condition in terms of principles operating directly on rules, only an approximation. It is not so clear though whether there would be an empirical difference between the approximation and the original condition, since constructed configurations that the approximation would filter out erroneously look quite implausible and will probably violate other principles.

The structure in fig. 3 is such a constructed example. It contains a split-up nominal phrase (meaning *the writing of some letters*), which at the same time is a mixed category (like the Italian example (18)), i.e., takes verbal complements. Note the lexical co-head status of the VP within the lower NP. Both the VP and the NP have D as their extended head.

Split NPs do certainly exist in languages like German (cf. (22), where the noun's complement – *über Linguistik* – is separated from its head – *Fragen*), and a doubling-style analysis making use of (functional-uncertainty-based) unification between the TOPIC and the potentially embedded OBJ has been proposed in (Kuhn 1998; Kuhn 1999a).

[15] In all non-head positions the meta-category `YP` is used, which is defined as `XP[+h]`.

[16] In essence, the `-h` marking functions like the slash marking in GPSG.

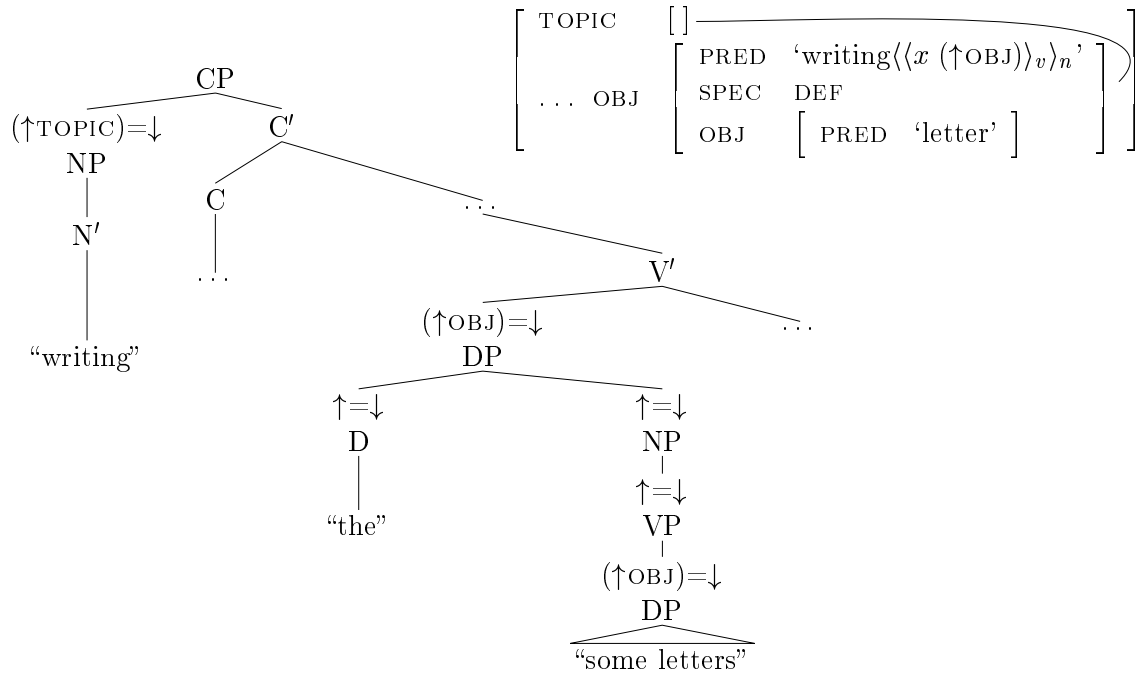


Figure 3: Implausible hypothetical structure in a hypothetical language

(22) Fragen stellte er nur zwei über Linguistik
questions asked he only two about Linguistics
'As for questions, he asked only two about Linguistics.'

Unfortunately, German nominalizations do not display the mixed category behaviour of their Italian counterpart, and there is no NP split in Italian, so for independent reasons we couldn't get a sentence with a structure like the one in fig. 3 in either of the two languages. This means that the example is only hypothetical, and one can't claim it is ungrammatical. Nevertheless it is intuitively weird.

What's implausible about fig. 3 is that the object *some letters* of the nominalized verb is introduced in the lower NP, while the nominalization itself is introduced in the upper NP. Local to the lower NP, there are no overt signals for an environment that makes verbal complements acceptable. Note however, that the Fully extended coherence condition (9) does allow for this sitution: the verbal OBJ feature in the f-structure of *writing* is licensed, since at the level of f-structure information from the other NP (which contains a PRED value typed both v and n) is available through unification – mediated by functional uncertainty equations. With principles operating directly on rules, the unbounded dependency could not be captured – such an account has to be limited to interactions within a single extended projection.

Within an extended projection, the compatibility of features and PRED-value-type demanded by Fully extended coherence can certainly be checked with a similar mechanism as the one for Endocentricity discussed above (at least as far as the interaction with mixed categories is concerned): the morphologically introduced typing of items like the heads in mixed categories could be carried along as another dimension of classification for c-structure categories. This would also cover split NP examples like (22), since here no extra category types are involved – the lower NP just contains typical nominal complements. Note that if examples like the one in fig. 3 are systematically

ill-formed, the formal limitations of this rule-based formulation wouldn't even mean an unwanted limitation in coverage, but would express a generalization.

Since the unrestricted combination of lexical categories as co-heads was another major source of c-structure level ambiguity, this brings us again closer to a tractable system. (For practical grammar writing, these two steps may actually suffice already to allow the inclusion of analyses inspired by Extended Head Theory, since one would restrict the total optionality in the rules anyway and work without the Economy of expression condition.)

5 The Comparison Problem

So far, we haven't looked at one of the wellformed-principles from the computational perspective: Economy of expression (11). One might hope that again a similar construction like the ones proposed in the previous section can be adopted. However, this is not possible in a straightforward way, since checking this condition requires a more powerful system: Economy of expression says that all syntactic phrase structure nodes are only used if required by independent principles (completeness, coherence, semantic expressivity). This condition cannot be checked by looking at a single LFG analysis (a c- and f-structure and a correspondence mapping), it essentially requires a comparison with alternative analyses. In this sense, a grammatical system working with an economy principle is similar to an Optimality Theoretic (OT) grammar, for which the definition of grammaticality makes explicit reference to a set of candidate analyses, from which the most harmonic one is picked (Prince and Smolensky 1993; Grimshaw 1997; Bresnan 1996; Bresnan 1998b).

It is important to be clear about what alternative analyses are potential competitors when the most economical analysis is determined. Here, as the definition says, semantic expressivity (and completeness and coherence) has to be taken into account: if an analysis means something different, it doesn't count as a competitor, even if it is very economical in terms of the number of syntactic nodes used. In effect, we can use the standard definition of a candidate set in OT-LFG (see e.g. Bresnan 1998b): all analyses with the same underlying semantic form compete – this can be captured in LFG terms by looking at all analyses whose f-structure is subsumed by an underspecified "input" f-structure that basically contains the PRED values and some semantically relevant features.

Now, what does a processing system have to do in order to check whether Economy of expression is satisfied? In language production, or generation, the task is comparatively simple: from the underlying form all analyses satisfying the other principles are generated; then from these analyses (which are all competitors in the sense just clarified) the analysis with the least number of phrase structure nodes is picked. The other ones are not well-formed.

In understanding, or parsing, the task is more complicated: it is not sufficient to determine all analyses over the input string satisfying the other principles and then pick the one with the least nodes: there may be another way of expressing the same underlying form that's even more economical, in which case the given string is ungrammatical according to Economy of expression (a similar point is made in Johnson 1998 for OT-LFG). Take the example of a pronominal object in French (14) which – according to the other principles – might be realized as a full NP or as a clitic; the full NP version violates Economy of expression and is thus ungrammatical. Now, when the ungrammatical string (14a) with a full NP pronominal object is parsed, it will certainly receive an analysis. In order to find the more economical competitor, an additional processing step becomes necessary after the parsing task has been finished: from each of the semantic forms arrived at in parsing, all alternative analyses have to be generated in order to check whether there is one that gets by with fewer nodes

than the analysis over the original string (cf. also Kuhn 1999b, 3.4, again for similar considerations in OT-LFG).

This shows that checking Economy of expression adds a complication that is of an entirely different nature than the c-structure level ambiguity problem discussed in sec. 4.2. Even if c-structural ambiguity could be seriously restricted (along the way discussed in the previous section), the additional generation task required for every reading adds considerable complexity. (Note that the readings for which generation is required are readings at the level of f-structure, i.e., here we have to deal with the "real" ambiguities, and we can't hope to find a more restrictive formulation of principles that will avoid this kind of ambiguity.[17])

5.1 Precompiling the comparison

Although a faithful rendering of the effect of Economy of expression in an implementation will require the considerably more complicated processing just discussed, it seems intuitively strange that for parsing every sentence, all possible analyses are effectively generated – including the most hopeless candidates. One somehow gets the feeling that the parser should know better.

For example, one might think that paths in the search space that already involve considerably more complex structures than an alternative path should be given up. Unfortunately, one cannot exclude until the entire analyses have been constructed that such a hopeless looking path will ultimately lead to the most economical candidate: some principle applying quite late may exclude the putatively winning competitor.

However, in (Kuhn 1999b, sec. 4), I discuss a potential way of precompiling the effect of comparison-based constraints like Economy of expression or OT constraints, which relies on the assumption that the linguistically relevant effect of such a comparison can be reached also when its domain is restricted to a locally confined structure – like a single extended projection. Interactions with material outside this domain can still be captured, as long as they can be communicated through an interface with a finite set of possible representations.

With these restrictions, it is possible to precompile the effect of the comparison, i.e., a grammar can be generated that will only accept structures that are maximally economical for their underlying semantic form (or, in the OT setting, optimal). The precompilation is inspired by Karttunen's (1998) computational account of OT phonology. Karttunen shows that for OT systems in which the generative component and all constraints can be formalized by regular expressions, a precompilation of the comparison is possible. The entire OT system is then modelled by a transducer that relates underlying forms with their optimal realization.

This idea can be exploited in the LFG setting for precompiling appropriately restricted rules (adopting the desciption-based approach of generalizations over rules, discussed in sec. 4.1).[18] Interestingly, a precompilation approach thus boils down to c-structure level restrictions on rules very similar to the approach discussed in sec. 4.3 – although the principles being rendered are conceptually much more high-level.

[17] It should be noted however that many of the processing issues involved here are still open, so possibly ways can be found that exploit the work done already in parsing for the "back generation" step in a clever way.

[18] One way to make this work is to assume rules that cover entire extended projections.

6 Conclusion

In this paper I argued that a generalized formulation of the structure-function mapping is highly desirable for large-scale grammars, since it would solve a significant engineering problem. Extended Head Theory constitutes such a set of principles. For expressing the underlying X-bar and annotation principles, there are at least two ways of introducing a level of abstraction over rules, without having to alter the architecture of LFG much.

The formulation of the Off-line Parsability restriction has to be modified slightly to capture the structures assumed by the theory (and thus ensure decidability of the parsing problem for grammars based on the theory). However, a naive implementation will nevertheless run into severe complexity problems in a standard LFG processing system, since at the level of c-structure the system is quite unrestricted.

To solve these problems, the Endocentricity and the Fully extended coherence condition can be reformulated to apply as a restriction on rules, thus avoiding the introduction of a vast number of ambiguities in c-structure parsing. The positive effect on processing can be reached without giving up the generality in the grammar specification. This makes an application of analyses using these or similar principles in grammar writing realistic. For some principles, the rule-level may even be the theoretically adequate level of application.

For checking Economy of expression a more powerful processing mechanism is required, since the principle relies on the comparison of different analyses. Under the assumption of certain restrictions on expressiveness, even the effect of comparison-based principles can be captured by compiling a grammar with appropriately restricted rules.

Acknowledgements

The work reported in this paper has been funded by the *Deutsche Forschungsgemeinschaft* within the *Sonderforschungsbereich* 340, project B12 (Methods for extending, maintaining and optimizing a comprehensive grammar of German). I'd like to thank Judith Berman, Joan Bresnan, Mary Dalrymple, Stefanie Dipper, Christian Fortmann, two reviewers, and the audience at the LFG99 conference in Manchester for comments and discussion. The mechanism of rule parametrization using complex category symbols (sec. 4.1) arose in discussions within the ParGram project (a collaboration between Xerox PARC, the Xerox Research Centre Europe, Grenoble, and the IMS Stuttgart), involving in particular Ron Kaplan and John Maxwell.

References

Bresnan, J. (1996). LFG in an OT setting: Modelling competition and economy. In M. Butt and T. H. King (Eds.), *Proceedings of the First LFG Conference*, CSLI Proceedings Online.

Bresnan, J. (1997). Mixed categories as head sharing constructions. In M. Butt and T. King (Eds.), *Proceedings of the LFG97 Conference*. CSLI Publications Online.

Bresnan, J. (1998a). Lexical-Functional Syntax. Book manuscript, ch. 6-8 in reader for ESSLLI'98 course: Bresnan/Sadler, Modelling Dynamic Interactions between Morphology and Syntax. Saarbrücken. Page numbers follow ms. version dated Summer 1998.

Bresnan, J. (1998b). Optimal syntax. In J. Dekkers, F. van der Leeuw, and J. van de Weijer (Eds.), *Optimality Theory: Phonology, Syntax, and Acquisition*. Oxford University Press. To appear.

Bresnan, J. (1999). Lexical-Functional Syntax. Manuscript version of Spring 1999.

Butt, M., S. Dipper, A. Frank, and T. King (1999a). Writing large-scale parallel grammars for english, french, and german. In M. Butt and T. H. King (Eds.), *Proceedings of the LFG99 Conference, Manchester, UK*, CSLI Proceedings Online.

Butt, M., T. King, M.-E. Niño, and F. Segond (1999b). *A Grammar Writer's Cookbook*. Number 95 in CSLI Lecture Notes. Stanford, CA: CSLI Publications.

Grimshaw, J. (1991). Extended projection. Unpublished Manuscript, Brandeis University.

Grimshaw, J. (1997). Projection, heads, and optimality. *Linguistic Inquiry 28*(3), 373–422.

Johnson, M. (1998). Optimality-theoretic lexical functional grammar. In *Proceedings of the 11th Annual CUNY Conference on Human Sentence Processing, Rutgers University*. To appear.

Kaplan, R. and J. Bresnan (1982). Lexical-functional grammar: a formal system for grammatical representation. In J. Bresnan (Ed.), *The Mental Representation of Grammatical Relations*, Chapter 4, pp. 173–281. Cambridge, MA: MIT Press.

Kaplan, R. M. and J. T. Maxwell (1996). LFG grammar writer's workbench. Technical report, Xerox PARC. `ftp://ftp.parc.xerox.com/pub/lfg/`.

Karttunen, L. (1998). The proper treatment of optimality in computational phonology. In *Proceedings of the International Workshop on Finite-State Methods in Natural Language Processing, FSMNLP'98*, pp. 1–12. To appear, ROA-258-0498.

Kroeger, P. (1993). *Phrase Structure and Grammatical Relations in Tagalog*. Stanford: CSLI Publications.

Kuhn, J. (1998). Resource sensitivity in the syntax-semantics interface and the german split np construction. In T. Kiss and D. Meurers (Eds.), *Proceedings of the ESSLLI X workshop "Current topics in constraint-based theories of Germanic syntax"*, Saarbrücken. Revised version to appear in a CSLI volume.

Kuhn, J. (1999a). The syntax and semantics of split NPs in LFG. In F. Corblin, C. Dobrovie-Sorin, and J.-M. Marandin (Eds.), *Empirical Issues in Formal Syntax and Semantics 2, Selected Papers from the Colloque de Syntaxe et Sémantique à Paris (CSSP 1997)*, The Hague, pp. 145–166. Thesus.

Kuhn, J. (1999b). Two ways of formalizing OT syntax in the LFG framework. Ms. Institut für maschinelle Sprachverarbeitung, Universität Stuttgart, 32 pp., revised version to appear in a CSLI volume edited by Peter Sells.

Maxwell, J. and R. Kaplan (1995). The interface between phrasal and functional constraints. In M. Dalrymple, R. Kaplan, J. Maxwell, and A. Zaenen (Eds.), *Formal Issues in Lexical-Functional Grammar*. Stanford, CA: CSLI Publications.

Prince, A. and P. Smolensky (1993). Optimality theory: Constraint interaction in generative grammar. Technical Report Technical Report 2, Rutgers University Center for Cognitive Science.

Jonas Kuhn
`jonas@ims.uni-stuttgart.de`

Institut für maschinelle Sprachverarbeitung
Universität Stuttgart
Azenbergstr. 12
D-70174 Stuttgart GERMANY