# Getting Things Out Of Order

**(An LFG-Proposal for the Treatment of German Word Order)**

*Klaus Netter*
Department of Linguistics
University of Stuttgart
West Germany

## 0. INTRODUCTION 1)

One of the most characteristic features of German word order seems to be a contrast between fixed ordering rules concerning the order of verbal elements and a much more variable ordering of their corresponding nominal arguments. As a consequence German word order seems to yield a large number of phenomena that may be classified as 'unbounded' or 'long-distance dependencies', without necessarily involving wh-constituents or 'movement' across sentence boundaries. Whereas in traditional LFG long distance dependencies are treated by means of constituent control, we will follow a recent proposal by KAPLAN and ZAENEN (1986) to give up the constraint known as 'functional locality' and instead allow regular expressions to appear as functional schemata annotated to c-structure rules. Exploiting the principles of completeness and coherence we will thus be able to cope even with absolutely free word order without the need of generating empty terminal nodes at all. The empirical assumption, underlying the proposed analysis in its most radical form, is the hypothesis that (with very few exceptions) the nominal arguments have to appear to the left of the verb by which they are assigned case. We will restrict the discussion to sentences with one finite verb as well as to subcategorized nominal arguments, largely ignoring ADJuncts. 2)

In chap. 1. we will introduce the mechanism of 'functional incertainty', applied to the example of extraposed infinitival constructions. Chap. 2. will deal with the sentence initial position(s) in verb-second clauses and chap. 3. will discuss some approaches to handle free word order in the topological field between COMP and the clause-final verb complex.

## 1. EXTRAPOSED INFINITIVAL CONSTRUCTIONS

To demonstrate the mechanism of 'functional incertainty' as well as some basic assumption in the proposed fragment of German grammar we will start with a relatively simple problem involving the treatment of embedded and extraposed infinitival constructions. Consider the following data: 3)

(1) *dass [der Mann [[ das Buch gelesen ] haben] wird].*
that the man the book read have will
(2) *dass [der Mann [[das Buch zu lesen] versucht] hat].*
that the man the book to read tried has
(3) *dass [der Mann [[das Buch gelesen] zu haben] scheint].*
that the man the book read to have seems
(4) *dass [der Mann [versucht] hat], [das Buch zu lesen].*
that the man tried has , the book to read
(5) **dass[der Mann scheint],[[ das Buch gelesen] zu haben].*
that the man seems , the book read to have
(6) **dass [der Mann wird], [[das Buch gelesen] haben].*
that the man will, the book read have
(7) *dass [der Mann [ versucht] hat], [zu behaupten], [[das Buch gelesen] zu haben]*
that the man tried has, to claim, the book read to have

Sentence (1) contains a main verb and two auxiliary verbs, (2), (4) and (6) contain equi verbs (versuchen, behaupten), 'scheinen' in (3) and (5) represents a raising verb. (2) is the same structure as (4) with the infinitival clause extraposed in the latter. The contrast between (2)/(4) and (3)/(5) or (1)/(6) illustrates two points: on the one hand extraposition is permitted iff the governing verb governs the 'zu' infinitive and is not a raising verb; on the other hand even with these conditions met extraposition is not obligatory. Informally extraposition can thus be characterised as an optional right-adjunction of an embedded constituent to a self-embedding structure. Sentence (7) is to show that this rule can be applied recursively.

In the treatment of equi- and raising-verbs, including (in our approach) auxiliary and any kind of modal verbs, we will follow the traditional LFG approach; these verbs are subcategorized for the 'open' grammatical function XCOMP and are correspondingly assigned a functional control equation that identifies the verb's own subject or object with the subject of this XCOMP. Raising verbs are distinguished from equi verbs by denoting that the 'raised' function will not be interpreted in the raising verb's semantic form. Beyond this it will have to be specified for each equi or raising verb what type of infinitive the verbal head of the governed XCOMP has to be (The feature referring to extraposition in (L3) will be attached only to the appropriate verbs). 4)

```
(L1) wird: V,     (↑ PRED) = "werden <(XCOMP)> (SUBJ)"
                  (↑ TENSE) = future
                  (↑ SUBJ CASE) = nom
                  (↑ SUBJ) = (↑ XCOMP SUBJ)
                  (↑ XCOMP INF) =c en

(L2) haben: V,    (↑ PRED) = "haben <(XCOMP)> (SUBJ)"
                  (↑ INF) = en
                  (↑ SUBJ) = (↑ XCOMP SUBJ)
                  (↑ XCOMP INF) =c ge

(L3) versucht: V, (↑ PRED) = "versuchen <(SUBJ) (XCOMP)>"
                  (↑ INF) = ge
                  (↑ SUBJ) = (↑ XCOMP SUBJ)
                  (↑ XCOMP INF) =c zu
                  (↑ XCOMP EXTRAPOSITION) = +.
```

Given a lexicon of the above format we could use the following rules to generate the embedded infinitival constructions (1)-(3):

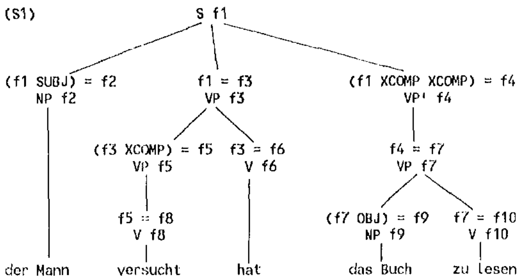
```
(R1)   S'    ->   COMP   ↑ = ↓
                         !(↑ TENSE)
                  S      ↑ = ↓.

(R2)   S     ->   NP     (↑ SUBJ) = ↓
                  VP     ↑ = ↓.

(R3)   VP    ->   [NP    (↑ OBJ2) = ↓]
                  [NP    (↑ OBJ) = ↓]
                  [VP    (↑ XCOMP) = ↓]
                  V      ↑ = ↓.
```

To cover the extraposition cases we can make efficient use of the notion of 'functional incertainty'. What we want to express is that the extraposed infinitival construction appears as the right-most daughter of the governing S-node, that it fulfills the function of an XCOMP and that it should be embedded in the resulting f-structure at an unpredictably deep level which is determined by the number of XCOMPs that have been generated by the self-embedding VP in (R3). To do this we will expand the rule (R2) by introducing a special VP' category that is annotated by a functional schema containing a regular expression involving the Kleene star on the grammatical function XCOMP (The constraining equation attached to the VP' nodes serves to make sure that only those XCOMPs will be dominated by this node that are governed by an appropriate verb satisfying this constraint through the corresponding defining equation in its lexical entry; see above (L3).):

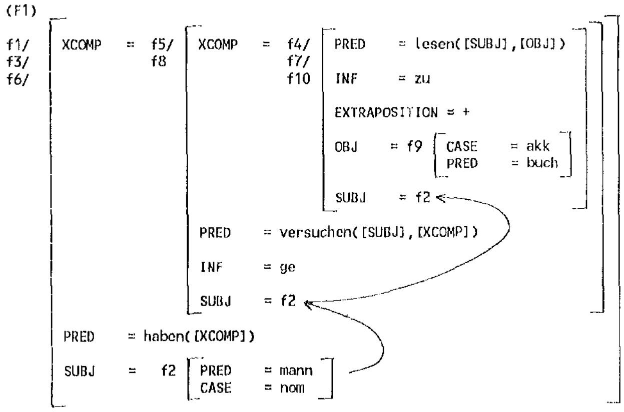
```
(R2.1) S     ->   NP     (↑ SUBJ) = ↓
                  VP     ↑ = ↓
                  [VP'   (↑ XCOMP* XCOMP) = ↓
                         (↓ EXTRAPOSITION) =c +].

(R4)   VP'   ->   VP     ↑ = ↓
                  [VP'   (↑ XCOMP* XCOMP) = ↓
                         (↓ EXTRAPOSITION) =c +].
```

These rules will produce the following c- and f-structures with the metavariables appropriately instantiated. Although the VP'-node appears in the c-structure as an immediate daughter of S it's corresponding f-structure (f4) is pushed down to the deepest level of embedding in the global f-structure.

(S1)




(F1)




That the number of iterations implied by the Kleene-star operator on the XCOMP in (R2.1) and (R4) will always be the correct one is guaranteed by principles that are central to traditional LFG: The principle of coherence requiring that any governable function must be governed by a local predicate in its f-structure will make sure that the number of iterations will not exceed the number of XCOMPs generated by the left sister VP structure plus one; the requirement for a minimal solution and the restriction that any predicate introduced by a lexical entry must not be unified with any other predicate entry will make sure that the number of iterations will not be smaller than the appropriate number of XCOMPs.

## 2. VERB-SECOND

Until now we have considered only clauses with the finite verb in a final position, and a strict order of nominal arguments. In the following we shall modify our rules to cover also verb second clauses, restricting ourselves to declaratives. 5)

The rule determining the position of the finite verb in German can be informally expressed by the following generalization: The finite verb appears in a clause final position iff the first position of the sentence is taken by a conjunction, or by a relative or (subordinative) interrogative pronoun. If these conditions do not apply we will assume that the finite verb takes the position that is dominated by COMP in (R1). In this case the position in front of the finite verb, which we will assume to be dominated by S" , may be filled by any major constituent (including VP). As this position is by no means restricted to be a SUBJ-position, we will introduce a variable G ranging over the full set of governable functions (including XCOMP). 6)

Furthermore we have to take into consideration, that only the surface realization SUBJ will always appear on the top-level of the global f-structure, whereas any other function may belong to a deeper level of f-structure embedding. So we will again make use of an annotation involving the Kleene-star operator, yielding (↑ XCOMP* G) = ↓. Thus we will be able to topicalize any governable function no matter, how deep it be embedded in the VP or VP' structures. The principle of consistency will again guarantee that no governable function will appear in this position that is not subcategorized for by a verbal predicate that can be reached along an f-structure path consisting of XCOMPs only. Restricting the f-structure path to XCOMPs will make sure that no governable function be 'extracted' from inside a complex NP or some node annotated as ADJUNCT. 7)

Another important modification concerns our S-rule: In order to avoid constituent control as well as a VP rule expanding to an optional V constituent, we 'flatten' the S rule, making all its immediately dominated constituents optional. The principle of consistency will then make sure that one and only one position for the verb will be selected. The existential constraint requiring tense will hold for both the final and the front position.

| | | | | |
|---|---|---|---|---|
| (R0) | S'' | -> | [XP | (↑ XCOMP* G) = ↓ |
| | | | | (↑ MOOD) = declarative] |
| | | | S' | ↑ = ↓. |
| (R1) | S' | -> | COMP | ↑ = ↓ |
| | | | | !(↑ TENSE) |
| | | | S | ↑ = ↓. |
| (R2.2) | S | -> | [NP | (↑ SUBJ) = ↓ |
| | | | [NP | (↑ OBJ2) = ↓] |
| | | | [NP | (↑ OBJ) = ↓] |
| | | | [VP | (↑ XCOMP) = ↓] |
| | | | [V | ↑ = ↓] |
| | | | [VP' | (↑ XCOMP* XCOMP) = ↓ |
| | | | | ...]. |
| (R5) | COMP | -> | V | ↑ = ↓ |
| (R6) | COMP | -> | CONJ | ↑ = ↓ |
| | | | | (↑ MOOD) = subordinate. |

Given this rule system it will be no problem to generate any of the following declarative sentences:

(8) *Der Mann hat [der Frau das Buch gegeben].*
the man has the woman the book given
(9) *Der Frau hat der Mann das Buch gegeben.*
to the woman has the man the book given
(10) *Das Buch hat der Mann der Frau gegeben.*
the book has the man to the woman given
(11) *Der Frau hat er versprochen, das Buch zu lesen.*
to the woman has he promised the book to read
(12) *Das Buch hat er ihr versprochen zu lesen*
the book has he her promised to read
(13) *Das Buch hat er ihr zu lesen versprochen.*
the book has he her to read promised
(14) *Das Buch zu lesen hat er ihr versprochen.*
the book to read has he her promised
(15) *Versucht hat er, das Buch zu lesen.*
tried has he the book to read

## 3. GERMAN AS A FREE WORD ORDER LANGUAGE

In the last section we assumed that it will be only the sentence initial positions that can be filled by any major constituents. However there is little reason to assume that the linear order of nominal arguments will always follow the pattern produced by the original rules (R2) and (R3):

(16) *dass dem Erfinder diese Entdeckung gelungen ist.*
to the inventor this discovery succeeded has
(17) *dass dem Mann jemand ein Buch geschenkt hat.*
to the man somebody a book given has
(18) *dass ihm das Buch jemand geschenkt hat.*
him the book somebody given has
(19) *dass es ihm jemand geschenkt hat.*
it him somebody given has
(20) *dass sich der Mann zu bewegen begann.*
himself the man to move started
(21) *dass ihn eine Studentin zu kuessen versuchte.*
him a student (female) to kiss tried
(22) *dass eine Studentin ihn versucht hat zu kuessen.*
a student him tried has to kiss
(23) *dass ihn eine Studentin versucht hat zu kuessen.*
him a student tried has to kiss
(24) *dass es ihm jemand zu lesen versprochen hat.*
it to him somebody to read promised has
(25) *dass sie ihm das Buch [ hat [[geben] wollen]].*
she him the book has given wanted

(16) is representative for a class of verbs which require an unmarked linear order OBJ2 < SUBJ. (17)-(19) show some permutations of SUBJ, OBJ and OBJ2 which seem to obey some linear precedence rules relating to a definiteness-scale for noun phrases. In (20) and (21) an argument depending on a verb governed by an equi verb precedes the subject of this equi verb. In the extraposition variants (22) and (23) of the embedded structure (21) the verbs are extraposed while leaving behind their arguments. In (24) the linear order of nominal arguments has been 'reversed' to yield a structure comparable to some Dutch infinitival constructions. In (25) the finite verb separates the all objects from heads appearing in a left-branching verb complex. All of these sentences have in common that the arguments of some embedded verb(s) are separated from their head(s) by some constituent belonging to a higher level of f-structure embedding, i.e. they are obviously not derivable by a simple self-embedding c-structure rule. Basically three different approaches to handle these cases come to mind:

a) One could stick to rule (R2.2) and modify it by introducing several NP-positions as left sisters of the SUBJ-NP annotated with a regular expression containing as its rightmost attribute a variable Go ranging over the set of various OBJects:

```
(R2.3)  S     ->    NP*    (↑ XCOMP* Go) = ↓
                    [NP    (↑ SUBJ) = ↓]
                    [NP    (↑ OBJ2) = ↓]
                    [NP    (↑ OBJ) = ↓]
                    [VP    (↑ XCOMP) = ↓]
                    [V     ↑ = ↓].
```

Apart from potentially yielding several different c-structure analyses in cases where the SUBJ has been topicalized, this rule would only cover the sentences (16)-(21).

b) In addition to the modifications under a) the V constituent in the VP-rule is made optional and complemented by introducing a V' rule: 9)

```
(R3.1)  VP    ->    [NP    (↑ OBJ2) = ↓]
                    [NP    (↑ OBJ) = ↓]
                    [VP    (↑ XCOMP) = ↓]
                    [V'    ↑ = ↓].

(R7)    V'    ->    [V'    (↑ XCOMP) = ↓]
                    V      ↑ = ↓
                    [V'    (↑ XCOMP) = ↓
                           (↓ INF =c en].
```

A rule system of this format (if annotated appropriately) seems to handle the double infinitive cases as well as some of the extraposition structures quite nicely, however it has several major disadvantages: As the clause final verb-complex can be generated by the VP or the V' rule (or both), in many cases this could lead to multiple c-structure ambiguities for identical input strings with identical f-structure outputs. To make sure that the verbs appear in a 'coherent' structure obeying the constraints on possible topicalization and extraposition, quite a number of additional constraining rules presumably will be necessary. Finally, as in this rule system c-structure derivations may not be avoidable, that follow a rule like

```
VP    ->    VP    (↑ XCOMP) = ↓.
```

(i.e. violate the constraint for valid c-structure derivations Kaplan/Bresnan (1982: 266f)), a slight increase in the generative power of the system may be the consequence.

c) The VP is given up altogether in non-peripheral positions. We define a variable ranging over the set of grammatical functions that may be fulfilled by NPs (let's call it Gs) and modify our rules in the following way:

```
Range of Variables:

G   =:   { SUBJ, OBJ, OBJ2, XCOMP }
Gs  =:   { SUBJ, OBJ, OBJ2 }
Go  =:   { OBJ, OBJ2 }
```

```
(R0), (R1), (R5), (R6)

(R2.4)  S     ->    [NP*   (↑ XCOMP* Gs) = ↓]
                    [V     ↑ = ↓
                           (↑ XCOMP INF =c en]
                    [V'    (↑ XCOMP) = ↓]
                    [V     ↑ = ↓]
                    [VP    (↑ XCOMP* XCOMP) = ↓
                           (↓ EXTRAPOSITION) =c +].

(R3.2)  VP    ->    [NP*   (↑ XCOMP* Go) = ↓]
                    [V'    (↑ XCOMP) = ↓]
                    V      ↑ = ↓
                    [VP    (↑ XCOMP* XCOMP) = ↓
                           (↓ EXTRAPOSITION) =c +].

(R7.1)  V'    ->    [V'    (↑XCOMP) = ↓]
                    V      ↑ = ↓.
```

## 4. CONCLUSION

As we mentioned in the introduction, the analysis proposed in 4.c) is a very radical solution which in its present form presupposes that German is a (relatively) free word order language. Its advantages should be obvious from the examples given above: Free word order in the domain of nominal arguments, extraposition and topicalization of incomplete VP-nodes etc. can be easily covered, its disadvantages might be limitable by the introduction of linear precedence rules. Still, the idea of functional incertainty certainly represents an interesting mechanism, that could make it much easier to get things out of order.

Footnotes:

1) This paper has been based on research presently carried out at the Department of Linguistics, Stuttgart, sponsored by the BUNDESMINISTER FUER FORSCHUNG UND TECHNOLOGIE (BMFT) (grant no. 1013207 0).
2) For lack of space we will have to presuppose the theory of LFG as it was developed in Kaplan / Bresnan (1982) and Bresnan (1982a). The paper on 'functional incertainty' by KAPLAN / ZAENEN to our knowledge has not been published yet; our use of this mechanism is for the moment restricted insofar as we will allow only iterations on XCOMPs and that we do not permit regular expressions to appear in functional control equations.
3) The infinitival constructions in German are described in great detail in BECH (1955), which has been claimed to be an early version of Bresnan's theory of control.
4) The syntax of our grammar rules follows the syntax accepted by the Stuttgart LFG-system (see EISELE / DOERRE this vol.): It differs from traditional LFG notations only in [] for optional constituents and ! for existential constraints. The Kleene star used in functional schemata is not yet implemented and is 'simulated' at the moment by annotating appropriate disjunctions.
5) See HAIDER (1985) for some recent analysis of the sentence initial positions in the framework of G&B.
6) For the use of variables ranging over sets of grammatical functions see WEDEKIND (1986).
7) This prediction is relevant for the difference between the following sentence pairs:

(i) Er hat versucht dieses Buch zu lesen.
(i') Dieses Buch hat er versucht, zu lesen.
(ii) Er ist gekommen, dieses Buch zu lesen.
(ii') *Dieses Buch ist er gekommen, zu lesen.

The extraposed VP in (i) is to be analysed as XCOMP, the VP in (ii) as XADJUNCT.
9) See JOHNSON (1985) whose description has been modelled after the approach given in BRESNAN et al. (1982).

REFERENCES

- BECH, Gunnar (1955), Studien ueber das Deutsche Verbum Infinitum. 2 vol. Kopenhagen
- BRESNAN, Joan (1982a), Control and Complementation. In: BRESNAN (1982b), 282-390
- BRESNAN, Joan (ed.) (1982b), The Mental Representation of Grammatical Relations. Cambridge, Mass.
- BRESNAN, J./R.KAPLAN/S.PETERS/A.ZAENEN (1982), Cross-Serial Dependencies in Dutch. In: Linguistic Inquiry 13.4
- EISELE, A./J. DÖRRE (this vol.), A Lexical Functional Grammar System in Prolog.
- HAIDER, Hubert (1985), Verb second in German. To appear in: HAIDER, H. / PRINZHORN, M. (eds.), Verb Second. Dordrecht.
- JOHNSON, Mark (1985), An LFG Description of the Double Infinitive Construction in Dutch and German. unpubl. Ms., Stanford
- KAPLAN, R./J.BRESNAN (1982), Lexical Functional Grammar. A Formal System for Grammatical Representation. In: BRESNAN (1982b), 173-281
- KAPLAN, R./A. ZAENEN (1986), Functional Incertainty in LFG. unpubl. ms., Stanford
- WEDEKIND, Juergen (1986), Ein Vorschlag zur nicht - konfigurationalen Kodierung grammatischer Funktionen in der LFG. in:Forum angewandte Linguistik, vol. 14, Tuebingen