Notice

This paper is the author's draft and has now been published officially as:

Micelli, Vanessa (2012). Field Topology and Information Structure: A Case Study for German Constituent Order. In Luc Steels (Ed.), *Computational Issues in Fluid Construction Grammar*, 178–211. Berlin: Springer.

BibTeX:

```
@incollection{micelli2012fieldtopology,
   Author = {Micelli, Vanessa},
   Title = {Field Topology and Information Structure:
            A Case Study for German Constituent Order},
   Pages = {178--211},
   Editor = {Steels, Luc},
   Booktitle = {Computational Issues in {Fluid Construction Grammar}},
   Publisher = {Springer},
   Series = {Lecture Notes in Computer Science},
   Volume = {7249},
   Address = {Berlin},
   Year = {2012}}
```

Field Topology and Information Structure: A Case Study for German Constituent Order

Vanessa Micelli

Sony Computer Science Laboratory Paris, France

Abstract. A widely used approach for handling German constituent ordering is based on the so called field topology surface model. This model proposes five fields of varying complexity in linear order whereby each field imposes more or less flexible constraints on the number and the types of sentence constituents it can capture. Both the placement of constituents into fields and the order of the constituents within a field can vary widely subject to an intricate interplay of diverse constraints including information structure. This chapter works out a complete operational solution illustrating this field topology approach within the context of Fluid Construction Grammar. It focuses in particular on the double object construction in ditransitive sentences.

1 Introduction

> An average sentence, in a German newspaper, is a sublime and impressive curiosity; it occupies a quarter of a column; it contains all the ten parts of speech – not in regular order, but mixed;" ([40], Appendix D).

German language learners in particular will agree with Mark Twain that German constituent order presents a difficult but also intriguing subject of study. Despite several strict rules, it is often the case that German constituent order is quite free. For instance, the finite verb must appear in the second position in declarative sentences but subject or objects can shift around without restriction.[1]

One way of formally describing the seemingly free constituent order in German sentences is through a topological surface model. This model traditionally proposes a maximum of five fields of varying complexity in linear order as discussed for example in [29].[2] Each field then imposes constraints on which and also on how many parts of the sentence it can capture. There are ongoing debates on the acceptability of constituent orders in German utterances. Examples of potential factors determining constituent order are focus, definiteness,

[1] This claim only holds as long as the noun phrases are either case marked or identifiable by the context they appear in or by selectional restrictions. Else reordering is not freely permitted (see [46, 45]).

[2] [19] gives a review of the topological fields. He proposes one additional field preceding the Vorfeld to accommodate left-dislocated-elements. This approach is, however, not accounted for in this case study.

case and animacy. (See for instance [16, 21, 24, 41].) In the present study, we focus on the constraints identified by [24]. He found that German constituent order, especially within the proposed fields, is subject to an intricate interplay of a number of diverse semantic, pragmatic or syntactic constraints. Significantly, these constraints must interact with aspects of *information structure* that are often ignored or treated as peripheral to grammar rather than as integral to it.

The information structure of a sentence refers to how that sentence is structured regarding its focus, ground, topic, comment and so forth. Identification of those issues within a sentence is still discussed in the literature, and so far no common consensus has been reached. Although it is often described in terms of general discourse principles or pragmatic parameters, empirical studies show that almost every language has developed concrete strategies for marking information structure. Within those languages, those strategies are conventionalized and are more systematic than often assumed.

This chapter tries to answer the question as to how a speaker's knowledge of the information structure of his or her language can be operationalized so that it can influence constituent order, focusing as a first step on the double object construction in declarative sentences in German. The chapter describes a Fluid Construction Grammar (FCG) implementation that follows a field topological approach and that tightly incorporates information structure into the grammar, making it an integral part of it. We refer to other papers in this volume [38] as well as [36] for introductions to FCG. The grammar is fully operational and can be used for both parsing and production of German interrogative and declarative sentences with intransitive, transitive or ditransitive verbs. This achievement is noteworthy as many other approaches to grammar are often purely descriptive.

The main points of the grammar implementation described in detail in this chapter are the following:

1. As previously suggested by [11], information structure has to be tightly integrated into the grammar, as it interacts both with phonology and syntax. FCG's openness with respect to features and constraints offers an elegant way for achieving this. To capture German's complex constituent order it is important to deal with various constraints on various levels of the language, which necessarily interact with aspects of information structure. Therefore, the syntactic, semantic and pragmatic constraints identified by [24] will be explicitly represented, evaluated and determined during language processing. Their states are accessible to other constructions from any other part of the grammar at all times. Pragmatic information is incorporated into the grammar without assuming a separate discourse layer so that all 'levels' of language are tightly coupled in a non-modular way.
2. The grammar engineer can either put all constructions in one big 'constructicon', (one construction set), or group them into several construction sets, whose order of application can then be controlled. The second option is chosen in this study, although it may seem to conflict at first glance with a non-modular approach to grammar. However, one must keep in mind that

> constructions can reach across layers and they only apply when all their constraints are met. So most of the time it is not at all relevant at which point in processing they operate. On the other hand, sometimes constraints in constructions are left as variables (i.e. they are underspecified). Therefore constructions might apply too early, thereby imposing an incorrect value on an underspecified constraint. One way of preventing this strongly undesirable behavior is to put the constructions that determine those values correctly into one construction set. It then has to be specified that this set is applied before other constructions which rely on these constraints.

In order to determine the focus of a proposition, it is common practice to assume that the proposition provides the answer to a lead-in information question. This question sets the context and ensures that the answer incorporates the most appropriate information structure, i.e. focus-marks the sentence constituent corresponding to the WH-constituent of the question [7, 23]. The presented grammar serves as the language representation in such a question-answering dialogue and can be used both for producing and interpreting questions and respective answers.

The remainder of this chapter is structured as follows: Section 2 presents the linguistic background of the present case study, introducing the notions of field topology and information structure. Section 3 takes a step back from the actual implementation and describes the different steps necessary for engineering a grammar, including the previously discussed linguistic findings. Section 4 highlights the main design patterns of the implementation, with Section 5 diving into several implementation details and presenting the constructions needed to produce an utterance. Section 6 briefly summarizes the handling of information structure in several other grammar formalisms and concludes the chapter.

2 Constituent Order of German Declaratives

The ordering of constituents in German declarative sentences is an intriguing subject, because, although several strict rules non-ambiguously determine some sentence constituents' position, it is often the case that German exhibits quite a free constituent order. The approach in this chapter assumes that German constituent order can be accounted for by following rules forming a topological model. This model divides sentences into topological domains or fields. All sentence constituents are captured by a field and then those fields are linearly ordered. Information structure plays a role in partially determining the constituent order within sentences and determining the ordering inside those topological fields that capture more than one constituent. This section introduces the notions of *field topology* and *information structure*.

2.1 Field Topology

The topological surface model traditionally proposes a maximum of five fields of varying complexity in linear order as discussed for example in [18] and displayed in Figure 1.[3] The two sentence brackets (called *linke Klammer* (LK; 'left bracket') and *rechte Klammer* (RK; 'right bracket')) form the sentence frame, embracing the *Mittelfeld* (MF; 'middle field'), preceded by the *Vorfeld* (VF; 'fore field'), followed by a *Nachfeld* (NF; 'end field'). All of those fields are optional, however, some of them more than others. For example, the linke Klammer is typically occupied by a finite verb, however, never in relative clauses.[4]

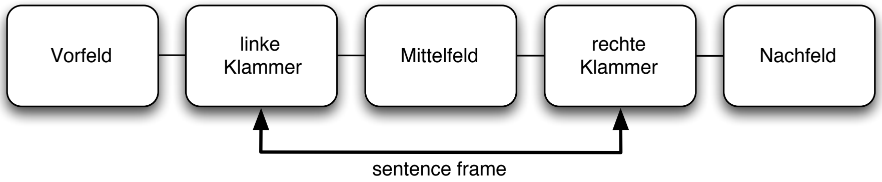


Fig. 1. Model of the five topological fields.

Each of these fields imposes constraints about the type and number of sentence constituents it can capture. The finite verb, for instance, is always positioned in the LK, whereas non-finite verb forms as for example the past participle, in case the verb is in perfect tense are considered to be always included in the RK. This study does not account for cases where the participle occurs in another position as this goes beyond the scope of this study. Here we only consider possible declarative sentences which can answer a WH-question about one of the event participants. The Vorfeld is generally composed of exactly one sentence constituent (every constituent is permitted except the finite verb).[5] Figure 2 shows an example of how a sentence can be divided into five fields.

Despite the previously mentioned constraints determining which constituent can go into which field, this sentence has more than a dozen possible constituent orders. The ordering of constituents in the Mittelfeld seems to be relatively free, but there are factors of varying nature that interact with one another and affect the sentence constituents' order in that field, particularly factors related to information structure.

[3] There is, however, controversial literature highlighting possible problems when assuming this approach. (See for instance [28].)

[4] For more detailed information on topological fields or a historical overview see for instance [18], [19] or [10] who first used the field-based terminology. For a discussion of sentences with an empty left bracket see [28].

[5] In this chapter, the investigations are constrained to declarative sentences and WH-questions. There are, of course, exceptions to the mentioned rules, but these are beyond the scope of this case study. See [26, 27] for an HPSG approach to account for multiple frontings in German.

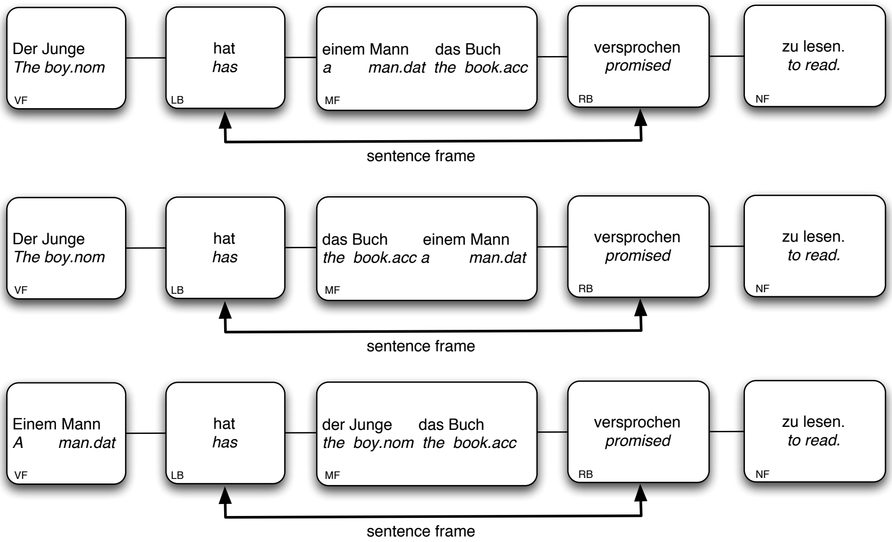


Fig. 2. Example sentences demonstrating three possibilities of placing sentence constituents into the fields. English translation: "The boy has promised a man to read the book". The glosses show the literal English translation

2.2 Information Structure

The term *information structure* has been first introduced by Halliday [14]. It refers to information that is provided to the listener about what aspect is assumed to be in focus, what is given and new, what perspective is emphasized. Within a shared communicative context, a speaker aims at avoiding misunderstandings in communication. Therefore, he follows certain conventions with regard to information structure so that the listener can understand it with minimal processing effort [45].

Strategies of Information Structure Different languages use different strategies to express information structure in their utterances depending on different communicative circumstances. It can be expressed prosodically, by grammatical markers or by a specific order of syntactic constituents in the form of complex grammatical constructions. English, for instance, uses constituent order as in (1) and (2), with possibly additional intonational stress or lexical means as in (3), to emphasize particular parts of sentences (underlined in the example sentences):

(1) *He likes flowers.*

vs.

(2) *Flowers he likes.*

(3) *As for flowers, he likes them.*

A language can have competing language strategies for expressing similar phenomena. German, like English, is an intonation language. Therefore, one of the strategies used for expressing information structure is the use of pitch accents on new or prominent parts of the sentence [32]. Those parts are here called the *focus* of the sentence. A different, frequently used strategy is for instance fronting of the most salient sentence constituents as exemplified in (3) for English.

Focus-marking in Question-Answering One of the main assumptions of traditional analyses of information structure is that the utterance answers a previously asked question which presents the relevant context [17, 42]. Different questions (see (4) and (6)) force different prosodic focus-marking (by using a pitch accent) on the truth-conditionally similar answers as exemplified in the following example sentences (The focus-marked constituent is put in square brackets; the index FOC stands for focus.):

(4) *Wer gibt dem Mann das Buch*?
Who gives the man the book?
transl.: 'Who gives the book to the man?'

(5) [*DerJunge*$_{FOC}$] *gibt dem Mann das Buch.*
The boy gives the man the book.
transl.: 'The boy gives the book to the man'

(6) *Was gibt der Junge dem Mann*?
What gives the boy the man?
transl.: 'What does the boy give the man?'

(7) *Der Junge gibt dem Mann* [*einBuch*$_{FOC}$].
The boy gives the man a book.
transl.: 'The boy gives the book to the man'

In these cases, the use of pitch accent focus-marks that part of the utterance asked for by the indicated question.

In this chapter, we focus exclusively on focus-marking for the following reasons:

1. Information about focus-marking is helping to account for an acceptable order of two objects in the Mittelfeld. (See the end of this section.)
2. Through the use of pitch accent the cognitive effort for the hearer is reduced, because the part of the response that answers directly the question is made more salient.

3. In some cases, intonation is necessary for disambiguation.

Example sentences (8) – (11) present a case where intonation helps disambiguate which sentence constituent plays which role in the situation. Sentence (9) presents an ambiguous answer to the question in (8): It is not clear who is seeing whom in that answer.

(8) *Wen sieht sie?*
transl.: 'Whom does she see?'

(9) *Die Professorin sieht die Studentin.*
transl.: 'The professor sees the student.' or
'The student sees the professor.'[6]

Sentences (10) and (11) are, however, unambiguous: in (10) the student sees the professor, in (11) the professor sees the student. However, the constituent order of both sentences is identical, and only intonation disambiguates the two different readings.

(10) $[DieProfessorin_{FOC}]$ *sieht die Studentin.*
transl.: 'The student sees the professor'.

(11) *Die Professorin sieht* $[dieStudentin_{FOC}]$.
transl.: 'The professor sees the student'.

The assumption that focused noun phrases are emphasized through prosodic means (through pitch accent) is based on Selkirk's findings [31]. She argues that questions allow control over which syntactic constituent in the answer has to be emphasized.

The present case study will not make a fine-grained difference of nuclear or prenuclear accents of declarative sentences. It additionally does not distinguish between rising or falling accents (adapted for German e.g. in [13]) as we are only interested in general focus-marking of newly introduced noun phrases to help determine constituent order. In this chapter we follow [30, 32] and use the term *focus* in the sense of particularly prominent or new. We concentrate on how information structure, and to be more precise, how focus-marking helps determine the ordering of noun phrases (NPs) in the Mittelfeld. To determine unmistakably which element should be focus-marked in an utterance, a preceding question is assumed.

Different Criteria Influencing Constituent Order Certainly there are several properties of very different kinds that have an impact on constituent order. Some studies concentrate on definiteness [22], animacy [16, 21] or length of the argument NP in question. (see [1, 139], [2, 86], or [15]). In his seminal work on constituent order in German, Lenerz identified three different constraints on definiteness, case and focus-marking that co-determine the ordering of noun phrases in the double object construction in the Mittelfeld [24]:

[6] The first reading is preferred slightly more, as 'the professor' is in front position of the sentence.

1. definite NP *precedes* indefinite NP (*Definiteness constraint*)
2. non-focused NP *precedes* focused NP (*Focus constraint*)
3. dative NP *precedes* accusative NP (*Case constraint*);[7] *unmarked* constituent order

These three constraints are taken into account in this study.[8] Speakers of German have to establish knowledge of these constraints to successfully produce and understand utterances. There are five valid ordering scenarios regarding the interplay of the identified constraints. Most interesting about those constraints is their complex interaction with each other. Table 1 displays the valid scenarios with the respective truth values of the constraints.

Scenarios	Definiteness	Focus	Case
1	+	+	+
2	+	+	−
3	−	−	+
4	−	+	+
5	+	−	+

Table 1. Constraints and their interaction between each other (+ means that a constraint is met, – means that it is not).

Assuming the question in (6), the example sentence in (7) displays Scenario 1. The sentence in (12), however, presents an example of an invalid ordering of NPs in the Mittelfeld, assuming the same preceding question.

(12) Der Junge gibt ein Buch FOC dem Mann.
The boy gives a book FOC *the man.*
transl.: 'The boy gives the man a book'.

The answers in (14) and (15) to the question in (13) present the Scenarios 2 and 3 respectively.

[7] Especially the case constraint is currently under debate: As described for instance in [25] or [8], the order of sentence constituents in the MF is dependent on the kinds of verbs and the type of its arguments. Cook suggests, for example, to not assume a case constraint as suggested by Lenerz but to divide dative objects into low and high datives regarding their grammatical function. Noun phrases are then ordered in the MF depending on their grammatical function and not on their case. This approach has the advantage that the grammar then captures all kinds of ditransitive verbs not only those that are captured by following Lenerz' approach. However, in this chapter we follow solely Lenerz seminal study and consider the further division of dative objects as future work.

[8] To ensure that two objects occur both in the MF, it is assumed in this case study that the subject of the sentence is fixed in first position. This decision, however, is not linguistically motivated but chosen to ensure the occurrence of the double object construction in the MF.

(13) Wem gibt der Mann das Buch?
Whom gives the man the book?
transl.: 'To whom does the man give the book?'

(14) Der Mann gibt das Buch einem Jungen FOC.
The man gives the book a boy FOC.
transl.: 'The man gives the book to a boy.'

(15) Der Mann gibt einem Jungen FOC das Buch.
The man gives a boy FOC *the book.*
transl.: 'The man gives the book to a boy.'

Lenerz' findings can be summed up as follows: As soon as the case constraint holds, i.e. as soon as the dative NP precedes the accusative NP, it is irrelevant which of the remaining two constraints is positive. For this reason, this constituent order is called *unmarked* [24].[9] However, in case it does not hold, both of the other two constraints have to be met (*marked* constituent order). All these special constraints and their interaction between each other are explicitly integrated into the present FCG case study.

3 Progressive Plan Refinement

Before diving into the actual grammar implementation, it is important to consider which different steps have to be taken when composing an utterance using a grammar that integrates the previously discussed linguistic findings. This planning process collects different constraints, tests their validity, makes refinements and in the end, composes the utterance, satisfying all constraints, hopefully leading to communicative success. After the first and foremost goal has been reached in the process of producing an utterance, i.e. **what** will be said has been decided, the planning of the actual **how** to say it starts: what are the utterance's smallest constituents?

3.1 Which Words to Use?

Let us focus on the sentence "The man gives the book to a boy". A question can be asked about each event participant. Here it is assumed that the question in (13) (repeated here for convenience in (16)) precedes the production of the utterance and sets the context necessary to determine which event participant has to be focus-marked.

(16) Wem gibt der Mann das Buch?
Whom gives the man the book?
transl.: 'Whom does the man give the book to?'

[9] See additional remarks on unmarked constituent order in Section 6.

Producing an utterance is a sequential process in which the planning is constantly refined. It generally starts with selecting the required lexical items depending on the speaker's lexicon. There are several possible answers to the question in (16), two of which take into account the constraints mentioned in the last section.

(17) Der Mann gibt das Buch einem Jungen FOC.
The man gives the book a boy FOC.
transl.: 'The man gives the book to a boy.'

Another appropriate answer, investigated more closely here, is the following:

(18) Der Mann gibt einem Jungen FOC das Buch.
The man gives a boy FOC *the book.*
transl.: 'The man gives the book to a boy.'

When composing the utterance that answers the context question, the previously mentioned constraints are considered: the event participant being asked for is made more salient by being marked with a pitch accent and the noun phrase newly introduced to the context is preceded by an indefinite article, according to convention.[10]
Figure 3 shows which lexical constructions are used to compose the utterance. Each lexical item is a bi-directional mapping of a form which traditionally is constituted by the actual string and a meaning. A first order predicate logic-based representation of the meaning is used, which, for the lexical item Buch (engl.: 'book'), looks like this: `(book ?x)`. The predicate `book` stands for the ontological class describing books, and the variable `?x` can be bound to the actual book in the real world, which is referenced in this specific context. Determiners do not yet have a form, depending on the semantic and syntactic role that the item they are combined with plays. The appropriate string is added in a later step. For simplification, Figure 3 does not list all predicate-logic meanings but only the placeholder `meaning`. (See Figure 11 for a meaning representation of the complete sentence.)

The foundational material for building the utterance has been collected. The next step is to decide which low-level constituents can be determined thus far.

3.2 What are Possible Sentence Constituents?

Each noun is preceded by a determiner, therefore, in this step nouns and determiners are combined into determined noun phrases. Syntactic and semantic information from each of the two components of a determined noun phrase are percolated up to the newly created noun phrase. In addition, the construction creating the noun phrases adds additional information both to its form part (i.e.

[10] This means that we assume that the NP has not been aforementioned to ensure the possibility to determine constituent order in the MF by accounting for Lenerz' findings (here the determination constraint respectively).

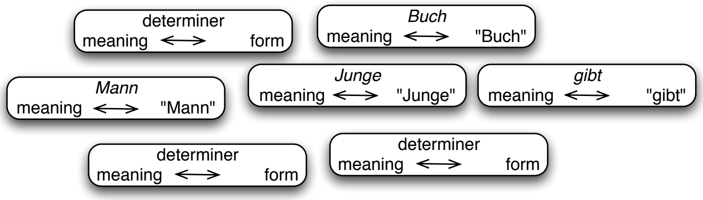


Fig. 3. All lexical items needed to produce the utterance.

constituent order constraints that the determiner has to precede the noun) and to its meaning part. The speaker wishes to emphasize the noun phrase asked for ('einem Jungen'), in order to minimize the cognitive effort of the hearer. Focus-marked sentence constituents are more salient than others, which is why the hearer immediately draws his or her attention to that constituent. Formally, this is represented by adding the marker "FOC" following the noun phrase in question. Figure 4 shows the constituents identified so far.

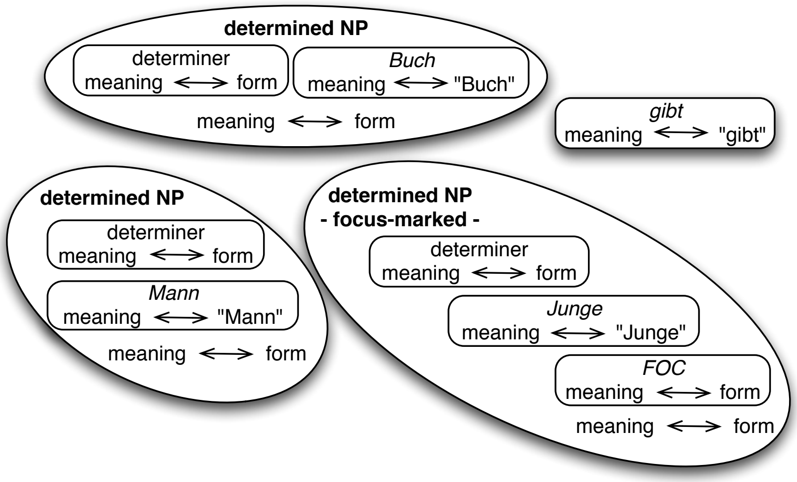


Fig. 4. The sentence constituents which so far have been identified.

3.3 Which Role Do the Sentence Constituents Play?

Knowing which sentence constituents are at disposal, their syntactic and semantic roles in the sentence can be assigned. Traditionally, argument structure constructions link semantic roles of event participants to their grammatical roles, and additionally they often impose constituent order on their constituents. (See for instance [43]). However, here constituent order is only accounted for in the very last step of building the utterance, when the topological fields have been determined and can be sorted. One of the reasons for this approach is that the same argument structure constructions can apply for both declaratives and questions.

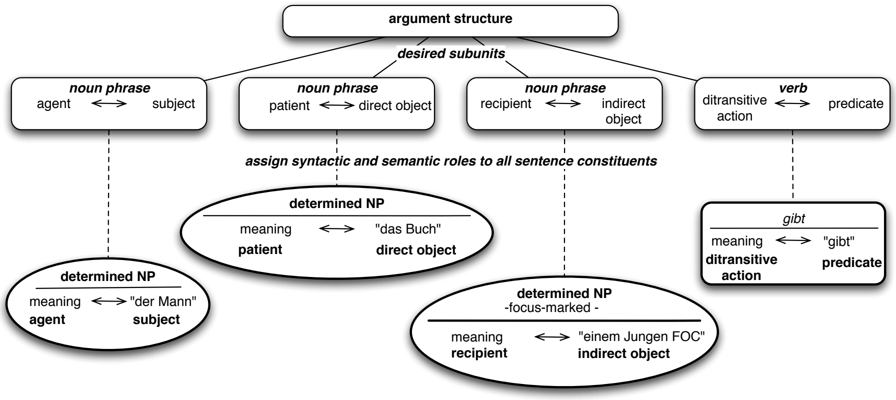


Fig. 5. Event participants are assigned their semantic and syntactic roles.

After each sentence constituent's semantic and syntactic role in the sentence has been identified, the determiners can be assigned their final forms. Figure 5 shows the constituents identified so far, their semantic and syntactic roles and the forms of the appropriate determiners. It shows that an argument structure construction asks for four subunits which are assigned semantic and syntactic roles, assuming that they meet the constraints the construction imposes on them beforehand.

3.4 What Will Be the Order of the Two Objects in the Mittelfeld?

Before putting sentence constituents that have been determined so far into the appropriate topological fields, another decision has to be verified: Will the order of the two objects in the Mittelfeld be *marked* or *unmarked*? (See Figure 6.) This study accounts for three different constraints determining the constituent order of the two objects in the double object construction. Those constraints are all initially underspecified as presented in Figure 6 by a question mark (?) following the three keywords designating the three mentioned constraints: case, focus and definiteness.

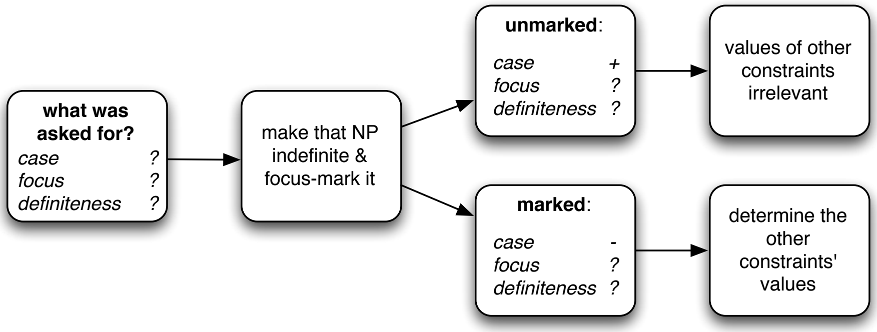

what was asked for?
case ?
focus ?
definiteness ?
make that NP indefinite & focus-mark it
unmarked:
case +
focus ?
definiteness ?
values of other constraints irrelevant
marked:
case -
focus ?
definiteness ?
determine the other constraints' values

Fig. 6. Which order in the Mittelfeld is preferred (marked or unmarked) and what are the subsequent steps after the decision has been made?

In the case of the chosen constituent order being unmarked (as in sentence (16)), the case constraint is satisfied, i.e. it is positive. This decision is made when exactly **what** to say is decided, i.e. the predicate `(unmarked)` is listed in the meaning of the complete utterance to be produced. (See Figure 11.) Figure 6 shows that the question mark following the term case has been substituted by a +, because the constituent order is going to be unmarked. This step does not happen magically, but there are mechanisms in the grammar whose job it is to check if a constraint is met and determine the explicitly presented values of the constraints. As soon as the case constraint is determined to be positive, however, it is irrelevant if the other two constraints are fulfilled. The next step in the plan can be executed, i.e. which constituent is captured by which field and – in the case of the Mittelfeld – in which order are the constituents put into the field? (See Section 3.5.)

Before that step is executed, the case should also be considered in which marked constituent order has been chosen (i.e. the accusative object NP precedes the dative object NP) (see (17)). In that case, the two other constraints regarding focus-marking and definiteness have to be determined (and in fact have to be met) in order to yield an appropriate utterance. Figure 7 shows the decision tree which is traversed in that case. Having decided on producing marked constituent order in the Mittelfeld, the case constraint can be assigned a negative value, which means that there is a split in the tree: either the focus constraint or the definiteness constraint has to be checked as a subsequent step. If the first one picked tests negative, the value of the remaining last constraint is irrelevant, since no appropriate ordering of the noun phrases is possible. Remember, that if the case constraint does not hold, both other constraints must be positive. Therefore, ideally at the end of this planning phase all three constraints will have been determined, and will describe Scenario 2 of Table 1.

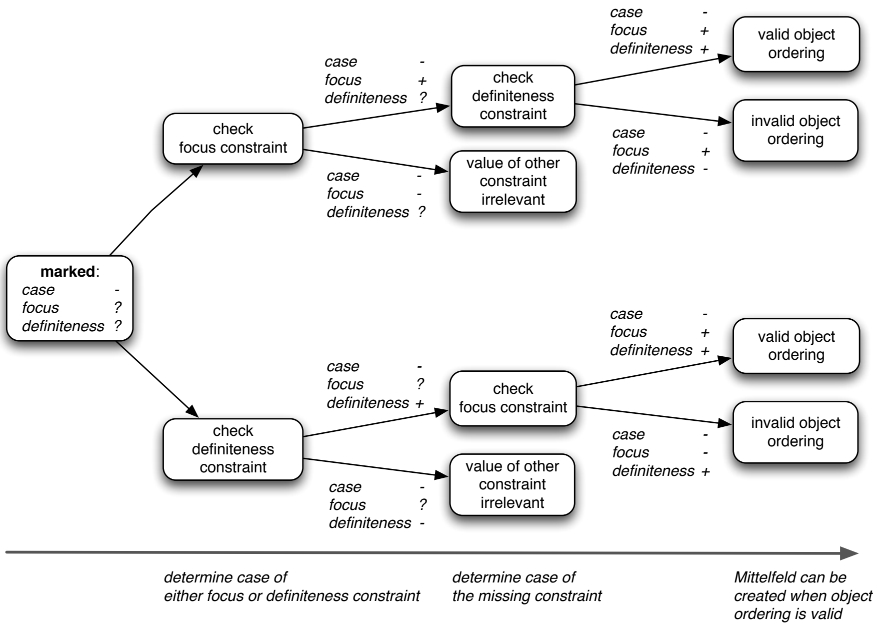


Fig. 7. When marked constituent order is preferred, all other constraints have to be checked successively. Both the definiteness and the focus constraint have to be positive to yield appropriate object ordering.

3.5 Which Sentence Constituent Goes into which Field?

As already mentioned, each field poses specific constraints on the constituent it can capture. If the constraints are met, the sentence constituent in question is put into the respective field. In this example study, the subject is always preferred to be in the Vorfeld, to make sure that the two objects are definitely in the Mittelfeld. Their order has already been determined in the last step where the three constraints and their validity were checked. The linke Klammer is in declarative sentences typically occupied by a finite verb. In the example sentence three topological fields result, which capture all of the present sentence constituents but which still have to be ordered linearly. Figure 8 illustrates the state in processing identified so far.

3.6 How Are the Topological Fields Ordered?

The last step of the plan still has to be executed: The topological fields have to be put into linear order. The Vorfeld is always in first position, followed by the linke Klammer, which is in turn followed by the Mittelfeld in which the order

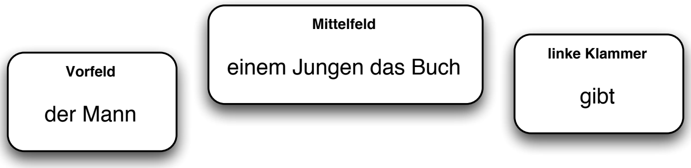


Fig. 8. Identified topological fields capturing their respective constituent(s) in arbitrary order. The order of the two objects in the Mittelfeld has already been determined.

of constituents has previously been determined (see Figure 9). The construction that creates the linear order of the fields is a declarative construction, which is not affected by the type of constituents in the fields but simply puts those fields that are at its disposal into the appropriate order. In this example this means that after the declarative construction has applied, the Vorfeld precedes the linke Klammer, which in turn precedes the Mittelfeld.

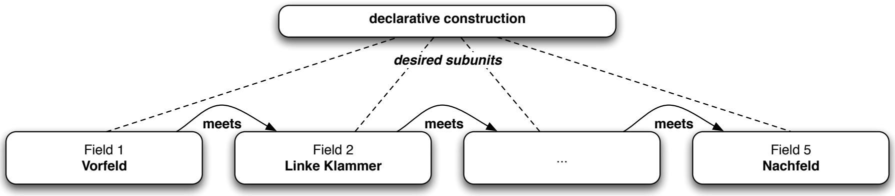


Fig. 9. A declarative construction puts the fields into linear order. The field containing the dots stands for any field that is put within the Linke Klammer and the Nachfeld.

3.7 Analyzing an Utterance

To parse the previously discussed utterance, exactly the same mechanisms are used. All steps are taken similar to the production process described above but starting from the surface form. The goal in this process is to pick the utterance apart and collect the sentence's expressed meaning. The order of the application of the mechanisms has, however, been slightly altered. First all lexical items (including the determiners) are picked out, and their meanings are collected. Determined noun phrases are then detected, semantic and syntactic roles of all event participants are assigned, the validity of the constraints are checked, fields and their constituents are assigned and the application of the declarative construction shows that the utterance is in fact a declarative.

After having discussed on a more general level how an utterance is built by assuming the approach of creating topological fields, sorting them and incorporating information structure to determine the fields' constituents' order, the next section presents some actual design patterns used in the applied FCG grammar to operationalize all of what has been mentioned before.

4 Design Patterns

This section highlights the main design patterns that have been used to implement the linguistic decision-making process discussed in the previous section.

4.1 Feature Matrices

The language user is faced with a complex search problem and needs to manage this problem in an efficient way. In the same way, a computational system has to find an efficient way to deal with it. One possible way to deal with the complex German case system is the use of feature matrices [44]. It is more efficient to represent a set of constraints (as case, gender and number) as attributes that take an array of values. Postponing the decision of a noun's case, gender and number to a point in the analysis where there is no longer ambiguity avoids unnecessary search or backtracking, in case a wrong hypothesis has been followed. As a result processing cost is highly reduced.

However, not only are feature matrices used to determine the case, gender and number of noun phrases or definiteness in determiners, attribute-value pairs are also put into the top-unit by the argument structure construction. Those attributes and their values are used to help determine the final constituent order of the double object construction in the Mittelfeld. That means that the final surface form of the utterance depends on a complex interaction of constraints. Only when the validity of the constraints has been identified, can the final constituent order be determined.

4.2 Construction Sets

Despite the fact that the linguistic inventory is a continuum from lexicon to grammar, various sets of constructions can be identified based on their function in the grammar. Particularly complex sentences, such as the one examined in this chapter, involve a considerable number of constructions of a different nature. Although FCG can also autonomously organize the search [47], the process of identifying the various **types** of constructions involved in the sentences' analysis is a first and necessary step into understanding and operationalizing a grammar. Hence, constructions are grouped into several construction sets regarding their function in interpretation and production.[11] (See Figure 10.) All constructions are bi-directional form–meaning pairings, however, they differ in their complexity.

[11] For more details on construction sets see [3].

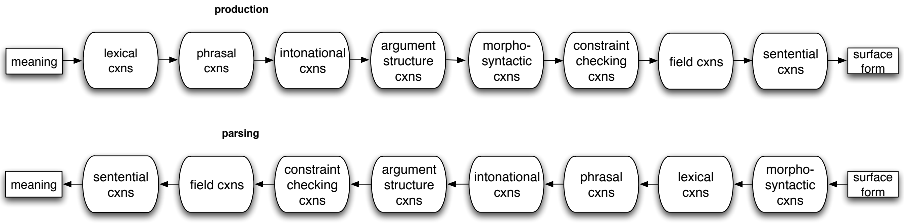


Fig. 10. The order in which FCG constructions are applied in production (upper part) and in interpretation (lower part).

Another major reason for arranging constructions in different sets is that the order of processing can thereby be controlled. Normally, the order of application does not matter, as constructions only apply when their conditions are fulfilled. However, aside from the fact that they can be grouped in functionally similar sets, making the application process more readable and easier to follow, it is important that some constructions only apply after other sets have already applied. Argument structure constructions add attribute value pairs to the top-unit, where the value is initially a logic variable. (See paragraph on explicit constraint representation below.) Later in processing, there are constructions that check the value of those constraints, whose application requires for instance either an explicit negative value (i.e. a –) or a positive one (+). However, construction application can also successfully happen if the value of the constraint is still a variable for which the technical reason is that `(unify '(a b c) '(a b c))` clearly unifies, however, `(unify '(a ?x c) '(a b c))` unifies as well and binds the variable `?x` to `b`. Certainly, this binding is not desired in this case where all variables should first clearly be determined.

4.3 Ontological Categories

Constructions usually have a `syn-cat` and a `sem-cat` unit-feature that list the syntactic or semantic categories, respectively. Examples are for instance the part of speech of a word, its `function` like `nominal` or a plain semantic category like `entity`. These categories are further used by higher-level constructions: For instance a determiner-noun-construction asks for two subunits: one whose `function` is `nominal` and whose semantic category is an `entity` and one whose part of speech is `determiner` and whose semantic category contains a definiteness value. When having found units that conform to those constraints, these units are combined into a determined noun phrase. In this grammar, whenever necessary, instead of single category values, a list of semantic or syntactic categories is given, describing the same item, but with differing granularity or specificity.

Let's have a closer look at one example construction: The `sem-cat` feature of the construction describing the lexical item *house*, for instance, lists the semantic categories (entity container). For a determiner-noun-construction it is enough to know the actual house belongs to the semantic category entity, which, therefore, can be combined with a determiner. However, some constructions, for instance the one for the lexical item *enter*, should solely be combined with nouns of a specific semantic category as, for instance, container. This approach ensures that *house* can be the object of verbs like *to enter*.

An equivalent method is adopted for syntactic categories: Ditransitive sentences have two objects: a direct and an indirect object. Both belong to the more general category object, therefore their syn-cat feature includes both (direct-object object) and (indirect-object object) respectively. A higher level construction like the one building the Mittelfeld for questions considers only that the constituents it captures are objects and not them being direct or indirect. Other constructions, however, need to make that distinction and therefore need more fine-grained information.

This method is chosen to account for simple ontological 'hierarchies', which theoretically can be as fine-grained as desired. *Hierarchy* here does not imply the integration of a real ontological hierarchy but refers to a list of ontological categories which are not linked in the construction at all. However, those categories can be mapped to ontological classes in a separate ontological model where properties and links have been established. The downside of this method is that so far many categories have to be copied and manually integrated into constructions again and again.

4.4 Explicit Representation of Constraints' Status

Constituent order in German is influenced by various kinds of constraints. Three of them and their interaction among each other have been discussed in 2.2. Those are all treated in the same way in the grammar implementation: The argument structure construction adds one attribute-value pair for each constraint to the top-unit of the transient structure representing the status of each of those constraints. This also shows that information structure as it is represented here is strongly interwoven with the rest of the grammar. The following pairs are explicitly added to the top-unit in a newly created unit-feature called constraint-status:

```
(constraint-status (==1 (focus-constraint ?fc)
                        (definiteness-constraint ?dc)
                        (case-constraint ?cc))
```

Each attribute is followed by its value, which initially is a variable, indicated by the ? preceding it. Only when it can be decided unambiguously which constraint is met, will the respective variable be changed either into a + (indicating that this constraint is met) or a – (indicating that it is not met). Similar category value pairs, where the value can either be a +, a – or a variable, will be used throughout the grammar on both the semantic and syntactic poles.

All mentioned high-level design patterns of the grammar will recur in the following section, where parts of the actual implementation are described.

5 Operationalization

Now that the linguistic background and more abstract ideas behind the presented grammar have been examined, technical issues that arise during operationalization can be addressed. Basic knowledge about FCG is assumed [37]. In the course of this section, several constructions are highlighted that contribute to the production of the answer to the respective question in (18). The answer is repeated in (19) for the reader's convenience.

(19) Der Mann gibt einem Jungen FOC das Buch.
The man gives a boy FOC *the book.*
transl.: 'The man gives the book to a boy.'

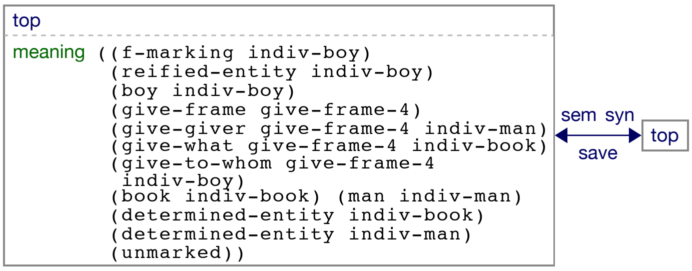
Extensive detail of the application of all of these constructions is not provided, rather mostly how they are created and what they look like is highlighted.[12] The interested reader is referred to `www.fcg-net.org` to inspect a complete dialogue where a question as well as a respective answer is produced and also parsed using the presented constructions. In the web demonstration, single constructions, as well as transient structures and their modifications after construction application, can be inspected. In the following sections, the constructions are described in their order of application in production as depicted in Figure 10.

5.1 Initial Linguistic Structure

Each production process starts with the creation of an initial transient structure that couples a syntactic and a semantic structure or pole. The semantic structure specifies the meaning of a sentence, while the syntactic structure specifies its form. Linguistic processing essentially involves the mapping of a semantic structure to a syntactic structure (i.e. meaning to form) during production, and the inverse during parsing. In terms of FCG, this is accomplished through the application of constructions to a transient structure. The consecutive application of constructions adds additional units and constituent structure to both poles of the structure as well as additional parts of meaning and form. All meaning to be rendered into an utterance is stored in the so-called *top-unit* (cf. Figure 11). The meaning is represented in first order predicate logic expressions.

Constructional meanings are stored in a frame-based ontology. A construction's meaning is represented by one of the frames present in the ontology, i.e. the value of its meaning feature is equivalent to a frame in the ontology. For example, the frame representing the meaning of the verb *to give* looks as follows:

[12] See [5] for a detailed explanation how constructions are applied.



```
((f-marking indiv-boy)
 (reified-entity indiv-boy)
 (boy indiv-boy)
 (give-frame give-frame-4)
 (give-giver give-frame-4 indiv-man)
 (give-what give-frame-4 indiv-book)
 (give-to-whom give-frame-4
  indiv-boy)
 (book indiv-book) (man indiv-man)
 (determined-entity indiv-book)
 (determined-entity indiv-man)
 (unmarked))
```

Fig. 11. Initial transient structure. The semantic pole (to the left of the arrow) contains the meaning which will be rendered into an utterance, the syntactic pole (to the arrow's right) is still empty.

```
((give-frame ?give-frame)
 (give-giver ?give-frame ?a-giver)
 (give-what ?give-frame ?the-given-thing)
 (give-to-whom ?give-frame ?the-recipient))
```

Again, names starting with a question mark are variables. The use of the same variable name ensures that every instance of its use refers to the same referent. You can see in Figure 11 that those variables which designate the actual objects and events the speaker wishes to talk about have been replaced by unique symbol names. For instance, instead of the variable `?give-frame`, an indexed `give-frame-4` is used, referring to the give-action in the context.

5.2 Lexical Constructions

To define a lexical construction, the same templates as described in [37] are used. The following template creates a lexical construction called `Mann-cxn` by first creating a skeleton, then incrementally adding syntactic and semantic categorizations and finally a feature matrix to account for case:

```
(def-lex-cxn Mann-cxn
 (def-lex-skeleton Mann-cxn
          :meaning (== (man ?man))
          :args (?man)
          :string "Mann")
 (def-lex-cat Mann-cxn
          :sem-cat (==1 mann person entity
                        (sem-function identifier)))
          :syn-cat (==1 (pos noun)
                        (syn-function nominal))
 (def-feature-matrix Mann-cxn
          :feature (:syn-cat :case-number-gender)
          :paradigm *german-case*
          :dimensions (nom-s-m acc-s-m dat-s-m)))
```

In this state of processing it is not yet possible to unambiguously determine the lexical item's case. It can potentially be nominative, dative or accusative masculine singular. Feature matrices are used to express that there is still ambiguity. (See [44].) To include the respective feature matrix into the `Mann-cxn` the template `def-feature-matrix` is called within `def-lex-cxn`. Within that template, the feature to which the matrix is added to is specified (here: `case-number-gender`). Also the unit-feature to which `case-number-gender` is added to is listed (here: `syn-cat`). Then the kind of paradigm to be used – here `*german-case*` – is indicated. The possible values (`nom-s-m acc-s-m dat-s-m`) are listed as the `Mann-cxn` can be of all three cases, however always being masculine and singular.[13] The German case paradigm has to be created before with a template called `define-paradigm`:

```
(define-paradigm :dimensions ((nom acc dat gen)
                              (s-m s-f s-n)))
```

The resulting paradigm defined by that template is the following:

```
(syn-cat (==1 (case-number-gender
               ((nom ?nom ?nom-s-m ?nom-s-f ?nom-s-n)
                (acc ?acc ?acc-s-m ?acc-s-f ?acc-s-n)
                (dat ?dat ?dat-s-m ?dat-s-f ?dat-s-n)
                (gen ?gen ?gen-s-m ?gen-s-f ?gen-s-n)))))
```

For illustration purposes, plural is ignored in this example. Please refer to [44] for a complete treatment of German determiners. The paradigm that is included in the `Mann-cxn`, then, looks as follows. Some possibilities are already ruled out (i.e. marked with a –), as the gender and number of the lexical item is already known:

[13] Please see [44] for more information on the feature matrices, including a detailed description of the design choices and the respective templates.

```
(syn-cat (==1 (case-number-gender ((nom ?nom-s-m ?nom-s-m - -)
                                   (acc ?acc-s-m ?acc-s-m - -)
                                   (dat ?dat-s-m ?dat-s-m - -)
                                   (gen - - - -)))))
```

In production, the application of the construction is triggered by the meaning (`man indiv-man`) in the `?top-unit`. A new unit is then created, which contains both a semantic and a syntactic pole. On the semantic side, an `args` feature, whose value is the same referent as the one in the meaning predicate, and a `sem-cat` feature, whose value is a simple hierarchy listing the unit's semantic categories `(mann person entity)`, are added. (See Section 4.3.) The syntactic pole contains the part of speech and the syntactic function of the unit (`nominal`). Each construction has a specific syntactic function. The `function` of more complex noun phrases is also `nominal`. Those functions enable the usage of the output unit in higher level constructions. For instance, the determiner-noun-phrase construction does not care if the determiner is combined with a common noun or an adjective-noun phrase. It only imposes one constraint on the unit the determiner is combined with: The function of that unit has to be `nominal`. All lexical constructions are created in the same way as described for the `Mann-cxn`.

5.3 Grammatical Constructions

Grammatical constructions are needed to determine semantic or syntactic roles of the sentence, its phrase structure and the topological sentence structure. The following sections describe the types of constructions needed to produce the utterance.

Phrasal and Intonational Constructions The only phrasal construction to apply three times is the `determiner-nominal-phrase-cxn`. To create this construction the `def-phrasal-cxn` template is used, which heavily resembles the one described in detail in [35] and utilizing the J-operator [9]. That chapter goes step by step through the creation of a phrasal construction by exploring different mechanisms handling each single issue in the construction. The following shows the template as it is used here:

```
(def-phrasal-cxn determiner-nominal-phrase-cxn
    (def-phrasal-skeleton determiner-nominal-phrase-cxn
       :phrase
        (?nominal-phrase
         :cxn-form (== (meets ?determiner-unit ?nominal-unit))
         :sem-function referring
         :phrase-type nominal-phrase)
        :constituents
        ((?determiner-unit
          :sem-function reference
          :syn-function determiner)
         (?nominal-unit
          :sem-function identifier
          :syn-function nominal)))
     (def-phrasal-agreement determiner-nominal-phrase-cxn
       (?nominal-phrase
        :sem-cat (==1 (determined +)
                      (definite ?definiteness)
                      referent)
        :syn-cat (==1 (definite ?definiteness)
                      (determined +)
                      (pos (== ref-expression))
                      (case-number-gender ?case) referent))
       (?determiner-unit
        :sem-cat (==1 (definite ?definiteness))
        :syn-cat (==1 (definite ?definiteness)
                      (pos determiner)
                      (case-number-gender ?case)))
       (?nominal-unit
        :sem-cat (==1 entity)
        :syn-cat (==1 (case-number-gender ?case))))
     (def-phrasal-linking determiner-nominal-phrase-cxn
       (?nominal-phrase
        :args (?referent))
       (?determiner-unit
        :args ?referent)
       (?nominal-unit
        :args (?referent))))
```

The construction created by the template does not add a constructional meaning but only specifies a constructional form (constituent order). However, the choice in semantic or syntactic categories presents the main differences to the construction creation described in [35]. To ensure agreement in case, number and gender, the `case-number-gender` categories of both the determiner and the nominal units carry the same value `?case`.

The rule's application is triggered by the presence of a `?determiner-unit` and a `?nominal-unit` complying with the semantic and syntactic constraints imposed on those units. Each nominal and its associate determiner are made subunits of a newly created noun phrase unit. Depending on which kind of determiner has triggered the application of the `determiner-nominal-phrase-cxn`, the respective definiteness value, `(definite +)` or `(definite -)`, is percolated both to the semantic and syntactic poles of the newly created noun phrase. Similarly, the value of the variable `?case`, which is a complete feature matrix including case, number and gender information, is percolated to the new unit `?nominal-phrase` where the same variable name is used. The part of speech of each newly created unit is `ref-expression`. This attribute is needed for the unit's later use in the ditransitive construction.

One of the created noun phrases is going to be the one that carries the pitch accent, i.e. that is focus-marked. On the form side, this is expressed by adding the marker "FOC" to the noun phrase that was asked for in the previously asked question ("einem Jungen"). Also a syntactic category will be added: (`f-marked +`). On the meaning side, this is represented by the meaning (`f-marking indiv-boy`). The semantic category (`focused +`) is additionally added. The construction which takes care of focus-marking that noun phrase is created with the same `def-phrasal-cxn` template. The only differences are in the choice of semantic and syntactic categories. Most important are those in the `def-phrasal-agreement` template:

```
(def-phrasal-agreement focus-unit-construction
      (?emphasized-phrase
       :sem-cat (==1 (focused +) referent)
       :syn-cat (==1 (f-marked +)
                     (pos (== ref-expression))
                     (case-number-gender ?case))))
```

After successful construction application, the newly created unit `?emphasized-phrase` contains the listed semantic and syntactic categories. They will later be needed to determine the value of the focus constraint in the `?top-unit` to determine the order of the two objects in the Mittelfeld. (See Section 5.3.) Additionally, case, number and gender information of the unit to be emphasized is percolated from the noun phrase to `?emphasized-phrase`.

Argument Structure The main task of argument structure constructions is to link semantic roles of event participants to their grammatical roles [34]. This section briefly examines the ditransitive construction needed to produce the utterance. For a complete chapter on how to deal with argument structure of utterances and how they can be created see [43]. There are several differences between the argument structure construction used here and argument structure constructions presented in that chapter:

1. It does not explicitly add any meaning in form of a meaning predicate, but it assigns semantic and syntactic roles to the event participants.

2. It does not impose constituent order on its subunits. The main reason for that design choice is that it can be used both in questions and declarative sentences, differing in their surface structure. Constituent order is only accounted for by ordering the topological fields. (See Section 5.3.)
3. It creates a new unit-feature `constraint-status` in the top-unit and adds three attribute-value pairs to this unit-feature which are needed to determine the order of the nominal objects in the Mittelfeld:

   ```
   (constraint-status (==1 (focus-constraint ?fc)
                           (definiteness-constraint ?dc)
                           (case-constraint ?cc))
   ```

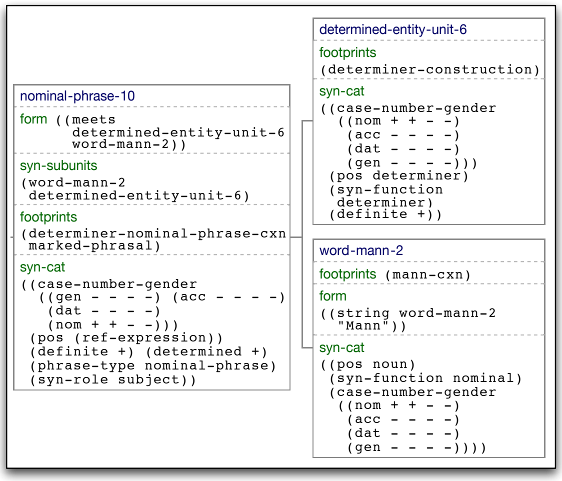
   After the application of the ditransitive construction, the values of those three attributes are still underspecified (marked by the preceding question mark (?) of the symbol name). They become atomic (either + or −) after their truth-value has been checked by the respective constructions (cf. Section 5.3). This is a requirement to decide whether or not a scenario as listed in Table 1 is valid.

The decision which sentence constituent plays which semantic or syntactic role in a sentence has been delayed to a point where it can unambiguously be decided. After application of that construction all event roles are specified, i.e. the mapping of semantic roles like `agent` and syntactic roles like `subject` has been taken care of, which means that the respective cases of the event participants can ultimately be assigned. For instance the feature matrix for the determiner noun phrase "*der Mann*" looks now as follows:

```
(syn-cat (==1 (case-number-gender ((nom + + - -)
                                   (acc - - - -)
                                   (dat - - - -)))))
```

Morphological Constructions Morphological constructions operating solely on the syntactic part of the structure determine the eventual form of the articles. Since the case, number and gender of determined noun phrases has been determined by the argument structure construction, the determiners' case, number and gender is known, as well. Therefore, their form can be assigned at this point in processing. The template to create a morphological construction for the determiner *der* looks like the following:

```
(def-morph-cxn der-morph
    :syn-cat (==1 (pos determiner)
                  (case-number-gender ?case)
                  (definite +)
                  (syn-function determiner))
    :string "der")
```

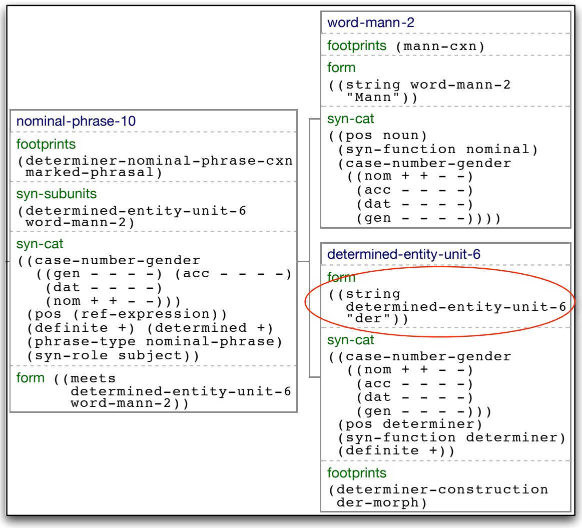


```
nominal-phrase-10
form ((meets
       determined-entity-unit-6
       word-mann-2))
syn-subunits
(word-mann-2
 determined-entity-unit-6)
footprints
(determiner-nominal-phrase-cxn
 marked-phrasal)
syn-cat
((case-number-gender
  ((gen - - - -) (acc - - - -)
   (dat - - - -)
   (nom + + - -)))
 (pos (ref-expression))
 (definite +) (determined +)
 (phrase-type nominal-phrase)
 (syn-role subject))
```

```
determined-entity-unit-6
footprints
(determiner-construction)
syn-cat
((case-number-gender
  ((nom + + - -)
   (acc - - - -)
   (dat - - - -)
   (gen - - - -)))
 (pos determiner)
 (syn-function
  determiner)
 (definite +))
```

```
word-mann-2
footprints (mann-cxn)
form
((string word-mann-2
  "Mann"))
syn-cat
((pos noun)
 (syn-function nominal)
 (case-number-gender
  ((nom + + - -)
   (acc - - - -)
   (dat - - - -)
   (gen - - - -))))
```

Fig. 12. The syntactic part of the noun phrase unit with its two subunits.



```
nominal-phrase-10
footprints
(determiner-nominal-phrase-cxn
 marked-phrasal)
syn-subunits
(determined-entity-unit-6
 word-mann-2)
syn-cat
((case-number-gender
  ((gen - - - -) (acc - - - -)
   (dat - - - -)
   (nom + + - -)))
 (pos (ref-expression))
 (definite +) (determined +)
 (phrase-type nominal-phrase)
 (syn-role subject))
form ((meets
       determined-entity-unit-6
       word-mann-2))
```

```
word-mann-2
footprints (mann-cxn)
form
((string word-mann-2
  "Mann"))
syn-cat
((pos noun)
 (syn-function nominal)
 (case-number-gender
  ((nom + + - -)
   (acc - - - -)
   (dat - - - -)
   (gen - - - -))))
```

```
determined-entity-unit-6
form
((string
  determined-entity-unit-6
  "der"))
syn-cat
((case-number-gender
  ((nom + + - -)
   (acc - - - -)
   (dat - - - -)
   (gen - - - -)))
 (pos determiner)
 (syn-function determiner)
 (definite +))
footprints
(determiner-construction
 der-morph)
```

Fig. 13. The same unit as above after the application of a morphological construction having added the form feature to the determination unit.

Figure 12 shows the syntactic part of the noun phrase unit combining the `Mann-unit` and the `determination-unit` before the application of the morphological construction. After its application the form feature with the string "der" has been added to the `determination-unit`, depicted in Figure 13.

Constraint-checking Constructions Constraint-checking constructions determine which of the three constraints, whose status is monitored in the `top-unit`, are met, regarding a certain order of noun phrases in the Mittelfeld. Depending on the result of the test, the value of each attribute will be either + or −. When the case constraint is met (dative NP *precedes* accusative NP), none of the other constraints no longer have to be checked (see Scenarios 1, 3, 4 and 5 of Table 1), which means their attribute value can stay underspecified. Only in the second scenario, i.e. when marked constituent order is chosen, the values of the definiteness and the focus constraint have to be validated. (See description of the decision process in Section 3.)

To account for the utterance in (19), only the construction checking the case constraint applies. All constructions checking any kind of constraint look very similar and are created with the `def-constraint-check` template. The following creates the `case-check-construction-positive` construction.

```
(def-constraint-check case-check-construction-positive
     :constituent-order (?first-object ?second-object)
     :meaning (unmarked)
     :constraint-status (case-constraint +)
     :constituents ((?first-object
                     :args (?recipient)
                     :sem-cat (==1 (sem-role recipient))
                     :syn-cat (==1 (syn-role (== indirect-object))))
                    (?second-object
                     :args (?patient)
                     :sem-cat (==1 (sem-role patient))
                     :syn-cat (==1 (syn-role (== direct-object))))))
```

On its syntactic side, the construction imposes a constituent order on its two constituents `?first-object` and `?second-object`. This constituent order is specified in the slot `:constituent-order`, then the construction's meaning is given. The following slot allows for the manipulation of the values of the constraints that are in the `top-unit`. Here, it is specified, that the case-constraint is going to be met (it is positive). When applied, the construction assigns a + to the previous variable `?cc` of the case constraint `(constraint-status (==1 (case-constraint +)))`. In the slot `:constituents`, both semantic and syntactic categories of all constituents (here `?first-object` and `?second-object`) are specified. The constituent `?first-object` must have the `(sem-role recipient)` and the `(syn-role (== indirect-object))`. Similarly, it is constrained that the constituent `?second-object` must have the `(sem-role patient)` and the `(syn-role (== direct-object))`. Those constraints and the imposed constituent order on the constituents equal the case constraint: dative NP *precedes* accusative

NP.

The following shows what the created construction looks like. The information which has been provided by the template is printed in bold:

```
(def-cxn case-check-construction-positive ()
         ((?top (sem-subunits (== ?first-object ?second-object))
                (sem-cat (==0 question-to-be-answered))
                 (meaning (== (unmarked)))
                (footprint (==0 case-check-construction)))
          (?first-object
           (sem-cat (==1 (sem-role recipient)))
           (args ?recipient))
          (?second-object
           (sem-cat (==1 (sem-role patient)))
           (args ?patient))
          ((J ?top)
            (constraint-status (==1 (case-constraint +)))
           (footprint (== case-check-construction))))
         <-->
         ((?top (syn-subunits (== ?first-object ?second-object))
                (form (== (meets ?first-object ?second-object)))
                (syn-cat (==0 question-to-be-answered))
                (footprint (==0 case-check-construction)))
          (?first-object
           (syn-cat (==1 (syn-role (== indirect-object)))))
          (?second-object
           (syn-cat (==1 (syn-role (== direct-object)))))
          ((J ?top)
           (constraint-status (==1 (case-constraint +)))
           (footprint (== case-check-construction)))))
```

After it has been determined that the case constraint is positive, the stage is set for the topological constructions to apply.

Topological Constructions To account for sentence structure in German, a field topology approach is followed as briefly described in Section 2. Each field is considered as a box in which constituents are put when they meet several conditions. They all look very similar and apply in exactly the same way as all previously discussed constructions. The following shows the template `def-field-cxn` used to create constructions that build fields:

```
(def-field-cxn  field name
               :constituent-order  list constituents that should follow each other
               :constituents ( ?unit-name
                                :args  arguments
                                :sem-cat  features
                                :syn-cat  features))
```

In the first slot `:constituent-order` the constituents that are supposed to follow each other are listed. The slot `:constituents` can capture a varying number of constituents. Those constituents will be put into the respective field. This is operationalized by creating a new field-unit and making the constituents subunits of that field-unit. A new unit-feature is introduced in the newly created field unit called `field-role`. This unit-feature is needed later by the sentential construction which puts the fields into linear order. The value of this feature is the actual name of the respective field the unit designates. The following subsections show how that template is used to create the linke Klammer, Vorfeld and Mittelfeld.

Linke Klammer The first field to be inspected is the linke Klammer. The following shows the template creating it:

```
(def-field-cxn linke-klammer-full-verb-constituent
     :constituents ((?clause-constituent
                     :args (?referent)
                     :sem-cat (==1 event)
                     :syn-cat (==1 (verb-form finite)
                                   (pos (== full-verb))
                                   (syn-role predicate)))))
```

This field is the simplest one here, as there is only one option which sentence constituent it can capture: the finite verb. Therefore, the semantic category of this clause constituent have to be of type event `(sem-cat (== event))` and the syntactic categories have to include `(syn-cat (==1 (verb-form finite) (pos (== verb)) (syn-role predicate)))`.

After the application of this construction, a new unit is created which captures the constituent and takes it as its subunit. To both its syntactic and semantic pole a new unit-feature is added:

```
(field-role (== linke_klammer))
```

Vorfeld Generally, each sentence constituent except the finite verb are allowed in the Vorfeld. There are, however, exceptions which are discussed in detail in [25]. With an interest in the double object construction in the Mittelfeld, only the subject is permitted in the Vorfeld in this grammar implementation. It is, however, also possible to implement the Vorfeld construction in a way so that it accepts other sentence constituents, but this construction will not be described here. To make sure that the subject will be put into the Vorfeld, the syntactic category of the constituent the construction applies on is required to be of (`syn-role subject`) and the semantic category of (`sem-role agent`). Again – behind the scenes – a `field-role` unit-feature and a footprint are added to the newly created unit.

The following shows the `def-field-cxn` template used to create a construction that builds the Vorfeld:

```
(def-field-cxn Vorfeld-construction
    :constituent-order (?clause-constituent ?left-bracket)
    :constituents ((?clause-constituent
                    :args (?agent)
                    :sem-cat (==1 (sem-role agent))
                    :syn-cat (==1 (syn-role subject)))
                   (?left-bracket
                    :field-role (==1 linke_klammer))))
```

Mittelfeld In our implementation, there are two different constructions that can create a Mittelfeld, however, only one of them can apply here based on the positive value of the case-constraint: `(constraint-status (==1 (case-constraint +)))`.
Below, the `def-field-cxn` template is used to create that Mittelfeld:

```
(def-field-cxn Mittelfeld-construction
    :constraint-status (case-constraint +)
    :word-order (?first-object ?second-object)
    :constituents ((?first-object
                    :sem-cat (==1 (sem-role recipient))
                    :syn-cat (==1 (syn-role (==1 indirect-object))))
                   (?second-object
                    :sem-cat (==1 (sem-role patient))
                    :syn-cat (==1 (syn-role (==1 direct-object))))))
```

In parsing, the `Mittelfeld-construction` applies when two subunits are present fulfilling the syntactic constraints that both of their `syn-roles` have to be objects, one being `(syn-role (== indirect-object))` and the other one `(syn-role (== direct-object))` respectively.

In production, the construction imposes semantic constraints on its constituents, i.e. their `sem-role` has to be either patient or recipient:
`(sem-cat (==1 (sem-role (== patient))))` or
`(sem-cat (==1 (sem-role (== recipient))))` respectively.

As soon as those constraints are met, a Mittelfeld-unit is created, taking the direct object and the indirect object as subunits and imposing constituent order on them `(form (== (meets ?first-object ?second-object)))`. By now, all necessary sentence constituents are captured in a topological field. The created fields are still in no specific order. Sentential constructions put the fields into linear order, how exactly is explained in the following.

Sentential Constructions Sentential constructions, as for instance an interrogative or a declarative construction, put the created fields into linear order. The declarative construction is triggered as soon as there are three fields, here called `?vorfeld, ?linke-klammer` and `?mittelfeld`, which fulfill the following constraints on both semantic and syntactic poles of the source structure:

```
(?vorfeld (field-role (== vorfeld)))
(?linke-klammer (field-role (== linke_klammer)))
(?mittelfeld (field-role (== mittelfeld)))
```

The declarative construction does not add additional conceptual meaning, however it contributes to the meaning by adding a semantic and a syntactic category to the top-unit: `(sem-cat (==1 assertion))` and `(syn-cat (== declarative-clause))`. It additionally does not care about the number and type of constituents captured in the fields. It solely sorts the fields by imposing `meets`-constraints on them in its form feature:

```
(form (== (meets ?vorfeld ?linke-klammer)
          (meets ?linke-klammer ?mittelfeld)))
```

Finally, all constructions which are needed to create the utterance in (19) have applied and the utterance, therefore, can be rendered.

6 Discussion and Conclusion

There are various efforts operationalizing German constituent order which have inspired this work. In [12], for instance, the implementation of a grammar is described, dealing with German constituent order based on a topological model starting from a syntactic dependency tree. The main emphasis of that work is, however, to cover all acceptable linear orders of German declarative sentences. There is no semantic or contextual information integrated in their formalism yet. Furthermore, they do not account for syntactic structure of the sentence at all, playing a fundamental role in constituent order, but only care about the topological phrase structure, i.e. about the order or the fields per se and not the order of elements within those fields or other phrases' structure. The HPSG description of German constituent order in [20] also describes linearization rules exclusively based on the topological structure of sentences.

Information structure is usually ignored in present-day computational systems, however, there are some efforts where its inclusion is regarded beneficial. [39] reports on the incorporation of information structure into UCCG – Unification-based Combinatory Categorial Grammar. The main goal of the involvement of information structure in that grammar implementation is to improve the performance of speech generation systems. Similar to the account taken here, pitch accents are implemented as autonomous units which can be combined with the constituent that has to be emphasized. This ensures that the lexicon is not unnecessarily expanded as it is, for instance, in Combinatory Categorial Grammar [33] where each lexical word is represented multiple times in the lexicon: one separate entry is created for the non-emphasized word and several entries for each possible accent that word can carry. However, that grammar implementation includes a much more fine-grained analysis of intonation as its main purpose is completely different to the grammar presented here, which is to be further used in a speech generating system.

In [11], the integration of information structure into the HPSG framework is described. This study is conform with this approach in several points, such as that information structure should be an integral part of the grammar instead of representing it independently. However, their analysis has never been expressed as a working computational implementation, but only describes a potential computational system. From the point of view taken here, it is considered to be essential to implement the analyses proposed to validate their assumptions.

There are other efforts in HPSG dealing with the implementation of grammars including aspects of information structure. For instance [4] describes an HPSG grammar implementation which takes into account the thematic role of clitic left dislocated arguments in Spanish, assuming that part of the focus of an utterance depends on that thematic role.

Most inspiring to our work is the approach taken in [6] who presents a detailed analysis – only descriptive and not computationally implemented – based on Optimality Theory grammar, accounting for the same linguistic constraints mentioned in Section 2.2. His main focus, the same proposed here, is to present a case study of the interaction of constituent order, prosody and focus. As we did in our study, he is limiting his discussion mainly to the double object construction.

This paper has presented an operational solution illustrating a field topology approach and including one aspect of information structure. The grammar can be used to produce and parse WH-questions and corresponding answers which not only include the most efficient and also context-sensitive focus-marking, but also account for a complete phrase structure of both questions and answers.

Various design choices have been made to achieve the desired grammar. One main aspect includes the tight integrating of focus-marking into the grammar which highlights the non-modular approach to language that FCG is attributed to. This way, the grammar can use information structure to determine constituent order (and vice versa). Besides several other design choices, a novel design pattern of representing the status of constraints in the top-unit of the transient structure has been explored. It has been shown that FCG is open enough to offer an appropriate framework to deal with the mentioned issues. Future work includes the further exploration of the monitoring of constraint status and the integration of further aspects of information structure into the grammar, such as for instance focus/background marking.

Acknowledgements

Research reported in this paper was funded by the Sony Computer Science Laboratory Paris and the EU FP7 ALEAR project. I thank Luc Steels and the whole team working on FCG at the University of Brussels (VUB AI lab) and at Sony CSL for their contributions in making FCG such a superb environment for doing sophisticated experiments in construction grammar. Also I would like to thank Remi van Trijp for his help in bug-fixing the grammar and Stefan Müller and

Philippa Cook for helpful comments on an earlier version of this chapter. All remaining errors are mine.

Bibliography

[1] Behaghel, O.: Beziehung zwischen Umfang und Reihenfolge von Satzgliedern. Indogermanische Forschungen 25, 110–142 (1909)

[2] Behaghel, O.: Von deutscher Wortstellung. Zeitschrift für Deutschkunde 44, 81–89 (1930)

[3] Beuls, K.: Construction sets and unmarked forms: A case study for Hungarian verbal agreement. In: Steels, L. (ed.) Design Patterns in Fluid Construction Grammar. John Benjamins, Amsterdam (2011)

[4] Bildhauer, F.: Clitic left dislocation and focus projection in Spanish. In: Proceedings of the 15th International Conference on Head-Driven Phrase Structure Grammar. pp. 346–357. CSLI publications (2008)

[5] Bleys, J., Stadler, K., De Beule, J.: Search in linguistic processing. In: Steels, L. (ed.) Design Patterns in Fluid Construction Grammar. John Benjamins, Amsterdam (2011)

[6] Büring, D.: Let's phrase it!—Focus, word order, and prosodic phrasing in German double object constructions. In: Müller, G., Sternefeld, W. (eds.) Competition in Syntax, pp. 101–137. No. 49 in Studies in Generative Grammar, de Gruyter, Berlin & New York (2001)

[7] Comrie, B.: Language universals and linguistic typology: Syntax and morphology. University of Chicago Press, Chicago (1981)

[8] Cook, P.: The datives that aren't born equal. beneficiaries and the dative passive. In: Hole, D. (ed.) Datives and other cases. Between argument structure and event structure. pp. 141–184. John Benjamins, Amsterdam (2006)

[9] De Beule, J., Steels, L.: Hierarchy in fluid construction grammar. In: Furbach, U. (ed.) Proceedings of the 28th Annual German Conference on Artificial Intelligence. Lecture Notes in Artificial Intelligence, vol. 3698, pp. 1–15. Springer Verlag, Berlin, Germany (2005)

[10] Drach, E.: Grundgedanken der deutschen Satzlehre. Diesterweg, FRANKFURT (1937)

[11] Engdahl, E., Vallduví, E.: Information packaging in HPSG. In: Edinburgh Working Papers in Cognitive Science. pp. 1–32 (1996)

[12] Gerdes, K., Kahane, S.: Word order in German: a formal dependency grammar using a topological hierarchy. In: ACL '01: Proceedings of the 39th Annual Meeting on Association for Computational Linguistics. pp. 220–227. Association for Computational Linguistics, Morristown, NJ, USA (2001)

[13] Grice, M., Baumann, S., Benzmüller, R.: German intonation in autosegmental phonology. In: Prosodic typology (2003)

[14] Halliday, M., Hasan, R.: Notes on transitivity and theme in English. Journal of Linguistics 3, 199–244 (1967)

[15] Hawkins, J.: A performance theory of order and constituency. Cambridge University Press, Cambridge (1994)

[16] Hoberg, U.: Die Wortstellung in der geschriebenen deutschen Gegenwartssprache. Heutiges Deutsch. Linguistische Grundlagen. Forschungen des Instituts für deutsche Sprache, No. 10, Max Hueber Verlag, München (1981)

[17] Höhle, T.N.: Explikation für "normale Wortstellung". In: Abraham, W. (ed.) Satzglieder im Deutschen - Vorschläge zur syntaktischen, semantischen und pragmatischen Fundierung. pp. 75–153. Studien zur deutschen Grammatik, Gunter Narr Verlag, Tübingen (1982)

[18] Höhle, T.N.: Topologische Felder. ms, Köln (1982)

[19] Höhle, T.N.: Der Begriff Mittelfeld, Anmerkungen über die Theorie der topologischen Felder. In: Weiss, W., Wiegand, H.E., Reis, M. (eds.) Textlinguistik contra Stilistik? Wortschatz und Wörterbuch - Grammatische oder pragmatische Organisation von Rede? pp. 329 – 340. Kontroversen, alte und neue, Max Niemeyer Verlag, Tübingen (1986)

[20] Kathol, A.: Linearization-based German Syntax. Ph.D. thesis, Ohio State University, Berkeley (1995)

[21] Kempen, G., Harbusch, K.: A corpus study into word order variation in German subordinate clauses: Animacy affects linearization independently of grammatical function assignment. In: Pechman, T., Habel, C. (eds.) Multidisciplinary approaches to language production, pp. 173–181. Mouton de Gruyter, Berlin (2004)

[22] Kurz, D.: A statistical account on word order variation in german. In: Proceedings of the COLING Workshop on Linguistically Interpreted Corpora (2000)

[23] Lambrecht, K.: Information Structure and Sentence Form: Topic, Focus, and the Mental Representations of Discourse Referents (Cambridge Studies in Linguistics). Cambridge University Press (1996)

[24] Lenerz, J.: Zur Abfolge nominaler Satzglieder im Deutschen. Narr, Tübingen (1977)

[25] Müller, S.: Deutsche Syntax deklarativ. Head-Driven Phrase Structure Grammar für das Deutsche. No. 394 in Linguistische Arbeiten, Max Niemeyer Verlag, Tübingen (1999)

[26] Müller, S.: Multiple frontings in German. In: Jäger, G., Monachesi, P., Penn, G., Winter, S. (eds.) Proceedings of Formal Grammar 2002. pp. 113–124. Trento (2002), http://hpsg.fu-berlin.de/~stefan/Pub/mehr-vf.html

[27] Müller, S.: Zur Analyse der scheinbar mehrfachen Vorfeldbesetzung. Linguistische Berichte 203, 297–330 (2005), http://hpsg.fu-berlin.de/~stefan/Pub/mehr-vf-lb.html

[28] Müller, S.: Elliptical constructions, multiple frontings, and surface-based syntax. In: Jäger, G., Monachesi, P., Penn, G., Winter, S. (eds.) Proceedings of Formal Grammar 2004, Nancy. CSLI Publications, Stanford (to appear)

[29] Reis, M.: On justifying topological frames: Positional field and the order of nonverbal constituents in German. Documentation et Recherche en Linguistique Allemande Contemporaine 22/23, 59 – 85 (1980)

[30] Rooth, M.: A theory of focus interpretation. Ph.D. thesis, University of Massachusetts, Amherst (1985)

[31] Selkirk, E.: Sentence prosody: Intonation, stress, and phrasing. In: Goldsmith, J. (ed.) Handbook of Phonological Theory, pp. 550–569. Blackwell (1995)

[32] Stechow, A.V., Uhmann, S.: Some remarks on focus projection. In: Topic, Focus, and Configurationality. pp. 295–320 (1986)

[33] Steedman, M.: The Syntactic Process. MIT Press, Cambridge, MA, USA (2000)

[34] Steels, L., De Beule, J., Neubauer, N.: Linking in Fluid Construction Grammars. In: Proceedings of BNAIC. pp. 11–18. Transactions of the Belgian Royal Society of Arts and Sciences, Brussels (2005)

[35] Steels, L.: A design pattern for phrasal constructions. In: Steels, L. (ed.) Design Patterns in Fluid Construction Grammar. John Benjamins, Amsterdam (2011)

[36] Steels, L. (ed.): Design Patterns in Fluid Construction Grammar. John Benjamins, Amsterdam (2011)

[37] Steels, L.: A first encounter with Fluid Construction Grammar. In: Steels, L. (ed.) Design Patterns in Fluid Construction Grammar. John Benjamins, Amsterdam (2011)

[38] Steels, L. (ed.): Computational Issues in Fluid Construction Grammar. Springer Verlag, Berlin (2012)

[39] Traat, M.: Information structure in a formal framework. In: CICLing. pp. 28–40 (2009)

[40] Twain, M.: A Tramp Abroad. Oxford University Press Inc (1880)

[41] Uszkoreit, H.: Word Order and Constituent Structure in German. CSLI Publications, Stanford CA (1987)

[42] Vallduví, E.: The dynamics of information packaging. In: Engdahl, E. (ed.) Information Structure into Constraint-Based Categorial Approaches. HCRC Publications, University of Edinburgh, Edinburgh (1994)

[43] van Trijp, R.: A design pattern for argument structure constructions. In: Steels, L. (ed.) Design Patterns in Fluid Construction Grammar. John Benjamins, Amsterdam (2011)

[44] van Trijp, R.: Feature matrices and agreement: A case study for German case. In: Steels, L. (ed.) Design Patterns in Fluid Construction Grammar. John Benjamins, Amsterdam (2011)

[45] Van Valin, R.D., LaPolla, R.J.: Syntax: Structure, Meaning, and Function. Cambridge University Press, Cambridge (1998)

[46] Wegener, H.: Der Dativ im heutigen Deutsch. Studien zur deutschen Grammatik No. 28, originally Gunter Narr Verlag now Stauffenburg Verlag, Tübingen (1985)

[47] Wellens, P.: Organizing constructions in networks. In: Steels, L. (ed.) Design Patterns in Fluid Construction Grammar. John Benjamins, Amsterdam (2011)