3.1 Allgemeine Angaben zum Teilprojekt B4

3.11 Constraints on grammar for efficient generation

3.111 Fachgebiete: Computerlinguistik, Syntax, Semantik

3.12 Leiter:

Prof. Dr. Erhard Hinrichs
Seminar für Natürlich-Sprachliche Systeme
Universität Tübingen
Biesingerstr. 10
W-7400 Tübingen
07071/29-5446

Dr. Dale Gerdemann
Seminar für Natürlich-Sprachliche Systeme
Universität Tübingen
Biesingerstr. 10
W-7400 Tübingen
07071/29-4279

3.13 Bisherige und beantragte Förderung des Teilprojekts (Ergänzungsausstattung)

Haushalts–jahr	Personalkosten	Sächliche Verwaltungs–ausgaben	Investitionen	Gesamt
	100	101	102	103
bis 1988 1989 1990 1991				
Zwischen–summe				
1992	240,4	14,6	53,2	308,2
1993	240,4	14,6		255
1994	240,4	14,6		255

3.2 Zusammenfassung

The goal of this project is to develop the constraints imposed on grammatical rules in order to allow efficient generation using head-driven generation techniques. While there is a growing consensus among computational linguists that the head-driven approach is the most promising approach (Shieber 1988, van Noord 1989, Shieber et al. 1990, Dymetman et al. 1990), there are a number of problematic constructions in which the head is either missing or displaced to another part of the sentence. There are two possible solutions to this problem. Either the generator could be modified to handle such, constructions or the grammar could be constrained so that these structures are non-problematic. This project will investigate the latter approach. The project will focus primarily on problems arising in German syntax. Since empty heads occur most frequently in the verb second construction, a primary goal will be to find an analysis of such constructions which are compatible with efficient generation algorithms.

Other constructions which are problematic for head-driven generation include VP ellipsis (assuming that VP is the head of S), supposed raising-to-object structures, idiomatic structures with non-compositional semantics, and as we have argued previously (Gerdemann & Hinrichs 1990), non-minimally type raised arguments in categorial unification grammar. In this project, we will inventory such structures and investigate what constraints need to be placed on the analysis of such structures for the purpose of efficient generation. We will also investigate the question of whether the optimal grammar for generation is also optimal for parsing.

3.3 Stand der Forschung

The idea of having a declaratively represented grammar which is neutral between parsing and generation originated in the Functional Unification Grammar framework of Kay (1979). A number of advantages of having such a grammar are discussed in Appelt (1987). The primary advantage is that having a single knowledge source ensures consistency between the two modes of processing. The grammar writer can write a single grammar, which will account for both the sentences which can be parsed and the sentences which can be generated. This differs significantly from the older Augmented Transition Network (ATN) approach of Woods (1970), in which a completely different ATN would have to be written for each mode of processing.

Despite the theoretical advantage of unification based grammars, however, until recently, most work was done on parsing and very little attention has been paid to generation with such grammars. Some parsers for unification grammars, such as the interpreter for definite clause grammars in Pereira & Warren (1983), are reversible, though generally parsing is much more efficient than generation.

One major difference between parsing and generation is that in parsing the most obvious strategy is to parse from left to right (or right to left), whereas in generation there is no reason for processing to be at all constrained by word order. In fact, the most widely adopted strategy is to follow a *head-driven* approach as in Shieber (1988) and Shieber et al. (1990). The *functor-driven* approach in Gerdemann & Hinrichs (1990)) is a variant of this strategy. The reason for adopting a head-driven approach is straightforward. If the head contains information as to required complements, then this information will not be available until after the head is generated. This basic idea of the head driven approach appears to be valid, though the problems with the approach which will the major focus of this project.

A second major difference between parsing and generation is difference in the flow of information. Whereas parsing establishes a mapping from a surface string to a logical form, generation establishes a mapping from logical form to surface form (see fig. ??). Logical forms, in this case, are represented as DAGs.

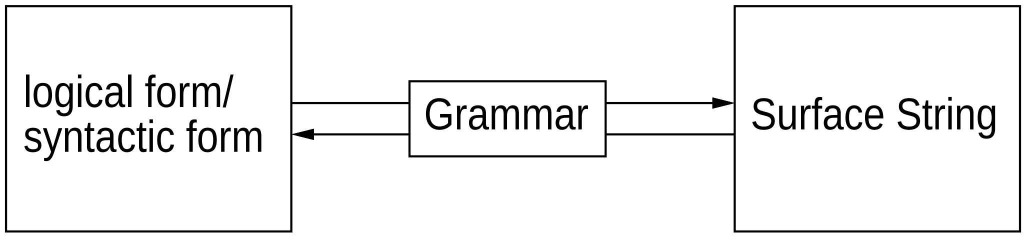

Figure 1: The relationship between parsing and generation

The logical form that is used to generate a sentence may be referred to as the *goal semantics*. It is also possible that generation might begin some functional structure as well as the purely semantic structure. This is particularly the case in the context of machine translation (Wedekind 1988) though, in general, at least a semantic structure should be a required input to generation.

Given that in this view of generation a great deal of information is contained in the initial start category, the most obvious strategy to use this information would be a top-down strategy. This strategy is, in fact, adopted by a number of researchers (Wedekind 1988, Dymetman & Isabelle 1989, Dymetman et al. 1990, and Strzalkowski 1990a,1990b). The problem with the top-down approach, however, is that recursive categories can cause the generator to go into a loop. Most of the top-down approaches avoid such recursive loops by means of node-ordering solutions. However, as Shieber et al. (1990) point out, such node ordering solutions will fail in a head-driven algorithm if the head itself is the recursive category. The Dymetman et al. (1990) approach, however, is exceptional since it does not rely on a node ordering solution, but rather relies on a precompilation to an extended Greibach Normal Form.

An alternative to the top-down algorithms is to use a bottom-up algorithm with top-down guidance. Shieber (1988) proposed the use of Earley's algorithm (Earley 1970), though his approach did not allow the prediction step in this algorithm to provide sufficient top-down guidance. This problem is corrected in Gerdemann (1991). Another approach is use a left-corner style parser as in Shieber et al. 1990, in which top down information is used to first predict the (normally lexical) head of a phrase and then the rest of the category is built bottom up from this head. While ther bottom-up generators avoid the problem of recursive loops, the control problem for such generators is very much the same as for the top-down generators. If the nodes are generated in the wrong order, then there will not be enough top-down information available to guide the bottom-up generation process. Thus the head-driven strategy is equally important in both approaches to generation.

3.4 Einige Vorarbeiten

In Gerdemann & Hinrichs (1988) a graph unification based parser (UNICORN) is described, which has been extended to handle generation in Gerdemann (1991). The generator uses a head-driven version of Earley's algorithm, which is modified to reduce dependence on string position indexing. The parser and generator are both implemented in Prolog.

In Gerdemann & Hinrichs (1990), we showed that the chain rule condition of Shieber et al. (1990) is not sufficient for efficient generation. In this work we developed a grammar in a version of categorial unification grammar, which contains nothing but chain rules, but which still will cause a head-driven generator to go into an unconstrained search. The problematic cases for this grammar arise when the lexicon contains type-raised **NP**s. The solution we proposed for this problem was to adopt the principles of *minimal type assingment* and *type raising only as needed.* These two principles were originally motivated on independent semantic grounds by Partee & Rooth (1983). These principles were incorporated into the grammar using the idea of supercombinators from Wittenburg (1987) and Wall & Wittenburg (1989).

The structure of complex verbs in German is studied in Hinrichs & Nakazawa (1989a) and Hinrichs & Nakazawa (1991). In these two papers it is noted that German verbal complex such as *wird lesen* in (**??**) constitute an exception to the head feature principle of Pollard & Sag (1987) since tense and aspect information is inherited from *wird* and subcategorization information is inherited from is inherited from *lesen*.

(1) daß Peter das Buch lesen wird

The proposed analysis takes *wird* to be a head which subcategorizes first for a verb and then inherits the rest of its subcategorization list from this verb. This analysis in effect encodes, in the framework of HPSG, a lexicalized version of functional composition familiar from extended categorial grammar as in Steedman (1985).[1] The existence of this more complex kind of head will need to be considered in any head-driven generation algorithm.

List of relevant previous publications

Gerdemann, D. (1991) *Parsing and Generation of Unification Grammars.* Ph.D. thesis, University of Illinois. Published as Beckman Institute Cognitive Science Technical Report CS-91-06.

Gerdemann, D. and Hinrichs, E. (1988). UNICORN: A unification parser for attribute-value grammars. *Studies in the Linguistic Sciences*, 18(2):41–86.

Gerdemann, D. and Hinrichs, E. W. (1990). Functor-driven natural language generation with categorial-unification grammar. In *COLING-90 Vol. 2*, 145–150.

[1]This function composition type of analysis has since been adopted by Kiss (1991) and applied to problems of control.

Hinrichs, E. and Nakazawa, T (1989a). Flipped out aux in German. In *Papers from the 25th Regional Meeting of the Chicago Linguistic Society*, 193–202.

Hinrichs, E. and Nakazawa T. (1989b). Review of *Word Order and Constituent Structure in German* by Hans Uszkoreit. *Language*, 65:141–149.

Hinrichs, E. and Nakazawa, T (1991). Subcategorization and VP structure. forthcoming *Technical report, Seminar für Natürliche-Sprachliche Systeme.*

3.5 Ziele, Methoden, Arbeitsprogramm und Zeitplan

3.5.1 Ziele und Methoden

The primary goal of this project is to investigate the problem of efficient generation using unification grammars. As the recent literature has shown there are two sides to this question. First, there is the question of the algorithms to be used and second is the question of what grammars can be used with these algorithms. We have chosen to focus on the second question. Currently, the head driven approach appears to be the most promising approach for the algorithm, so grammars to be used for generation must be consistent with this strategy.

A major goal of the project will be to investigate the problem that arises for head driven generation when heads are displaced as in German verb-second or topicalization constructions (Shieber et al. 1990, Russell et al. 1990) or missing as in the case of ellipsis (Nerbonne et al. 1989). The head driven approach, in its simplest formulation, would not be able to handle such cases. The head-driven approach may also be problematic when the relationship between the head and its sisters is not the prototypical head-complement relationship. One such case arises when the head combines with optional arguments such as adjuncts. In this case, generating the head first may not help to determine which adjuncts need to be generated. Another problem for the head-driven approach may arise in the case of displaced complements. If, for example, a grammar were to allow raising to object structures of the form $[_{\mathrm{VP}}$ V NP VP], then the generator would have to generate the **V** first and then it would have the option of generating either the **NP** or the **VP** next. Clearly, however, the information as to which **NP** to generate will be contained in the **V**.Another problematic case arises in generating with Categorial-Unification grammars (Gerdemann & Hinrichs 1990). In this case, if the "head" is actually a type raised argument, then it will not contain sufficient information to guide the generation process. Given the variety of constructions which appear to be problematic for head driven-generation, we will take as one of our goals simply to catalog such constructions. In describing the problem here, however, we will concentrate on the verb-second construction.

Specifically we will investigate the problem of empty heads is seen in the sentence in (**??**). In this sentence, the verb, which would occur in final position in a subordinate clause is displaced into second position in a main clause. Different

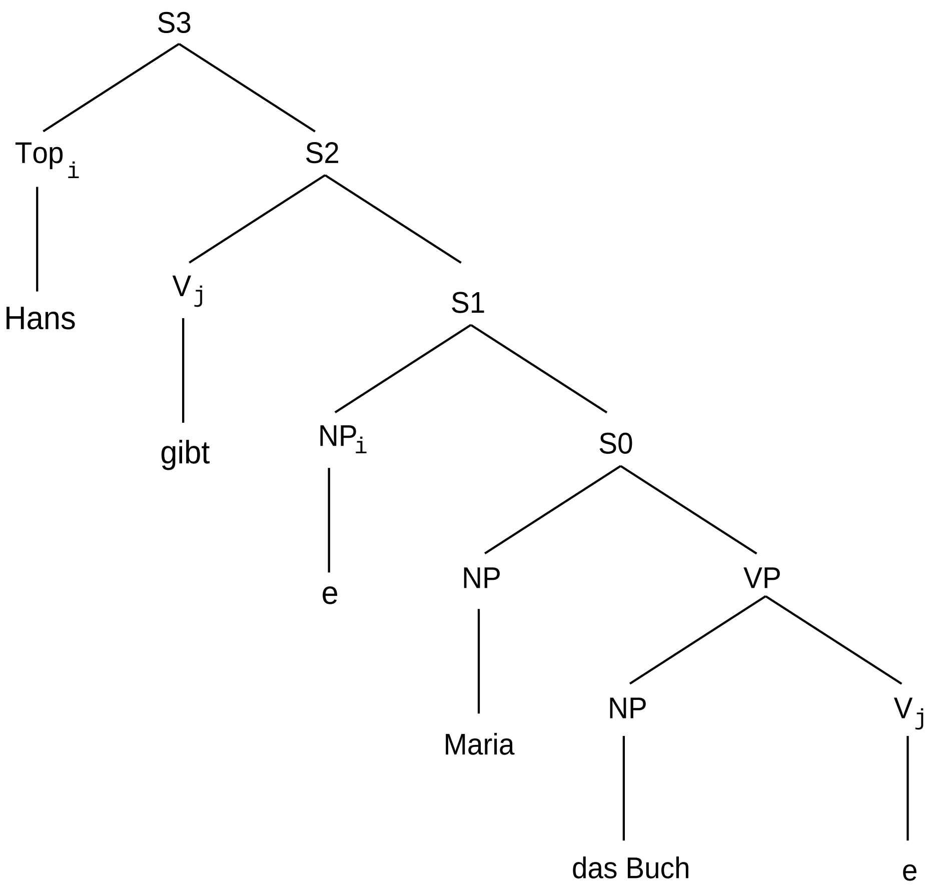

Figure 2: An empty head in German main clause

linguistic theories will differ on the details of the analysis, however there is general agreement that such a dependency exists (Uszkoreit 1987, Hinrichs & Nakazawa 1989b).

For a head driven strategy, however, this displacement can cause serious problems. Assuming that the verbal trace is the lexical head of the sentence, then the first lexical item that the generator will try to generate is this verb in final position. However, since the verb turns out to be gapped, only a trace is generated at this point. But the generator needs to find the verb at this point in order to know what complements to generate.

In order to clarify this problem further, it will be useful to consider a grammar which will account for sentences such as that in fig. ??. The grammar we present here is in a simplified version of HPSG, though the same problem can easily be illustrated within other linguistic frameworks. The lexical entry for a verbal trace in this grammar is shown in (??), whereas the lexical entry for the non-null verb *gibt* is show in (??).

(2)
$$\begin{bmatrix} \text{head:} & [1]\begin{bmatrix}\text{maj:} & \text{vp}\end{bmatrix} \\ \text{bind:} & \begin{bmatrix} \text{v2-slash:} & \begin{bmatrix}\text{head:} & [1] \\ \text{subcat:} & [2]\end{bmatrix} \\ \text{top-slash:} & \text{none} \end{bmatrix} \\ \text{subcat:} & [2] \end{bmatrix}$$

(3) $$\begin{bmatrix} \text{head:} & \begin{bmatrix} \text{maj:} & \text{vp} \\ \text{sem:} & \text{geben([1],[2],[3])} \end{bmatrix} \\ \text{bind:} & \begin{bmatrix} \text{v2-slash:} & \text{none} \\ \text{top-slash:} & \text{none} \end{bmatrix} \\ \text{subcat:} & \left\langle \begin{bmatrix} \text{cat:} & \text{np} \\ \text{sem:} & [3] \end{bmatrix}, \begin{bmatrix} \text{cat:} & \text{np} \\ \text{sem:} & [2] \end{bmatrix}, \begin{bmatrix} \text{cat:} & \text{np} \\ \text{sem:} & [1] \end{bmatrix} \right\rangle \end{bmatrix}$$

In these lexical entries, the *v2-slash* attribute may take as its value the verbal category that appears in second position and the *top-slash* attribute may take as its value a category to be topicalized. The *none* value for these attributes is simply an arbitrary atomic value, which indicates that a complex value is not allowed. The primary difference between these two lexical entries is that whereas *gibt* has a fully instantiated subcategorization list, the trace simply indicates that its *subcat* value is the same as the *subcat* value in the category value of the *v2-slash* attribute. The idea that this lexical entry for trace represents is that traces do not have inherent properties such as subcategorization, but rather inherit these values from elsewhere. Although the lexical entry in (**??**) is cast in the framework of HPSG, any other linguistic theory would also need a way of expressing the fact that traces have inherited properties.

The grammar should allow categories to be cancelled off of the subcategorization list in either of two way. The rule in (**??**) allows a **VP** to combine with a category that it is subcategorizing for and the rule in (**??**) allows a category to be moved from the subcategorization list to *top-slash*.[2]

(4) $$\begin{bmatrix} \text{head:} & [1] \\ \text{bind:} & [2] \\ \text{subcat:} & [4] \\ \text{dtrs:} & \begin{bmatrix} \text{head-dtr:} & \begin{bmatrix} \text{head:} & [1] \\ \text{bind:} & [2] \\ \text{subcat:} & \begin{bmatrix} \text{first:} & [3] \\ \text{rest:} & [4] \end{bmatrix} \end{bmatrix} \\ \text{comp-dtrs:} & \langle\ [3]\ \rangle \end{bmatrix} \end{bmatrix}$$

(5) $$\begin{bmatrix} \text{head:} & [1] \\ \text{bind:} & \begin{bmatrix} \text{v2-slash:} & [2] \\ \text{top-slash:} & [3] \end{bmatrix} \\ \text{subcat:} & [4] \\ \text{dtrs:} & \begin{bmatrix} \text{head-dtr:} & \begin{bmatrix} \text{head:} & [1] \\ \text{bind:} & \begin{bmatrix} \text{v2-slash:} & [2] \\ \text{top-slash:} & \text{none} \end{bmatrix} \\ \text{subcat:} & \begin{bmatrix} \text{first:} & [3] \\ \text{rest:} & [4] \end{bmatrix} \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

Finally, the sentence rule is shown in (**??**). For simplicity, we have assumed

[2]The Head Feature and Binding Inheritance principles are, in effect, incorporated into these rules. Since there are so few rules in HPSG, it makes little computational sense to try to extract out general principles applying across all rules.

flat structure at the top even though most linguistic analyses assume the branching structure shown in fig. ??.

(6)
$$\begin{bmatrix} \text{head:} & [1] \\ \text{dtrs:} & \begin{bmatrix} \text{head-dtr:} & \begin{bmatrix} \text{head:} & [1] \\ \text{bind:} & \begin{bmatrix} \text{v2-slash:} & [2]\ {}_{\textit{complex}}[\,] \\ \text{top-slash:} & [3]\ {}_{\textit{complex}}[\,] \end{bmatrix} \\ \text{subcat:} & \text{none} \end{bmatrix} \\ \text{comp-dtrs:} & \langle\ [2], [3]\ \rangle \end{bmatrix} \end{bmatrix}$$

The most important thing to notice about this rule is that the head daughter is designated to have complex values for both of its *slash* features. As a sentence is generated, these values will percolate downward. The rule in (??) will allow the value for *top-slash* to become *none*, though the complex value for *v2-slash* will percolate all the way down to the lexical head. Thus, given the start DAG [head|sem|geben(hans,marie,buch)], when the generator gets to the point of generating a lexical item, this start DAG will have become instantiated to the DAG in (??).

(7)
$$\begin{bmatrix} \text{head:} & \begin{bmatrix} \text{maj} & \text{vp} \\ \text{sem:} & \text{geben(hans,marie,buch)} \end{bmatrix} \\ \text{bind:} & \begin{bmatrix} \text{v2-slash:} & {}_{\textit{complex}}[\,] \\ \text{top-slash:} & \text{none} \end{bmatrix} \end{bmatrix}$$

The goal DAG at this point, of course, contains all of the needed semantics but still it will not unify with the lexical entry for *gibt* since the complex value for *v2-slash* will not unify with the atomic value *none* in the lexical entry for *gibt*. So at this point it is still not possible to to generate this lexical item with its subcategorization list.

This complex value for *v2-slash* will, however, unify with the complex value for *v2-slash* in the lexical entry for the trace in (??). Thus, only the trace can be generated as the lexical head. Once the lexical head is generated, the subcategorization list of this head needs to be used to build the rest of the phrase bottom-up. But the subcategorization list of the trace in (??) is completely uninstantiated, so the generator will, at this point, have no constraints at all that can be used to build the rest of the phrase. The semantics of the phrase is available, as shown in (??), but it is not known how this semantics corresponds to a subcategorization list.

Our goal, then, in this project will be to evaluate various possible strategies that could be used to overcome this problem. Although we are aware of the shortcomings of certain approaches, we do not know at this point which approach will ultimately be preferable.

One strategy to avoid this problem was proposed by Shieber et al. (1990). These authors propose that rules such as (??) which introduce gap filler constructions should be given some kind of mark or flag by the grammar writer.

Then when the generator encounters such a flagged rule, the head-driven strategy can be temporarily abandoned in order to generate the filler first. In the grammar given here, the head features of the trace (including the semantic value) are identical to the head features of the verb in second position. Thus, the goal semantics could be used to immediately generate the second position verb before generating the head daughter. If this is done, then when the trace is generated as the lexical head of the head daughter, it will have an instantiate subcategorization list.

A general problem with this idea is that it reduces the modularity between the grammar and the processor. In this case, the grammar writer needs to be aware of processing concerns in order to properly annotate the rules.

A more specific problem with this approach is that it will fail when the same rule sometimes results in the displacement of a head and sometimes results in the displacement of a non-head. Topicalization appears to be an example of such a rule. In (**??**), a lexical non-head *Hans* is topicalized, however in (**??**) the lexical head is topicalized.[3] And similarly, in (**??**), a phrasal head (i.e. the **VP** *Das Buch lesen*)is topicalized.

(8) Lesen wird Peter das Buch.

(9) Das Buch lesen wird Peter.

Given the existence of rules of this sort, it will not be possible for the grammar writer to determine once and for all whether or not the topic constituent needs to be generated first. Whether or not the topic needs to be generated first depends on how the topic is instantiated in a particular application of the rule.

Another possibility would be to adopt the position that head movement can only move a head into another head position. This is, in fact, the analysis favored in Government Binding theory (Baker 1988). Thus, in fig. **??**, if *gibt* is in an **INFL** position which is the head of **S2** and **S3**, then it would be quite unexceptional for it to be generated first. This is clearly the kind of analysis that we would prefer, since in this case, constraints which are apparently linguistically motivated conform very nicely to the constraints imposed by efficient generation. However, this analysis again runs into the problem of topicalization. If **TOP** is considered to be the head in fig. **??**, then the generator will attempt to generate *Hans* first. But, of course, there will not be enough top-down information to generate *Hans* until the *lesen* is generated with its subcategorization list. On the other hand, if **TOP** is not considered to be the head, then the reverse problem arises in generating sentences like (**??**) and (**??**). Still, the idea of constraining head movement is appealing and deserves more investigation, even though at present it is not clear how such a constraint might help in generation.

Another strategy for dealing with dependencies is suggested in Gerdemann (1991). The strategy suggested there was to treat all dependencies as being strictly phonological. Given the existence of a *phon* feature as in HPSG, it would

[3] *Lesen* is the head here in the sense that it contains the relevant subcategorization list (Nakazawa & Hinrichs 1991).

be possible for the phonology of a constituent to pass up a tree, say as the value of a *slash* type of feature. Then it would be possible for a phrase to "actually" occur in a lower part of a tree, while its phonology occurs somewhere higher in the tree. In this analysis, a phonological dependency could be introduced by a rule such as that in (**??**).

(10)
$$\left[\begin{array}{ll} \text{phon:} & \langle\,\rangle \\ \text{head:} & [1] \\ \text{bind:} & \left[\begin{array}{ll}\text{phon-slash:} & [2]\end{array}\right] \\ \text{dtrs:} & \left[\begin{array}{ll} \text{head-dtr:} & \left[\begin{array}{ll}\text{phon:} & [2] \\ \text{head:} & [1] \\ \text{bind:} & \text{none}\end{array}\right]\end{array}\right]\end{array}\right]$$

This approach, then, would allow for certain mismatches between syntax and phonology. The idea is reminiscent of Hendrix's (1987) idea of having different algebras apply in the syntax and semantics to account for quantifier scopings that don't match surface word order. More generally, the idea is similar to the idea of having multiple levels of representation is Government Binding theory or Lexical Functional Grammar. It would be interesting to compare these various approaches to see how they relate.

One problem with this approach, though, is that with such phonological dependencies, it may no longer be possible to parse in a strictly left-to-right fashion. Thus it may turn out that the optimal grammar for generation is no longer optimal for parsing. There appear to be two possible solutions to this problem. One idea would be to develop new parsing algorithms which could deal with such dependencies. Or a second idea would be to use a grammar compiler to produce separate grammars for parsing and generation.

One other idea for handling dependencies in general is suggested by Steedman (1985). The basic idea in this approach is that type changing rules and rules of composition will allow dependencies to be expressed without actual use of an empty category. Thus, for example, the sentence *John, I like* could be analyzed as in fig. **??**. Once the subject **NP** is type raised in this analysis, then *I like* may form the constituent **S/NP** by function composition without the use of any kind of trace. Then when *John* is type raised by a special harmonic version of type raising, it will be possible for the resulting category to combine with the **S/NP** constituent to form a sentence. In this case, then, the empty-head problem is eliminated.

This analysis, however, is not unproblematic. Moortgat (1988) has shown that unconstrained harmonic type raising significantly increases the power of the Lambek system for categorial grammar. Even so, harmonic type raising is only sufficient to handle extractions from peripheral positions such as that in fig. **??**. Moortgat shows that in order to handle cases of non-peripheral extraction in sentences such as (**??**), it is necessary to have additional harmonic rules, such as harmonic function composition.

(11) The book, I put on the table

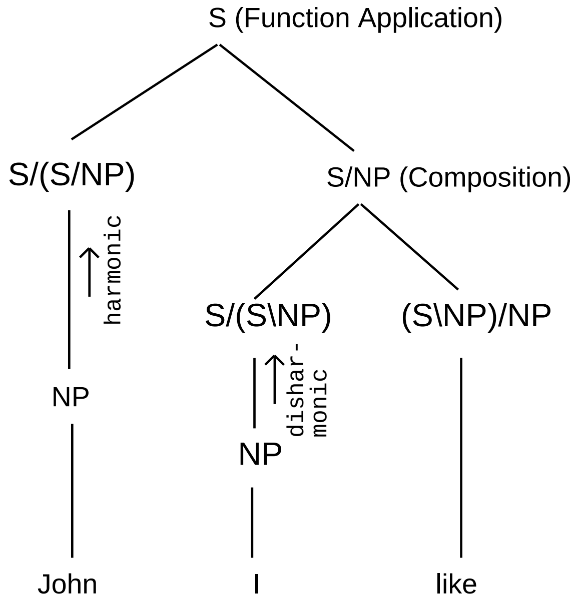

Figure 3: Categorial grammar treatment of dependencies

Given the complications of this system, Bouma (1987) and Pollard (1988) have argued that it would be much simpler to extend categorial grammar (or categorial unification grammar) to include a slash feature as in GPSG. Moortgat (1988), as well, uses a device very similar to a slash feature (i.e. a *gap operator*), though Moortgat's operator is more fully integrated into the Lambek system.

Even if Steedman's approach of eliminating gaps can be made to work, the resulting grammar may still be problematic for head-driven generation approaches. Some work on generation with categorial grammars was reported in Gerdemann & Hinrichs (1990), though the work presented there would need to be extended in order to handle more complex rules such as composition. A reasonable approach to explore would be to generate primary functors (i.e. a functor which determines the ultimate resulting category) before subordinate functors, though the details of this proposal need to be investigated thoroughly.

Finally, whatever analysis one adopts for head dependencies must allow for more complex cases such as that shown in fig. **??**. In this case, a gap occurs inside of a filler. It's quite likely that straight forward analyses of filler gap dependencies will not be extendible to handle such more complex cases.

At this point, we do not know what solution will ultimately be preferrable. We conjecture that the problems involved in efficient generation can only be fully understood if the range of data is significantly expanded. Although we have concentrated here on the problem of verb second constructions in German, it may ultimately turn out that other phenomena are even more problematic. In this regard, cases of **VP** ellipsis (Sag, 1977), empty head nouns (Nerbonne et al. 1989) or the possibility of empty prepositions in phrases such as *next Tuesday* (Larson 1985, McCawley 1988) constitute related phenomena that will require close attention. At the same time, however, any theory of head-driven generation must incorporate a linguistically adequate account of headedness.

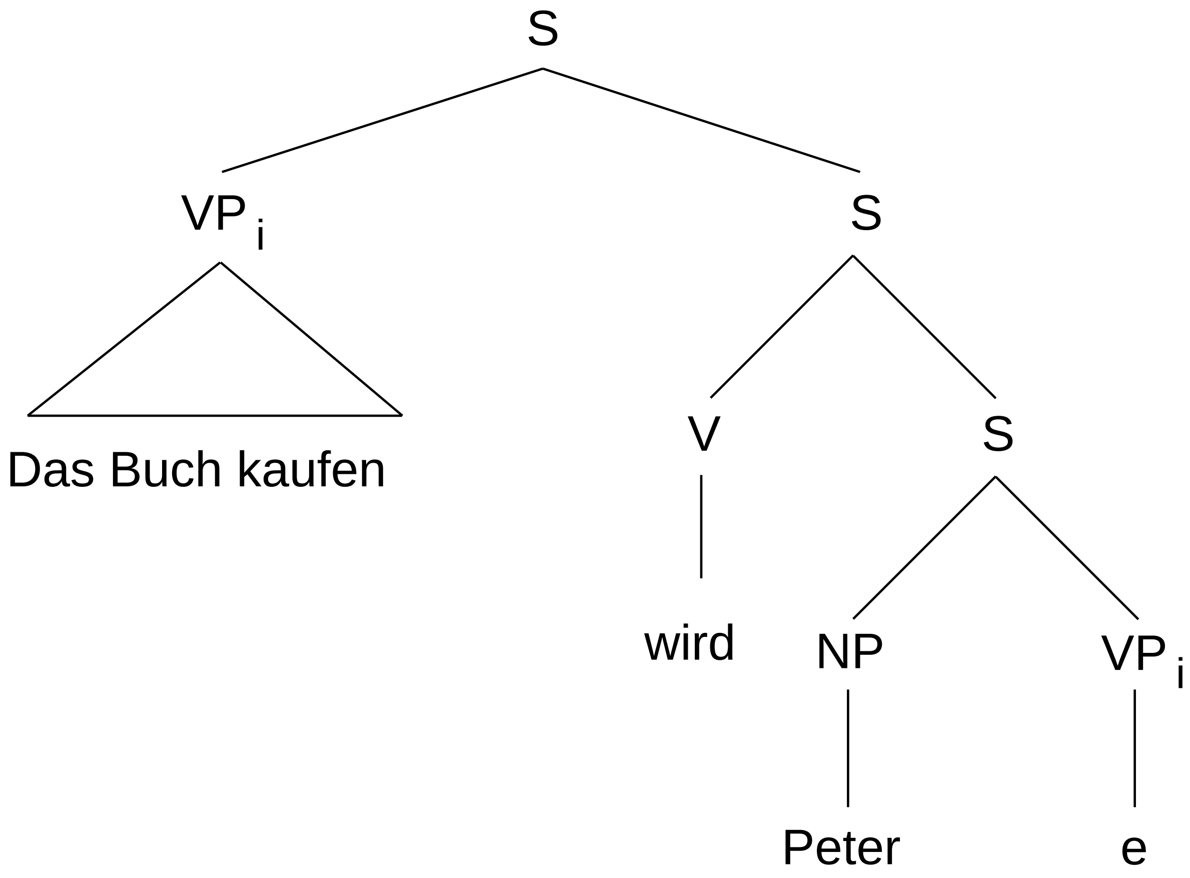

Figure 4: An example of a gap within a filler

Therefore, linguistic studies of headedness such as Zwicky (1985) and studies on X-bar need to be investigated for their impications for head-driven generation.

3.5.1 Zeitplan

1991 I. Outline of which constructions are problematic for head-driven generation

II. Exploration of the range of possible analyses for such constructions

1992 I. Determination of which analyses are conducive to efficient head-driven generation

II. Determination of the linguistic adequacy of analyses that are conducive to head-driven generation

III. Exploration of the consequences of different linguistic conceptions of *head* for head driven generation

1993 I. Examination of problematic structures for extensions of head-driven generation and for other generation algorithms

II. Examination of the relationship between parsing and generation. Is the optimal grammar for generation also optimal for parsing?

3.6 Stellung innerhalb des Programms des Sonderforschungsbereichs

The project proposed here is related to the following other projects proposed within the SFB.

B3 Kontextuelle Constraints und Inferenzen (Rohrer, Wedekind)
This project is related to our project in that both are concerned with generation and both projects will employ complex feature structure grammars. Clearly, machine translation is the ultimate test case for the reversible approach to parsing and generation that we will investigate. Thus, the results that we provide will be quite useful for this project, and conversely this project will likely provide us with significant test cases. This project differs from our project, however, in that it is more concerned in the pragmatic problems in text planning, whereas our project is concerned with syntactic problems for generation algorithms. Thus the two projects are concerned with different aspects of the generation problem. Still, we expect there to be significant interaction between the two projects since the problem of sentence generation must be included in the problem of text generation.

B5 The Formalization and Implementation of Grammar (Rohrer, Johnson) Similar to our project, this project is also concerned with the question of how grammars can be transformed in order to achieve computational tractibility. Although this project is concerned parsing rather than generation, it is quite likely that the axiomatization of phenomena such as move-α in goverment binding theory will suggest new ways of handling such phenomena which will also be less problematic for the purpose of generation.

B6 Computational Constraints on Grammatical Axiomatizations (Hinrichs, Macias) This project differs from the Rohrer, Johnson project in that it is primarily concerned with the investigation of new computational techniques for parsing rather than with the problem of formulating computationally tractible grammars. As emphasized above, it will also not be possible for us to work on the problem of designing grammars to be used for generation without considering different possible approaches to the control problem. It is quite likely, then, that the computational techniques developed in this project will lead to new insights which could be applied in the field of generation.

A5 Die Semantische Signifikanz Syntaktischer Strukturen (Kamp, Frey) Like our project, this project is also concerned to some extent with the problem of ellipsis. Although this project is concerned with the question of whether ellipsis provides arguments for a syntactic level of LF, it is possible that the analysis of ellipsis provided will suggest ways of analysing empty heads which will be less problematic for head-driven generation algorithms.

A6 Zum Antrag für das Projekt Syntax-Fragment (Frey, Tappe) The German grammar fragment developed in this project should be very useful for our project since the fragment will have to deal with empty head constructions such as the verb second and ellipsis phenomena. The analysis provided in this fragment may suggest approaches to these phenomena which will be compatible with head-driven generation.

3.7 Ergänzungsausstattung für das Teilprojekt B4

Es bedeuten:
Pk = Personalbedarf und Personalkosten (Begründung vgl. 3.71)
SV = Sächliche Verwaltungsausgaben (Begründung vgl. 3.72)
I = Investitionen (Begründung vgl. 3.73)

1992			1993			1994		
Vergütungs– grüppe	Anzahl	Betrag in DM	Vergütungs– grüppe	Anzahl	Betrag in DM	Vergütungs– grüppe	Anzahl	Betrag in DM
105a	106	107	108	109	110	111	112	113
BAT IIa	2	163,2	BAT IIa	2	163,2	BAT IIa	2	163,2
gepr. HK	1	28,1	gepr. HK	1	28,1	gepr. HK	1	28,1
ungepr. HK	1	17,6	ungepr. HK	1	17,6	ungepr. HK	1	17,6
Gastprofessor	0,25	31,5	Gastprofessor	0,25	31,5	Gastprofessor	0,25	31,5
zusammen	4,25	240,4	zusammen	4,25	240,4	zusammen	4,25	240,4

	1992		1993		1994	
SV	Bezeichnung	Betrag in DM	Bezeichnung	Betrag in DM	Bezeichnung	Betrag in DM
	114	115	116	117	118	119
	Verbrauchsmittel	4	Verbrauchsmittel	4	Verbrauchsmittel	4
	Reisemittel	5,2	Reisemittel	5,2	Reisemittel	5,2
	Bücher	2,4	Bücher	2,4	Bücher	2,4
	Druck	1	Druck	1	Druck	1
	Kopien	2	Kopien	2	Kopien	2
	zusammen	14,6	zusammen	14,6	zusammen	14,6
I	Investitionen insges. DM		Investitionen insges. DM		Investitionen insges. DM	
	120		121		122	
	53,2					

	Name, akad. Grad, Dienstellung	engeres Fach des Mitarbiet.	Bezeichn. des Instituts	Zeit	be– rat.	im SFB tätig seit	Einstuffung
	123	124	125	129	130	131	132
Grundausstattung 3.71.1	1. Hinrichs, Prof.Dr. Univ.Prof.	Sprw. (Semantics, Syntax,Computer– linguistik	SNS	5			
wissenschaftliche Mitarbeiter (einschl. Hilfskräfte)	2. Gerdemann, Dr. wiss. Angest.	Sprw (Computer– linguistik, Syntax)	SNS	10			
3.71.2 nichtwissenschaft. – Mitarbeiter	3. Saile Systeme Prog.						
Ergänzungsausst.							
	4. N.N. (3 Mo. pro Jahr)	Sprw. (Syntax, Computerling.)		38,5			Gastprof. (C4)
	5. N.N.	Sprw. (Syntax, Semantics)		38,5			BAT IIa
	6. N.N.	Sprw. (Computer– linguistik)		38,5			BAT IIa
3.71.3 wissenschaftl. Mitarbeiter (einschl. HK)	7. N.N. 8. N.N.	Sprw. (Syntax) Sprw.		19,25 19,25			gepr. HK ungepr. HK
3.71.4 nichtwissenschft. Mitarbeiter							

Description of responsibilities

1. Project Leader: Consultation for development of semantic aspects of grammar and processing algorithms, teaching responsibilities, leadership of project.

2. Project Leader: Consultation for development of syntactic aspects of grammar and processing algorithms, teaching responsibilities, leadership of project.

3. System programming: system maintenance, backups, updating of software.

4. N.N. Gastprofessor: The project would greatly benefit from outside consultation with prominant researchers in the following three areas: development of specific generation algorithms, extensions to unification based grammar formalisms, and the treatment of headedness in the theory of grammar. We plan to invite a guest professor three months each year in each of these areas at crucial phases of the project.

5. N.N. To work on the development of unification based grammar of German treating movement and ellipsis phenomenon. Should have a good background in syntax and complex feature formalisms. Some background in computational linguistics will also be desirable.

6. N.N. To work on reversible algorithms and specialized generation algorithms for complex feature formalisms. A background in parsing would be desirable.

7. N.N. To work on aspects of German grammar which are most problematic for generation algorithms. A background in syntax and knowledge of generation algorithms will be required.

8. N.N. To work on further development of the grammar. A background in syntactic theories will be required.

3.72 Aufgliederung und Begründung der sächlichen Verwaltungsausgaben (nach Haushaltsjahren)

	1992	1993	1994
	133	134	135
Mittel für Neuanschaffung von Klein–geräten (515) sowie Verbrauchs–material (522) — aus der Grundausstattung — aux der Ergänzungsuasstattung beantragt (bgl. Sp. 114–119	4	4	4

— *Vorhandene Grundausstattung siehe 2.1*

Begründung der beantragten Ergänzungausstattung

(512) Bücher und Zeitschriften:

For acquisition of new project specific publications (including dissertations, journals, working papers and technical reports.)

(522) Verbrauchsmaterial:

Office and computer supplies.

(527) Reisekosten:

To present our work at relevant conferences, such as the European ACL, IJCAI, DGfS and COLING.

To take part at German and European summer institutes.

To interact with related ongoing projects at other universities such as the projects conducted by Prof. Uszkoreit and his associates at the University of Saarbrücken.

(531b) Printing costs:

For 50 copies per person per month

3.73 Investitionen (Geräte über DM 20.000 (brutto) und Fahrzeuge für das Teilprojekt)

Bezeichnung des Gerätes	beantragt für das Haushaltjahr		
	Einzelpreis DM 1992	Einzelpreis DM 1993	Einzelpreis DM 1994
136	137	138	139
1 SUN IPC 4/40M-8-P40, 16MB, 1,2GB HD	25		
1 SUN II 4/75M-16-P40, 16MB	28,2		
Summe	53,2		

These two computers will be used for testing and developing implementations of various sophisticated generation algorithms. In order to test these algorithms with larger grammars, a significant amount of internal and external memory will be required. The SNS will provide the necessary software, such as Quintus Prolog and Allegro Common Lisp, out of its regular budget, but is unable to provide the necesary machine time on any of the machines currently available in the institute. The SUN II will act as the server for the IPC workstation applied for in this project as well as teh for the additional equipment needed for the B6 project (Hinrichs/Macias).

Bibliography

Appelt, D. (1987). Bidirectional grammars and the design of natural language generation systems. In Y. Wilks (ed) *Theoretical Issues in Natural Language Processing–3*, New Mexico State University, Las Cruces, New Mexico, 185–191.

Baker, M. (1988). *Incorporation.* Chicago University Press, Chicago.

Bouma, G. (1987) A Unification-based analysis of unbounded dependencies in categorial grammar. In J. Groenendijk, M. Stokhof and F. Veltman (eds) *Proceedings of the Sixth Amsterdam Colloquium*, ITLI, University of Amsterdam.

Dymetman, M. and P. Isabelle (1988) Reversible logic grammars for machine translation. In *Proceedings of the 2nd International Conference on Theoretical and Methodological Issues in Machine Translation of Natural Language.*

Dymetman, M., P. Isabelle and F. Perrault (1990) A Symmetrical approach to parsing and generation. In *COLING-90*, Vol. 3, 90–96.

Earley, J. (1970). An efficient context-free parsing algorithm. *Communications of the ACM*, 13(2):94–102.

Gerdemann, D. (1991) *Parsing and Generation of Unification Grammars.* Ph.D. thesis, University of Illinois. Published as Beckman Institute Cognitive Science Technical Report CS-91-06.

Gerdemann, D. and Hinrichs, E. (1988). UNICORN: A unification parser for attribute-value grammars. *Studies in the Linguistic Sciences*, 18(2):41–86.

Gerdemann, D. and Hinrichs, E. W. (1990). Functor-driven natural language generation with categorial-unification grammar. In *COLING-90 Vol. 2*, 145–150.

Hendrix, H. (1987) Type change in semantics: the scope of quantification and coordination. In E. Klein & J. van Benthem (eds) *Categories, Polymorphism and Unification.* Centre for Cognitive Science, University of Edinburgh and Institute for Language, Logic and Information, University of Amsterdam.

Hinrichs, E. and Nakazawa, T (1989a). Flipped out aux in German. In *Papers from the 25th Regional Meeting of the Chicago Linguistic Society*, 193–202.

Hinrichs, E. and Nakazawa T. (1989b). Review of *Word Order and Constituent Structure in German* by Hans Uszkoreit. *Language*, 65:141–149.

Hinrichs, E. and Nakazawa, T (1991). Subcategorization and VP structure. forthcoming *Technical report, Seminar für Natürliche-Sprachliche Systeme.*

Kay, M. (1979). Functional grammar. In *Proceedings of the Fifth Annual Meeting of the Berkeley Linguistics Society*, 17–19.

Kiss (1990) Variable subkategorisierung: Eine theorie unpersönlicher einbettung. IBM IWBS REport 161, Wissenschaftliches Zentrum, Institut für Wissenbasierte Systeme.

Larson (1985) Bare-NP Adverbs. *Linguistic Inquiry*, 16:595–621.

McCawley (1988) Adverbial NPs: Bare or clad in see-through garb? *Language*, 64:583–590.

Moortgat, M. (1988). *Categorial Investigations: Logical and Linguistic Aspects of the Lambek Calculus.* Foris Publications, Dordrecht.

Nerbonne, J., Iida, M. and Ladusaw, W. (1989). Running on empty. In Fee, E. J. and Hunt, K., editors, *Proceedings of the Eighth West Coast Conference on Formal Linguistics*, 276–288.

van Noord, G. (1989). A directed bottom-up generator for unification based formalisms. Working Papers in Natural Language Processing 4, Katholieke Universiteit Leuven.

Partee, B. and Rooth, M. (1983). Generalized conjunction and type ambiguity. In Bauerle, R., Schwarze, C. and von Stechow, A., editors, *Meaning, Use and Interpretation of Language*, 361–383. Walter de Gruyter.

Pereira, F. C. N. and Warren, D. H. D. (1986). Definite clause grammars—a survey of the formalism and a comparison with augmented transition networks. *Artificial Intelligence*, 13:231–278.

Pollard, C. (1988) Categorial grammar and phrase structure grammar: An excursion on the syntax-semantics frontier. In R. Oehrle, E. Bach and D. Wheeler (eds) *Categorial Grammars and Natural Language Structures*, Reidel, Dordrecht.

Pollard, C. and Sag, I. (1987). *An Information-Based Approach to Syntax and Semantics: Volume 1 Fundamentals.* CSLI Lecture Notes No. 13. Chicago University Press, Chicago.

Russell, G., Warwick, S. and Carroll, J. (1990). Asymmetry in parsing and generation with unification grammars: Case studies from ELU. In *ACL Proceedings, 28th Annual Meeting*, 205–211.

Sag, I. (1977) *Deletion and logical form.* Ph.D. Thesis, MIT. Distributed by Indiana University Linguistics Club.

Shieber, S. (1988). A uniform architecture for parsing and generation. In *COLING-88*, 614–9.

Shieber, S., van Noord, G. Moore, R. C. and Pereira, F. C. N. (1990). Semantic-head-driven generation. *Computational Linguistics*, 16:30–42.

Steedman, M. (1985). Dependency and coordination in the grammar of Dutch and English. *Language*, 61:523–568.

Strzalkowski, T. and Peng, P. (1990a). Automated inversion of logic grammars for generation. In *ACL Proceedings, 28th Annual Meeting*, 212–219.

Strzalkowski, T. (1990b) How to invert a parser into an efficient generator. In *COLING-90*, Vol. 2. 347–352.

Uszkoreit, H. (1987). *Word Order and Constituent Structure in German*. CSLI Lecture Notes No. 8. Chicago University Press, Chicago.

Wall, R. , and K. Wittenburg (1989) Predictive normal forms for composition in categorial grammars. In *Proceedings of International Workshop on Parsing Technologies*, 152–161.

Wedekind, J. (1988). Generation as structure driven derivation. In *COLING-88*, 732–737.

Wittenburg, K. (1987). Predictive combinators: A method for efficient parsing of combinatory categorial grammars. In *Proceedings of the 25th Annual Meeting of the Association for Computational Linguistics*, 73–80.

Woods, W. A. (1970). Transition network grammars for natural language analysis. *Communications of the ACM*, 13:591–596.

Zwicky, A. (1985) Heads. *Journal of Linguistics*, 21:1–30.