Feature-Based Inheritance Networks
for Computational Lexicons*

Hans-Ulrich Krieger
krieger@dfki.uni-sb.de

John Nerbonne
nerbonne@dfki.uni-sb.de

Deutsches Forschungszentrum für Künstliche Intelligenz
Stuhlsatzenhausweg 3
D-6600 Saarbrücken 11, Germany

Abstract

The virtues of viewing the lexicon as an inheritance network are its succinctness and its tendency to highlight significant clusters of linguistic properties. From its succinctness follow two practical advantages, namely its ease of maintenance and modification. In this paper we present a feature-based foundation for lexical inheritance. We shall argue that the feature-based foundation is both more economical and expressively more powerful than non-feature-based systems. It is more economical because it employs only mechanisms already assumed to be present elsewhere in the grammar (viz., in the feature system), and it is more expressive because feature systems are more expressive than other mechanisms used in expressing lexical inheritance (cf. DATR). The lexicon furthermore allows the use of default inheritance, based on the ideas of default unification, defined by Bouma [7].

These claims are buttressed in sections sketching the opportunities for lexical description in feature-based lexicons in two central lexical topics: inflection and derivation. Briefly, we argue that the central notion of paradigm may be defined directly in feature structures, and that it may be more satisfactorily (in fact, immediately) linked to the syntactic information in this fashion. Our discussion of derivation is more programmatic; but here, too, we argue that feature structures of a suitably rich sort provide a foundation for the definition of lexical rules.

We illustrate theoretical claims in application to German lexical structure. This work is currently under implementation in a natural language understanding project (DISCO) at the German Artificial Intelligence Center (Deutsches Forschungszentrum für Künstliche Intelligenz).

1   Introduction

The best inheritance mechanisms for representing lexical information have been Flickinger, Pollard and Wasow's [19] work on "structured lexicons", and Evans and Gazdar's [16] work on DATR. Both Flickinger's work and DATR aim to supplement feature-based grammars, but both require an explicit translation step to convert lexical information into grammatical features. Furthermore, they are both hampered in expressive power, so that they accommodate some sorts of information poorly, even information which is

---

standardly found in feature systems, e.g., disjunction, negation, and complex feature structures used as values.

The present proposal draws both from the work above on default inheritance networks and from the lexical ideas of feature-based theories in general (cf. the section on PATR-II lexicons in Shieber, [43], pp.54-61). PATR-II present lexicons via a collection of TEMPLATES or MACROS, which are purely syntactic abbreviations for feature-structure descriptions. Pollard and Sag's (1987) sketch of the lexicon in Head-Driven Phrase Structure Grammar (hence HPSG), presented in Pollard and Sag [36] and Sag and Pollard [42], extended these ideas by interpreting lexical definitions as feature-structure descriptions and inheritance specifications as subsumption statements, rather than treating them in a purely syntactic fashion. Pollard and Sag (Chap.8.2) furthermore suggest a use of lexical rules which brings their work closer to standard linguistic views (e.g., LFG, [8]).

This and most other work on feature structures, on the other hand, has failed to allow the use of DEFAULTS or OVERWRITING, which is crucial for a practical lexical tool.[1] The key advantage of default specifications is that they allow the description of SUBREGULARITIES, classes of items whose properties are largely, but not perfectly regular. In a system with default inheritance, these may be regarded not as anomalous, but rather as imperfectly regular, or regular within limits. We shall employ default inheritance regularly, perhaps most crucially in the specification of derivational relations (cf. below and cf. Flickinger et al. [19]; and Gazdar [20], [21]; and Flickinger and Nerbonne [18] for arguments supporting the use of defaults in lexical specifications). This has seemed suspicious within the context of feature systems because these were developed (in part) to allow monotonic processing of linguistic information, and the use of defaults leads to nonmonotonicity.[2] But, as Bouma [7], p.169 points out, the use of lexical defaults is a fairly harmless form of nonmonotoncity, since the lexicon is nonmonotonic only with respect to lexical development—the syntactic use of information specified via lexical default leads to none of the problems associated with nonmonotonic reasoning; e.g., inferences about phrases never need to be retracted, and the NL system may be configured to be perfectly monotonic at run-time. If we employ default inheritance for the specification of lexical information, then the inheritance hierarchy does NOT correspond to a subsumption or subtyping hierarchy—information may be overwritten which renders subsumption invalid. Care needs to be taken that the two notions of hierarchy—the classes involved in the default inheritance relationship and the feature structure types defined there—not be confused (cf. Cook, Hill and Canning [13]). The mechanism we shall employ for the default combination of lexical information is the DEFAULT UNIFICATION developed by Bouma [7]; we may employ this within the lexicon, even while eschewing its use for parsing and generation.

The present work is closest to Pollard and Sag's in that it proceeds from a view of the lexicon in which feature structures bear the burden of linguistic description. It differs from their work in advocating the use of default inheritance, and perhaps more significantly, in using ONLY feature structure descriptions to represent lexical information, including especially lexical rules. Where Pollard and Sag viewed lexical rules as operators on feature structures, we propose defining lexical rules purely in terms of feature structures. On our view, one need not assume a distinct sort of linguistic entity, lexical rule, which maps feature structures into feature structures. Instead, one begins with feature structures and a description language for them, and this suffices to characterize lexical rules. Lexical rules are thus an emergent phenomenon in language, ultimately reducible to feature-structures (cf. Section 4 for a discussion of alternative views of lexical rules).

Our purpose in this paper is to present the feature-based lexicon not as a linguistic theory of lexical structure, but rather as a framework within which such theories may be formulated, i.e., a tool for lexical description. This means that we shall at points demonstrate the formulation of competing views of lexical phenomena, e.g., matrix-based and form-based views of the inflectional paradigm, and inflectional and derivational views of the passive. It is also worth noting that, although we are quite interested in the question of efficiently processing feature-based descriptions—especially the question of lexical access for generation and recognition, we will not have anything to say about it in this paper.

It is probably worth sketching here in broad strokes the sorts of advantages we shall claim for our proposals. These lie primarily in the expressive power of the feature formalisms. The added expressive capacity is exploited in order to characterize both inflectional paradigms and derivational word formation

---

[1]But cf. Pollard and Sag [36], p.194, Note 4; Sag and Pollard [42], p.24; and Shieber [43], pp.59-61.

[2]It is probably worth noting that there have nonetheless been several attempts at using nonmonotonic inference rules in unification-based systems. Kay's [30] FUG included an ANY value, which defaulted to $\bot$ unless unified with, in which case it was $\top$; Shieber [43], p.59 presents a scheme for the default interpretation of feature structure templates; Kaplan [25] presents nonmonotonic "constraining equations" for LFG; and Gazdar et al. [22], p.29 *et passim* propose nonmonotonic FEATURE SPECIFICATION DEFAULTS.

rules as first-class linguistic objects—feature structures—of the same sort as words or phrases. We believe this proposal is made here for the first time, certainly the first time in a formalized theory. We illustrate the advantages of the added expressive power in analyses of inflectional paradigms as disjunctive further specifications of abstract lexemes, and in analyses of derivationally complex forms as instantiations of rules, where the latter are given a feature-based interpretation.

Our proposals are couched in HPSG terms, both because this is often readily understandable but also because HPSG—and to a lesser extent, FUG (Kay, [29], [30])—have most vigorously explored the hypothesis that feature structures are a sufficient representation scheme for all linguistic knowledge. The idea is that feature structures (also called "functional structures" in FUG, "attribute-value matrices" in HPSG) can encode not only syntactic information, but also semantic, pragmatic, morphological and lexical information. In HPSG, even recursively structured aspects of syntax such as syntactic rules and phrase structure trees are ultimately characterized as constraints on feature structures, so that the attempt to construe lexical information strictly within the limits provided by feature structures would not seem inappropriate.

We commence from the treatment of the lexicon in HPSG I (Pollard and Sag, [36], Chap.8) as an inheritance structure, which we accept (with minor modifications noted below). Now, HPSG I invoked "lexical rules" as operations or functions ON feature structures, but this proposal in many ways violates the spirit which infuses HPSG, that of demonstrating that all linguistic knowledge can be modeled directly in feature structures. Our proposed construal of lexical rules as feature structures obviates any use of lexical rules as operations—also familiar from LFG (Bresnan, [8]) and PATR-II (Shieber et al., [44]), especially D-PATR (Karttunen, [26]).[3] We believe therefore that the present paper is a contribution to feature-based theories, as well, in that it shows how lexical rules can be construed in terms of feature-structure. This reduces the theoretical inventory, which is by itself desirable, and, as we argue below, it has significant descriptive advantages as well.

The structure of the paper is as follows: we first continue the introduction with a discussion of the distinction between morpheme- and lexeme-based analyses. In Section 2, we summarize the use we shall make of feature structures, which is essentially that provided by HPSG. We then explore the advantages of feature-based lexicons in two central lexical topics: the treatment of inflection and derivation. (There is, of course, an influential school of linguistic thought which denies the significance of this distinction, and our tools do not presuppose the distinction—we too can generalize all lexical rules to "word formation" à la Sadock's "autolexical syntax" [41]. But feature structures offer interesting alternatives for inflection.)

1.1 Lexemes and Morphemes

Before we examine the feature-based analysis of inflection and derivation in more detail, it is worth reminding ourselves that the developing research paradigm of computational lexicology—within which this paper might be located—differs from computational morphology (e.g., two-level or finite-state morphology) in that the former takes a LEXEME-based view of lexical variation (such as inflection and derivation), while the latter generally takes a MORPHEME-based view. While we do not aim here to settle any debate over which is preferable, it may be useful to sketch the differences.[4]

A lexeme is an abstract unit which characterizes what is common among the infectional variants of a word. It contains, for example, what is common among the variants of the verb *institute (institutes, instituting, instituted)*, etc, but NOT *institution, institutional, reinstitute,...*, all of which are related, but distinct lexemes. Note that the lexeme is an abstract unit, which need not even be associated with any particular form. The lexeme-based view of lexical processes such as inflection and derivation maintains that these processes are based on properties of lexemes. We clarify this below after sketching the morpheme-based alternative. A morpheme is also an abstract unit which is the minimal unit to which meaning is attached. Note that a lexeme may contain meaningful parts, e.g., *ab- + leit- + bar*—this is a lexeme with its own set of inflectional variants (the case, gender, number, grade paradigm of German adjectives). The question of whether lexical processes are morpheme-based or lexeme-based thus centers on the analysis of such complex words. Are these optimally analyzed solely in terms of component morphemes (morpheme-based), or must one take into account intermediate lexemes (lexeme-based)? In the example *ableitbar,* can one derive all properties from *ab-, leit-* and *-bar*, or must one take into account the intermediate lexeme *ableiten*?

---

[3]And nearly everywhere else: GPSG metarules (Gazdar et al., [22]) and Categorial Morphology treatments (Hoeksema, [23]) are quite similar in treating lexical rules as fundamentally distinct from lexical entries.

[4]Cf. Matthews [35] pp.20ff for a much more thorough defense of this material; cf. Anderson [2], [3] and Zwicky [49], [50] for more recent defenses of the lexeme-based view.

Returning to our English example to clarify further the distinction between the morpheme-based and lexeme-based views, we note that it is very unlikely that *institute* could be divided into further meaningful parts, so that we may examine the morpheme *institute* (which constitutes all of the material of the lexeme above), and which also appears in *institution, institutional, reinstitute, institutionalize...* As we see, a single morpheme may be shared by many lexemes (but never *vice versa*). The morpheme-based view analyzes lexical processes as depending on the properties of morphemes.

There is a fairly clear division among the practitioners of each type of analysis. Linguists are fairly unanimous in seeing lexical processes as lexeme-based, while computational linguists have generally conducted morpheme-based analyses. The computational advantage of morpheme-based analysis is fairly easy to see: one can develop procedures for isolating the morphemes in a given form, and assign properties to the form based on these. Lexemes, involving (as they potentially do) nontransparent combinations of morphemes, are more difficult to recognize, are therefore more frequently stored *in toto*, and are thus ultimately more demanding of memory resources—there are simply many more lexemes than morphemes (perhaps an order of magnitude more).

Linguists have been assiduous in turning up cases where the larger number of lexemes seems useful in analyzing lexical processes. The suffixation of *-bar* in German, which we examine below in detail, provides examples of the most important sorts of cases.[5] The crucial observation is always of the form that important properties of complex words depend NOT on the morphemes of the complex, but rather on the lexemes. Wellformedness is one such important property. And it turns out that the wellformedness of forms with the suffix *-bar* cannot be predicted only on the basis of the component morphemes. We find patterns such as the following:

morpheme	lexeme 1	lexeme 2
meid-	**meidbar*	*vermeidbar* 'avoidable'
lad-	**ladbar*	*aufladbar* 'loadable'
hab-	**habbar*	*handhabbar* 'manageable'
stell-	**stellbar*	*einstellbar,* 'adjustable'
		vorstellbar 'imaginable'
meß-	*meßbar* 'measurable'	**bemeßbar*
arbeit-	**arbeitbar*	*bearbeitbar* 'workable'

This table demonstrates that the wellformedness of *-bar* derivatives cannot be predicted on the basis of the stem morphemes with which they combine, and the last pair suggests that prefix morphemes are likewise poor predictors. (This is true, but we won't adduce further evidence here. It is trivial (but very tedious) for a German speaker with a dictionary to collect this evidence.) It is worth anticipating one reaction to the evidence above: one might object that there are perfectly respectable accounts of the illformedness of some of the examples here—some are not derived from transitive verbs, for example. But this sort of objection is merely a more detailed diagnosis of the same problem: the illformedness arises when the lexeme source of a derived word is not transitive. There is no claim made here that these patterns are ultimately inexplicable, merely that they are inexplicable on the basis of morphemic analysis. The hypothesis of a lexeme base enables more exact hypotheses in this area (e.g., about transitivity).

A second important property of derived words is their meaning, and here again, the best predictor of derived meaning is the meaning of component lexemes rather than that of component morphemes. We can illustrate the evidence for this using the same suffix, *-bar*. The general meaning accruing to a *-bar* adjective is that it describes the property of something which could stand (in object postion) in the relation denoted by the verb. Thus something is *faxbar* if one can fax it. The cases which demonstrate the lexeme basis of derivation all involve some irregularity in meaning. For example, German *eßbar* 'edible' involves a slight narrowing in meaning from the verb *essen* 'eat', since something must be capable of being safely eaten to be *eßbar*. More complicated derivatives involving these two morphemes could either preserve this narrowing, in which case they would appear to support the hypothesis of a lexeme basis, or they might fail to preserve it. The example of *Eßbarkeit* 'edibility' indicates that the narrowing is exactly preserved. And in general, meaning changes are persistent in further derivation. For a further example, note *Kostbarkeit* 'valuableness' derived from the semantically irregular *kostbar* 'valuable'.

---

[5]A great deal of what we analyze below may be found in a very thorough study of this process, Jindrich Toman's *Wortbildung* [47].

A third important property of derived words is their form, and form is likewise lexeme-dependent. The form of the argument here is the same as that above: irregularity is persistent throughout further derivation. Thus irregular *sichtbar* 'visible' (instead of **sehbar*) is found further in *Sichtbarkeit*.[6]

It is worth noting that *-bar* suffixation is a derivational process, and that most of the evidence for a lexeme basis has been accummulated from studies of derivation. We shall likewise present a lexeme-based treatment of inflection below, however. The same sorts of evidence for a lexeme basis may be adduced in the case of inflection, but rather less of it than in the case of derivation. There are, e.g., verbs whose paradigms differ from those of their component morphemes, e.g., the weak verbs *handhaben* and *veranlassen*, which are derived from the strong *haben* and *lassen*. If inflection depends on morphemes, these examples must be analyzed as involving distinct pairs of morphemes.[7] Matthews [35], Chap. VIII, presents arguments of a different sort that inflection should not be reduced to manipulations of morphemes.

We have continued at some length to justify our choice of lexeme-based analysis here, in order to emphasize that this is a deliberate, and not merely a customary assumption. Morpheme-based work has its purpose under this scheme, however. In particular, since our treatment of derivation allows regular derivations not to be listed in the lexicon, we need a method of recognizing the parts of a regular derivation in order to assign the correct properties to it. As we see it, a fully regular derivation such as *faxbar* 'faxable' need not appear in the lexicon at all. Its properties are fully specified under the specifications for the lexeme *fax-* 'to telefax', the suffix *-bar*, and the morpholological head-complement rule scheme. But in order to recognize *fax-* and *-bar* in *faxbar*, some analysis must be performed, and morpheme-based processing seems well-suited for this purpose.

1.2 Morphotactics and Allomorphy

There is a traditional distinction in morphology between MORPHOTACTICS, the arrangement of morphological elements into larger structures, and ALLOMORPHY or MORPHOPHONEMICS, variations in the shape of morphological units (cf. Anderson [2], p.147). It is our goal here to show how morphotactics may be subsumed into the lexicon, and we shall discuss this at length in Sections 3 and 4 below. But we shall deliberately have very little to say about morphophonemics, which we do not intend to treat here. We take up this distinction more concretely in Section 3.1 below, but only to reinforce the point here: we do not propose subsuming morphophonemics into the lexicon.

2 Background

In this section we review the background material in the theory of feature-based description. This section may be skipped over by those familiar with feature structure theory and HPSG.

2.1 Feature Structures

The fundamental analytical tool of FEATURE-BASED or UNIFICATION-BASED theories is the feature structure, represented by the attribute-value matrix (AVM)—a set of pairs of ATTRIBUTES (such as PER) and VALUES (such as FIRST, SECOND or THIRD). Shieber [43] is the standard introductory reference to feature-based grammars, and we assume basic familiarity with this sort of analysis; here we review only the bare essentials needed for the lexical analysis below. Feature structures are mathematical objects which model linguistic entities such as the utterance tokens of words and phrases. It is important to note that values may themselves be feature structures, so that we allow that an AGR attribute may specify its value using the complex AVM below, resulting in a hierarchical AVM:

$$\left[\text{ AGR } \left[\begin{array}{l} \text{PER THIRD} \\ \text{NUM SG} \end{array} \right] \right]$$

A useful conceptualization of feature structures is that of rooted, directed labeled graphs, where values correspond to nodes, attributes to labeled edges, and where the AVM as a whole describes the root.

---

[6]One would expect irregular syntactic properties to show the same persistence through derivation, but we do not know of relevant studies.

[7]A further sort of example may be forthcoming if one examines perfect participle formation in German. The generalization to be captured is that a prefix *ge-* is employed when stress is on the first stem syllable (stem or inseparable prefix) of the lexeme. The argument is complicated by the fact that this is often—perhaps always—predictable on the basis of the morphemes involved.

This conceptualization is particularly useful when it comes to specifying values within complex structures, which we do by concatenating attributes to form paths from the root into the interior of the structure. Because AVM descriptions can quickly become quite large, we will employ path descriptors, abbreviating, e.g., the person information in the AVM above to simply:

$$\left[\ \text{AGR|PER THIRD} \ \right]$$

We shall even take the liberty occasionally of suppressing prefixes where no confusion arises, and specifying (as equivalent to the above):

$$\left[\ \text{PER THIRD} \ \right]$$

But we shall take care to do this only where the type of the AVM is clear (so that it is the type of AVM in which AGR occurs, as opposed to one in which HEAD|AGR occurs, etc.). Such abbreviations are generally disambiguated when interpreted in the light of type information. They reduce the complexity of AVM's a great deal, making them easier to read and write.

We shall have frequent occasion to employ AVM's with DISJUNCTIVE value specifications. These are descriptions of objects whose value is included in one of the disjuncts, i.e., it is FIRST or THIRD:

$$\left[\ \text{AGR|PER \{FIRST, THIRD\}} \ \right]$$

In order to link particular choices with formal elements, we make extensive use of DISTRIBUTED DIS-JUNCTIONS, investigated by Backofen, Euler and Görz [5] and Dörre and Eisele [14]. This technique was developed because it (normally) allows more efficient processing of disjunctions, since it obviates the need to expand them to disjunctive normal form. It adds no expressive power to a feature formalism (assuming it has disjunction), but it abbreviates some otherwise prolix disjunctions:

$$\begin{bmatrix} \text{PATH1 } \{_{\$1} \ a, b\} \\ \text{PATH2 } \{_{\$1} \ \alpha, \beta\} \\ \text{PATH3 } [\ \dots \ ] \end{bmatrix} = \left\{ \begin{bmatrix} \text{PATH1 } a \\ \text{PATH2 } \alpha \\ \text{PATH3 } [\ \dots \ ] \end{bmatrix} , \begin{bmatrix} \text{PATH1 } b \\ \text{PATH2 } \beta \\ \text{PATH3 } [\ \dots \ ] \end{bmatrix} \right\}$$

The two disjunctions in the feature structure on the left bear the same name '$1', indicating that they are a single alternation. The sets of disjuncts named covary, taken in order. This may be seen in the right-hand side of the equivalence. Two of the advantages of distributed disjunctions may be seen in the artificial example above. First, covarying but nonidentical elements can be identified as such, even if they occur remotely from one another in structure, and second, features structures are abbreviated. The amount of abbreviation depends on the number of distributed disjunctions, the lengths of the paths PATH1 and PATH2, and—in at least some competing formalisms—on the size of the remaining structure (cf. [PATH3:   ] above).[8]

A final point to be appreciated about the use of feature structures is that two different attributes may be specified as having the same value, even when that value is unknown. For example, we might specify subject verb agreement in the following fashion, where the boxed numbers are just "tags" that identify the values as being the same:

$$\begin{bmatrix} \text{AGR } \boxed{1} \\ \text{SUBJECT } [\ \text{AGR } \boxed{1} \ ] \end{bmatrix}$$

Returning to the graph conceptualization above, the need for this sort of specification demonstrates that the class of graphs we're interested in are not simply trees, but objects of the more general class of directed graphs.

What we have written above are AVM's or FEATURE DESCRIPTIONS—they describe the abstract objects (feature structures) we use to model linguistic phenomena. Attribute-value descriptions such as the ones above are standardly interpreted in one of two ways: either directly, as descriptions of linguistic objects (cf. Johnson [24]; Smolka [46]), or algebraically, as specifications of FEATURE STRUCTURES (cf. Pollard and Sag, [36] Chap. 2; Carpenter, [11], Chap. 17), which then may be regarded as models of the linguistic objects. The distinction is mathematically interesting, but it will not be pursued here, since it is irrelevant

---

[8]Cf. Backofen et al. [5] for a discussion of a third advantage of distributed disjunctions, namely a normal increase in processing efficiency.

to grammars and lexicons written in this notation. Indeed, we shall often speak informally of the AVM's as if they were the linguistic objects, as is common.

UNIFICATION is the normal means of combining the compatible information in two or more feature structures into a single one. Unification fails when there is incompatible information. We shall not provide a formal definition of the notion here, even though it is used frequently below, since it is defined in the works cited above. Here is an example of two AVM descriptions which are compatible, and a further example of two which are not:

$$\left[\text{ AGR } \left[\text{ PER THIRD } \right] \right] \sqcap \left[\text{ AGR } \left[\text{ NUM SG } \right] \right] =$$

$$\left[\text{ AGR } \left[\begin{array}{l} \text{PER THIRD} \\ \text{NUM SG} \end{array} \right] \right]$$

$$\left[\text{ AGR } \left[\begin{array}{l} \text{PER THIRD} \\ \text{NUM PL} \end{array} \right] \right] \sqcap \left[\text{ AGR } \left[\text{ NUM SG } \right] \right] = \bot$$

(Note the incompatible specification of the value at the AGR|NUM path.)

One point about the use of coreference and disjunction is worth special mention: there is "normally" no way to specify that structure is shared between one disjunct of a (possibly distributed) disjunction and anything outside the disjunction. Thus the feature structure below is misleading:

(1)
$$\left[\begin{array}{l} \text{ATTR1 } \boxed{1} \\ \text{ATTR2 } \{ \left[\text{ ATTR3 } \boxed{1} \right] , \left[\text{ ATTR4 } \boxed{1} \right] \} \end{array} \right]$$

We said that there is "normally" no such allowable specification, but this should be clarified: the semantics of the formula here is clear enough (it can be readily reduced to disjunctive normal form), but two things must be noted. First, the formulation is somewhat deceptive, in that the value in [ATTR1 $\boxed{1}$] in the formula above is implicitly disjunctive. This may be seen if one considers the result of unifying the above with a (nondisjunctive) description:

$$\left[\text{ ATTR2 } \left[\begin{array}{l} \text{ATTR3 } a \\ \text{ATTR4 } b \end{array} \right] \right]$$

which yields:

$$\left[\begin{array}{l} \text{ATTR1 } \{a,\ b\} \\ \text{ATTR2 } \left[\begin{array}{l} \text{ATTR3 } a \\ \text{ATTR4 } b \end{array} \right] \end{array} \right]$$

This is perhaps not immediately appreciated when such structures are encountered. The second point is related, namely, that the administration of coreference which spans disjunction can become fairly involved. It is worth noting that neither of these points calls the legitimacy of the coreference spanning disjunction into question, they merely point out that it is a sensitive area. For this reason, we shall not avoid it completely here. It will allow us more succinct representations.

The reason why a restriction against coreference spanning disjunction is not felt to hinder expressivity is clear: there is a reasonably succinct alternative to the form (1) above:

$$\left[\begin{array}{l} \text{ATTR1 } \{\$_1\ \boxed{1}, \boxed{2} \} \\ \text{ATTR2 } \{\$_1\ \left[\text{ ATTR3 } \boxed{1} \right] , \left[\text{ ATTR4 } \boxed{2} \right] \} \end{array} \right]$$

Notice that coreferences do not span disjunctions here, since we employ coreferences only within single alternatives of a distributed disjunction.

The significance of feature structures for the present work is twofold: first, contemporary grammatical work in computational lingusitics is nearly universally conducted in feature-based grammars, so that it is important that lexical specifications for feature-based theories be clearly interpretable. The present work also aims to provide a lexicon for feature-based grammar, where the lexicon may be viewed as a (disjunctive) collection of feature descriptions. Feature structures will be the only content of the lexical specifications in the lexicon; in other words, specifications associate word class types with feature structure descriptions.

This brings us to the second point of connection between feature-based theories and lexicon theory. The work on "structured lexicons" cited in the introduction by Flickinger et al [19], Flickinger [17] and

Evans and Gazdar [16] emphasized the value of lexicons in which specifications were as free as possible of redundancy, and these works eliminated redundancy by exploiting a relation of INHERITANCE between lexical classes, realized as a relation between nodes in a directed graph (inheritance hierarchy). In structured lexicons, the word class TRANSITIVE VERB inherits properties from the word class VERB as well as the word class TRANSITIVE. The usual formulation in feature-based theories takes the inverse relation 'bequeath' as primitive. This relation is very naturally characterized in feature-based theories as the relation 'less-informative-than', or SUBSUMPTION, which we symbolize '$\sqsupseteq$'. For example, the feature structures below stand in this relation:

$$\left[\text{ AGR } \left[\text{ NUM SG } \right] \right] \sqsupseteq \left[\text{ AGR } \left[\begin{array}{l} \text{PER THIRD} \\ \text{NUM SG} \end{array} \right] \right]$$

We shall therefore formulate statements about inheritance (the inverse of bequeath) using '$\sqsubseteq$'. We implement this notion of inheritance using a procedure, 'unify' (or 'default-unify'), which unifies the information of every superclass with LOCAL idiosyncratic information to determine a fully expanded prototype of the class in question. Feature description languages thus provide a natural formalization for work in structured lexicons.

A final aspect of modern feature theories that we shall have cause to exploit is their use of TYPING (cf. Carpenter, [11] for a presentation). A type system imposed on a system of feature structures has several tasks: first, provides a means of referring to CLASSES of feature structures of a given restricted sort. We shall put this to good use, e.g., in representing derivational relationships as complex inheritance. Second, attributes are restricted by type to being appropriate on a limited class of feature structures; thus the attribute VFORM will be limited in appropriateness to objects of type *verb* or *verbal-head*. Third and finally, the values of attributes will be restricted by type.

2.2 HPSG

Although most of what we propose might be realized in formalisms weaker than HPSG, it is worth noting that we shall employ RECURSIVE TYPE SPECIFICATIONS of a kind found in HPSG, but generally not elsewhere. In HPSG the type *sign* has an attribute SYNTAX|LOCAL|SUBCAT which is restricted in value to lists of signs. This attribute encodes SUBCATEGORIZATION information, which is lexically based in HPSG, much as it is in Categorial Grammar (Bach, [4]). Grammatical heads specify the syntactic and semantic restrictions they impose on their complements and adjuncts. For example, verbs and verb phrases bear a feature SUBCAT whose content is a (perhaps ordered) set of feature structures representing their unsatisfied subcategorization requirements. Thus the feature structures associated with transitive verbs include the information:

$$\left[\begin{array}{l} \textit{trans-verb} \\ \text{SYN|LOC|SUBCAT } \langle \left[\begin{array}{l} \textit{NP} \\ \text{CASE ACC} \end{array} \right] , \left[\begin{array}{l} \textit{NP} \\ \text{CASE NOM} \end{array} \right] \rangle \end{array} \right]$$

(where *NP* is the type of noun phrase signs, and *trans-verb* the type of transitive verb sign).

The significance of subcategorization information is that the subcategorizer may combine with elements listed in its SUBCAT feature (perhaps only in a particular order) in order to form larger phrases. When a subcategorizer combines with a subcategorized-for element, the resultant phrase no longer bears the subcategorization specification—it has been discharged (cf. Pollard and Sag, 1987, p.71 for a formulation of the HPSG SUBCATEGORIZATION PRINCIPLE). We shall have cause to return to subcategorization in our presentation of derivation.

In order to appreciate the point about recursive specification, let us regard the subcategorization list as represented in [FIRST, REST] form (so that every SUBCAT either is null or occurs in [FIRST, REST] form). Then, the important point is to note that we have a type *list*, one of whose attributes, REST is restricted to values of type *list*, including the empty list. This is a recursive type specification. In general, SUBCAT is restricted to taking values which are of the type *list(sign)*—and this attribute occurs within signs. A similar recursion obtains when we define the type *tree* as a *lexical-sign* or a *sign* whose attribute DAUGHTERS is a list of signs of the type *tree*. We shall employ recursive type specifications in a proposal for the representation of derivational relationships.

3 Inflection

We turn our intention here to the treatment of inflectional paradigms as exemplified by verbal, adjectival and nominal paradigms in German. We illustrate with a fairly traditional representation of the inflectional endings used in the present active paradigm of weak verbs below (and we explore alternatives to this below):

	sg	**pl**
1st	+ e, *kriege*	+ en, *kriegen*
2nd	+ st, *kriegst*	+ t, *kriegt*
3rd	+ t, *kriegt*	+ en, *kriegen*

3.1 Interface to Morphophonemics

It is worth noting here that the forms found in paradigms may be more complex than simple concatentations of stems and the inflectional endings specified in paradigms. Full forms cannot always be derived from simple concatenation since internal sandhi rules, i.e., rules of morphological juncture of various sorts, may need to be applied. In fact, this is the case in the simple example here. For weak verb stems ending in alveolar stops, a rule of schwa-epenthesis must be invoked in the third-singular position (and this is NOT a general phonological process in German).

	sg	**pl**
1st	arbeit + e	arbeit + en
2nd	arbeit + st	arbeit + t
3rd	arbeit + t, *arbeitet*	arbeit + en

The status of *umlaut* in strong conjugations (*ich schlage, du schlägst*) is similarly sensitive to morphological juncture—thus it is triggered by the second singular present /st/, but not by the second singular preterite /st/.

We draw attention to these phenomena only to emphasize that, while we are aware of such complications, we do not intend to treat them here because a proper treatment involves morphophonemic detail which is not our primary focus here.[9] Our focus here will be on the morphological interface to syntax, rather than on the interface to morphophonology.

The interface to allomorphy is then quite simple:

$$
\left[\text{MORPH} \left[\begin{array}{l} \text{STEM } \boxed{2} \\ \text{ENDING } \boxed{2} \\ \text{FORM } \boxed{2} \, \& \, \boxed{3} \end{array} \right] \right]
$$

where '&' designates the concatenation operation between morphs. It is the task of the allomorphy to "spell out" the combination under FORM. We also emphasize that we indulge below in the convenient fiction that the inputs to allomorphy can be adequately specified using strings—even though we are convinced that a more abstract characterization (e.g., in feature structures) is necessary.

We shall not provide a detailed proposal—either here or in the following section on derivation—about the interface of the lexeme-based lexicon to morphophonemics. The main issue we see as significant here is the relationship between the lexeme-based lexicon and morpheme-based generalizations which might arise in morphophonemics. This is a subject of current investigation.

---

[9]For the theorist who would like to maintain the interesting claim that feature structures can represent ALL linguistic knowledge, the feature-based treatment of morphophonemics is of great potential interest. Cf. Bird [6]. For efficiency reasons, however, a second path might be chosen, viz., the employment of a hybrid feature-based two-level morphology. This has efficiency advantages, and it is our intention to pursue this line. Cf. Trost [48]. A third possibility would be to simply anticipate the effects of morphophonemics in the specification of paradigms (even if this results in the multiplication of paradigms).

3.2 Approaches to Inflection

The dominant approaches to the treatment of inflection in computational linguistics have been either (i) to model inflection using collections of lexical rules, or (ii) to employ two-level morphology. The deployment of lexical rules may be found in Flickinger's approach (cf. Flickinger, [17], pp.107-110), in the Alvey tools project (cf. Ritchie et al., [38], p.298), and in HPSG (cf. Pollard and Sag, [36], pp.209-213). Paradigmatic morphology improves upon these ideas by defining the paradigm as a sequence of lexical rules on which subsumption relations can be defined (cf. Calder, [9]), but the fundamental analytical tool is still the lexical rule.

In this view, the inflectional paradigm above is described by postulating rules which relate one paradigm element to another (or relating it to another form), including perhaps a rule to derive first singular forms from infinitives. While nothing would prohibit a lexical rule from operating on abstract stems to create forms, this was seldom done (cf. Karttunen's LFG-style treatment of passive in D-PATR, Karttunen, [26], pp.12-14 for an exception). In the following section on derivation (Section 4), we sketch a feature-based theory of lexical rules within which these notions of the paradigm could be recast, but we prefer to demonstrate the flexibilty of the feature-based approach here by developing a more purely paradigmatic view. While it would clearly be possible to formulate a rule-based view of the paradigm in feature structures, the analytically more challenging task is to describe directly the abstract variations which constitute paradigms—i.e. to try to characterize paradigms directly without recourse to lexical rules.

The two-level approach to inflection may be found in Koskenniemi [32] (for an interesting extension in the direction of feature-based processing, cf. Trost [48]). This differs from the current proposal in being morpheme-based. It is, however, compatible with various lexical structures, as is demonstrated by its use in the Alvey project, noted above [38].

The direct characterization of the paradigm has been the alternative approach both in linguistics (cf. Matthews, [34], Chap.IV) and in computational linguistics (cf. Evans and Gazdar's DATR, [16], and Russell et al.'s ELU lexicon, [40]). The fundamental idea in our characterization is due to the work in DATR, in which paradigms are treated as alternative further specifications of abstract lexemes. We express the same fundamental idea in feature structures by defining paradigms as large disjunctions which subsume appropriate lexical nodes. Each disjunct represents conceptually one element of the paradigm, and specifies what is peculiar to that element. The fundamental person-number verb paradigm in German is just an association of forms with the six disjuncts in the disjunctive normal form of the following disjunction:

$$\left[\text{AGR} \left[\begin{array}{l} \text{PER} \ \{1\text{ST}, 2\text{ND}, 3\text{RD}\} \\ \text{NUM} \ \{\text{SG}, \text{PL}\} \end{array} \right] \right]$$

3.3 A "Word and Paradigm" Approach

We may now employ distributed disjunctions (cf. Section 2) to link the fundamental alternation in (1) to the expression of forms. We obtain the following description of the present weak paradigm:

$$\left[\begin{array}{l} \text{MORPH} \left[\begin{array}{l} \text{STEM} \ \boxed{2} \\ \text{ENDING} \ \boxed{3} \ \{_{\$1} \ \text{"e", "st", "t", "n", "t", "n"}\} \\ \text{FORM} \ \boxed{2} \ \& \ \boxed{3} \end{array} \right] \\ \\ \text{SYN|LOCAL|HEAD|AGR} \ \left\{_{\$1} \left[\begin{array}{l} \text{PER 1ST} \\ \text{NUM SG} \end{array} \right], \left[\begin{array}{l} \text{PER 2ND} \\ \text{NUM SG} \end{array} \right], \cdots, \left[\begin{array}{l} \text{PER 3RD} \\ \text{NUM PL} \end{array} \right] \right\} \end{array} \right]$$

Each of the disjunctions tagged by '$1' constitutes effectively the same set of alternatives, taken in sequence. Thus we understand the feature structure above as denoting a disjunction of six disjuncts, the first of which has "e" as a value for MORPH|ENDING AND $\left[\text{PER 1ST}, \text{NUM SG} \right]$ (1st-sing) as a value of SYN|LOCAL|HEAD|AGR; etc. This node may be inherited by all German weak verbs.

In general, in representing a paradigm, the EXPONENTS of the paradigm are listed as alternatives under [MORPH|ENDING $\{_{\$n}\ldots\}$], while their associated PROPERTIES appear elsewhere within the feature structure under the same distributed disjunction '$\{_{\$n}\ldots\}$' (for this terminology, cf Matthews, [35]).

One advantage of representing paradigms in this fashion—as opposed to representations via lexical rules, as in Flickinger [17] or in the other rule-based approaches cited above—is that the paradigm is represented in the same formalism in which word classes and lexemes are represented. A paradigm may thus participate in the same inheritance relationships that relate word classes and individual lexemes. We

may, e.g., represent modal paradigms as inheriting by default from standard paradigms, modifying only the "morph" value:

$$\left[\; \text{MORPH|ENDING} \; \{_{\$1} \text{ "", "st", "", "n", "t", "n" } \} \; \right]$$

(Note that we assume that the co-naming of disjunctions is inherited, so that the endings specified above are still to be understood as covarying with syntactic agreement features. The specification may be this compact because it expoits the nonlocal information in naming disjunctions.) The modal verb *sollen* is an example of this subparadigm:

	sg	**pl**
1st	soll +	soll + n
2nd	soll + st	soll + t
3rd	soll +	soll + n

The example is misleading in that we have shown the only modal paradigm with just this distinction from standard paradigms. We are aware that modal paradigms are in general also characterized by stem vowel alternations *kann, können*, etc., and we should have to represent this information as well. We could, e.g., allow lexemes to have a present singular stem and a present plural stem which are normally identical, but which are distinguished for these modals (and a very few other verbs such as *wissen*), or we might even try to describe the vowel alternations directly, if this were not going further into the morphophonemics than we care to (at least in this paper).

Genuine SUPPLETION we likewise propose to treat via default overwrting:

$$\left[\begin{array}{l} \text{MORPH} \quad \left[\begin{array}{l} \text{STEM} \\ \text{ENDING} \\ \text{FORM } \{_{\$1} \text{ "bin", "bist", "ist", "sind", "seid", "sind"} \} \end{array} \right] \\ \\ \text{SYN|LOCAL|HEAD|AGR} \; \left\{_{\$1} \left[\begin{array}{l} \text{PER 1ST} \\ \text{NUM SG} \end{array} \right], \left[\begin{array}{l} \text{PER 2ND} \\ \text{NUM SG} \end{array} \right], \cdots, \left[\begin{array}{l} \text{PER 3RD} \\ \text{NUM PL} \end{array} \right] \right\} \end{array} \right]$$

Here we exploit default overwriting to describe the suppletive forms of *sein*. Note that we avoid the sticky question here of what the STEM and ENDING of such suppletive word forms ought to be. Perhaps a more graceful means of avoiding this question would be to deny that the forms are concatenations of STEM and ENDING (which we have not done above—we merely fail to identify them).

A DEFECTIVE PARADIGM, e.g., that of the verbs *dünken* 'to think', or *jemandem an etwas liegen* '(for something) to be important to someone', which occur only in the 3rd person singular, may be analyzed as a more extreme instance of overwriting inheritance—the entire disjunction is eliminated. (It is worth noting that non-overwriting analyses may also be formulated here, just as in most other places.)

$$\left[\; \text{SYN|LOCAL|HEAD|AGR} \; \{_{\$1} \perp, \perp, \left[\begin{array}{l} \text{PER 3RD} \\ \text{NUM SG} \end{array} \right], \perp, \perp, \perp, \} \; \right]$$

Of course, one can formulate alternative linguistic descriptions of the requirement that this verb appear only in the third-person singular. For example, rather than say that the other forms do not exist, which is roughly the content of the specification above, one could postulate that impersonal verbs subcategorize for an (abstract) subject which is 3rd-sg. Our goal here is not to defend these particular "paradigmatic gap" analyses, but rather to show that the phenomenon may be satisfactorily formulated in feature structures.[10] A futher alternative would be to overwrite the AGREEMENT value nondisjunctively. There are probably alternative descriptions available.

These examples of paradigm inheritance point out limitations to the technique worth noting, even if they eventually prove to be convivial. First, although we can allow the inheritance of distributed disjunctions to model the inheritance of paradigms, there is no way to make sense of an inherited paradigm

---

[10]Better candidates for genuine paradigmatic gaps in German are e.g., *Eltern* 'parents', which lacks a singular (except in the speech of biologists), or the verb *verscholl, verschollen* 'to get lost', which lacks all finite present forms. The English auxiliaries *come* and *go* are also frequently cited to demonstrate the existence of gaps in paradigms. Thus *I come/go see her daily; *He comes/goes see her daily*, even though *He wants to come/go see her daily*.
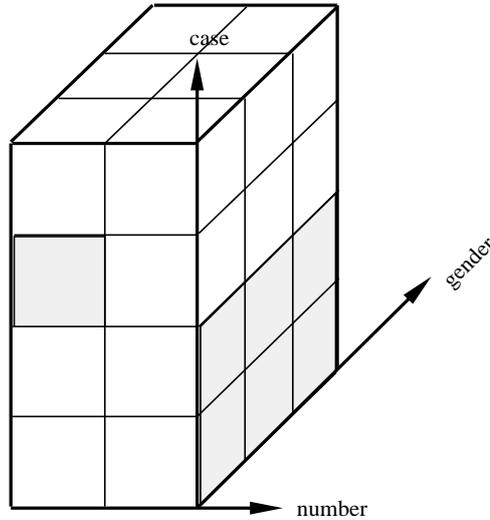
Figure 1: A three-dimensional view of the German determiner (or nondeclinable noun) paradigm, with dimensions corresponding to person, number, and case.

being larger or smaller than the ancestor from which it inherits—so that we cannot sensibly construe a four-element paradigm as inheriting from a six-element one, or *vice versa*. This limitation arises because a distributed disjunction is always an alternation of elements in order. Second, there is no way to note that a single form in a paradigm is exceptional without respecifying the entire paradigm—the disjunction must be respecified as a whole. This stems from the fact that there is no way to identify a particular alternation within the distributed disjunction. Thus the defective paradigm above had to respecify the entire paradigmatic disjunction (this suggests perhaps using features to identify forms, which is the technique employed in DATR).

This in general is the case, and it brings us to a slight modification of the style of representation we shall employ. We examine this in the following section.

3.4 Matrix- vs. Form-based Approaches

The specifications above are written in what might be called the "matrix" style of the word-and-paradigm model—every cell in the matrix of cross-cutting paradigmatic distinctions is specifically assigned a form value. The incorporation of a further dimension with n distinctions would increase paradigm size by a factor of n—even if NO new forms were introduced. This is the presentation preferred in didactic grammars. Thus we might view a determiner or nominal paradigm in the fashion shown in Figure 1.

Opposed to this "matrix-based" view of the paradigm is a "form-based" view—only distinct forms are assigned feature specifications. The same (weak present) paradigm can be represented as an alternation of only four forms, two of which are (disjunctively) underspecified for agreement values (note that we employ nested distributed disjunctions in order reduce the top-level 6-set to a 4-set). In general, this is the presentation used by linguists:

$$\left[\begin{array}{l} \text{MORPH} \left[\begin{array}{l} \text{STEM} \\ \text{ENDING } \{_{\$1} \text{ "e", "st", "t", "n"}\} \end{array} \right] \\ \\ \text{SYN|LOCAL|HEAD|AGR} \left[\begin{array}{l} \text{PER } \{_{\$1} \text{ 1ST, 2ND}, \{_{\$2} \text{ 3RD, 2ND}\}, \{\text{1ST, 3RD}\}\} \\ \text{NUM } \{_{\$1} \text{ SG, SG}, \{_{\$2} \text{ SG, PL}\}, \text{PL}\} \end{array} \right] \end{array} \right]$$

Which representation is preferable? A feature-based lexicon need not distinguish between these two representations; indeed, they are provably equivalent within the formalism, even if they illustrate two distinct styles in representing paradigms.

This assertion of equivalence concerns the properties of the feature structures only. The two representations are distinct in their default inheritance properties (which are thus in a sense intensional), so

12

that overwriting defaults from the one or the other structure would be different. But the representations describe the same structures.[11]

3.5 Complex Paradigmatic Elements

In traditional treatments of the paradigm, the passive is normally listed as one of the elements in the paradigm. But passive and active forms differ not merely in the assignment of agreement features (of fairly simple structure) but also in the subcategorization classes to which they belong, and these are quite complex when expressed as feature structures, involving lists of feature structures. The representation of passive as an inflectional variant is something of a challenge for lexical representation schemes.

It is therefore worth noting that even paradigmatic elements as complex as passive may be described as inflectional variants using distributed disjunctions in the way just sketched. That is, we may describe not only the covariation between inflectional affix and a syntactic feature [PASSIVE ±], but even the effect of passive, e.g., on an HPSG subcategorization feature. We hasten to add that feature-based lexicons are not forced to this point of view—one could likewise formulate a derivational rule of passive (cf. Section 4), nor do we wish to advocate the inflectional analysis at this time.[12] But there are clearer examples of languages with passives which are paradigm elements, e.g. Latin (cf. Allen and Greenough [1]), and our primary goal is to establish that this (traditional) analysis is formulatable in a feature-based lexicon. It seems impossible to formulate in other structured lexicons.

$$
\left[
\begin{array}{l}
\text{MORPH} \left[
\begin{array}{l}
\text{STEM} \\
\text{PREFIX } \{_{\$1} \text{ ``'', ``ge''}\} \\
\text{ENDING } \{_{\$1} \{_{\$2} \text{ ``e'', ..., ``n''}\}, \text{``n''}\}
\end{array}
\right] \\[2em]
\text{SYN|LOCAL} \left[
\begin{array}{l}
\text{SUBCAT } \{_{\$1} \langle \text{NP[ACC]}_{\boxed{4}}, \text{NP[NOM]}_{\boxed{3}}\rangle, \\
\qquad\qquad\quad \langle (\text{PP[VON]}_{\boxed{3}},)\text{NP[NOM]}_{\boxed{4}}\rangle \} \\[1.5em]
\text{HEAD|AGR } \left\{_{\$2} \left[\begin{array}{l} \text{PER 1ST} \\ \text{NUM SG} \end{array}\right], ..., \left[\begin{array}{l} \text{PER 3RD} \\ \text{NUM PL} \end{array}\right]\right\}
\end{array}
\right] \\[3em]
\text{SEM} \left[
\begin{array}{l}
\text{PRED} \\
\text{SOURCE } \boxed{3} \\
\text{THEME } \boxed{4}
\end{array}
\right]
\end{array}
\right]
$$

where NP[ACC] abbreviates $[np, \text{CASE ACC}]$, etc. The structure above provides for two alternate (sets of) forms, active and passive.[13] This alternation is indicated by the disjunction name '$1'. The second, passive alternative correlates with a subcategorzation in which, e.g., an optional PP[von] phrase fills the same semantic argument slot as the active subject. The first, active alternative in $1 is simply the active paradigm sketched above.[14]

The version of passive shown here is the version typically used to demonstrate grammatical analysis, i.e. passive is taken to be a variant of transtive verbs only, and no connection between passive and perfect participle is noted, even though these are never distinguished in form. Both of these flaws may be remedied, by what amount to essentially disjunctive specifications. For example, we may allow that the second alternative in $1 above correspond either to the passive syntactic structure (given) or to a perfect participle. It is worth clarifying that we do not claim that anything particularly insightful is gained by a disjunctive analysis of this sort; for insight we need some more detailed work in linguistic analysis. It is nonetheless worth noting the possiblity of the further (disjunctive) detail, given the program of specifying all inflectional variants of a given lexeme as alternative further specifications of the lexeme. These further disjunctive and perhaps uninsightful specifications do not stand in the way of realizing this program.

---

[11]While it is pleasant to obtain the intuitively correct result, it also means that there is no way to distinguish the structures described. Any putative distinction related to the two types of representation would have to be explained in another fashion.

[12]Cf. Kathol [28] for a third, perhaps most interesting analysis, under which a single participial form serves in both active and passive voices, so that passive is neither an inflection, nor a derivational alternation.

[13]But it is clear that a description such as the one above cannot hold of the Latin DEPONENT forms—those with passive inflection, "active" meaning, and no active counterpart (*loquor*).

[14]It would be useful to find a way to allow the active paradigm not merely to be reused, but actually to be inherited in cases such as this (Latin). We have not seen a way of specifying this, since it amounts to specifying that a value be inherited as one of several disjuncts "to be further specified".

3.6 Alternative Frameworks

We discussed above why we prefer NOT to analyze inflectional variation as an employment of lexical rules (but cf. Section 4 for a proposal for representing in feature structures relations which linguists normally designate as 'lexical rules'). There are two similar proposals in inheritance-based, computational lexicons for analysing inflectional variation as the further specification of an abstract lexeme. We discuss these in the present section.

DATR (cf. Evans and Gazdar [16]) is a graph description language (and inference engine) which can encode information about lexical items. DATR provides a formally clean version of the sort of default inheritance first advocated in de Smedt [45] and Flickinger et al. [19], and as such represents a significant advance in the understanding of default lexical inheritance. Although DATR superficially resembles feature notation, its semantics are in some ways different, so that an interface between DATR and a feature system is required, if DATR is to be used as a lexicon in combination with a feature-based parser (or morphological processor)—in keeping with the intention of DATR's developers. The fundamental advantage of our approach over that of DATR is that no such interface is required: lexical structures are exactly the structures required in syntax (and morphology).[15]

A further advantage accrues to feature-based approaches, because they come with a logic which has proven useful in linguistic description. Cf., for example, the extensive employment of disjunction above. Disjunction could probably be added to a system such as DATR, but it does not seem that anyone is trying to do this. A more subtle point is the status of coreference, a notion which feature logics are designed to treat in a very particular way, but which seems inexpressible in DATR.[16]

A final point of divergence is the degree of complexity which an inflectional specification may be allowed to have. The DATR scheme is to use specifications of the form:

$$< \text{path} >==< \text{value} >$$

to encode dependencies between properties and exponents, e.g.

$$< \text{present first plural} >== \text{ "en"}$$

and this works fine as long as the properties involved fit neatly along a simple path. If properties become complex, on the other hand, as they seem to in the case of the (at least the Latin) passive, then this scheme breaks down. The feature-based model using ditributed disjunctions is freer: dependent properties may be distributed in various positions in a feature structure.

The ELU lexicon (Russell et al., [40]) uses equations describing feature structures for the most part, but the equations are divided into a "main set" and a "variant set", the latter of which (monotonically) describe inflectional variants. The paradigm is then described using a set of implicational equations, expressing information such as "if $x =$ [pers: 3, num: sg], then $y =$'s' ".[17] The use of distributed disjunctions seems to be more concise method of specifying inflectional alternatives, but the underlying logic of the ELU approach is very similar to ours.[18]

4 Derivation

Looking at work done in the area of word formation, various formalisms and theories can be ordered according to different dimensions. One possible dimension of classification is, for instance, the distinction

---

[15]For a rather bleak view of prospects for interfaces between DATR and feature formalisms, cf. the proposal in Kilbury et al. [31].

[16]What makes this point subtle is that DATR does allow the expression of "path equivalences", e.g., a statement such as $<$ agr $>==<$ subject agr $>$. Thus, if $<$ subject agr person $>=$ first AND $<$ subject agr number $>=$ sg then the same values will accrue to agr. But the relation is not symmetric; assignment of values to agr either override or are overridden by the above. (DATR's '==' is like the assignment operator in imperative programming languages, not like the identity relation.) A further refinement is that, even in the direction in which this does work (like identity) no distinction is made between two paths which have the same values and those which have distinct, but equivalent values. But such "structure-sharing" is very widely exploited in feature-based linguistic analyses.

[17]Information in variant sets is not subject to default overwriting, for reasons which are not explained. Although we find the proposal that some information not be subject to default overwriting congenial, it would seem desirable to view the specification of alternations as an orthogonal point.

[18]We have also benefited from the opportunity to examine unpublished work of Andreas Kathol at The Ohio State University, who has independently developed a similar treatment of inflectional variation—this one based on constrained relations.

between procedural vs. declarative formulation of rules relating phonological, morphological and orthographic phenomena (cf. Calder, [9]).[19] Classifying specific treatments is subjective. Most linguists will interpret lexical rules procedurally. But lexical rules can also be regarded purely declaratively, even if the procedural view is the most prominent one.[20]

4.1 External vs. Internal Lexical Rules

Instead of treating the issue of declarative vs. procedural formulations, we want to turn our attention to another dimension of classification: *Where does word formation take place—within or without the lexicon?* WITHOUT means that the form of lexical rules is different from the structure of lexical entries (lexemes, also possibly morphemes). Lexical rules in PATR-II [44] or D-PATR [26], e.g., look like feature structures and are represented via a collection of path equations, but their interpretation is completely different from that of (normal) feature structures. The same is true, if we move to other theories: *f-structures* differ in form (syntax) and interpretation (semantics) from lexical rules stated in LFG (cf. the articles in Bresnan, [8]). This same observation holds for HPSG [36], Ch. 8.2,[21] for the Alvey tools project [38], for the early days of HPSG [19], for the work of Flickinger [17], and also for Hoeksema's Categorial Morphology [23].

By its nature, an EXTERNAL lexical rule sets up a relation between two lexemes (or classes of lexemes)—or, in the case of feature-based theories, between two feature structure descriptions. But specifying the exact meaning of this mapping is an open question—nearly all theories have different viewpoints when interpreting (external) lexical rules:

- Are external lexical rules functions or perhaps even relations?

- If functions, do they take one argument or arbitrarily many? (*Mutatis mutandis* for relations)

- Are they unidirectional or bidirectional?

- If unidirectional, will they be interpreted declaratively (AVM_1 implies a corresponding AVM_2) or procedurally (lexical rule as an instruction, to build AVM_2 out of AVM_1)?

Instead of treating external lexical rules further, we'd like to propose a novel interpretation of lexical rules, which we call INTERNAL. An internal lexical rule is an information-bearing object, indistinguishable in its form and meaning from other entries of the lexicon—strictly speaking, it just IS a lexical entry.

Derivational rules will be modeled as feature structure descriptions, just as lexical entries are. This is aesthetically pleasing, and has the further advantage of formal clarity. Perhaps most importantly, however, the fact that lexical rules and lexical entries are of the same formal type allows one to liberate yet another level of linguistic structure from PROCEDURAL considerations and therefore allows one to interleave, e.g., morphological and phrasal processing in a way that is otherwise prohibited.[22]

It is worth noting that there is no one-to-one correspondence between traditional lexical rules (what linguists have called lexical rules) and single objects (feature structures) in our approach. Rather, the information in a lexical rule is distributed among lexical entries, principles, and morphological dominance schemata—our closest analogue to "rule".

In general, internal (or external) lexical rules cannot be realized as independent lexemes (or morphemes); instead rules serve as FILTERS (in the sense of well-formedness conditions), used to rule out

---

[19]Our briefly sketched treatment of derivation (and inflection) in the introductory section is in its essence declarative, because linguistic knowledge is encoded in terms of feature structures only and unification is the sole information-building operation.

[20]Interestingly to note, Pollard & Sag, [36], pp. 209, suggest a third interpretation of lexical rules coming directly from the field of many-sorted abstract data types—an ALGEBRAIC perspective on lexical rules.

[21]Feature structures descriptions and lexical rules (form: $AVM_1 \longmapsto AVM_2$) in HPSG have nothing in common, because they differ in form as well as in interpretation. This remark is supported by the following observation: feature structures in HPSG are always typed, and these types can be ordered (partially) via subsumption. But this isn't true for lexical rules. A lexical rule as a whole does not have a type and there's no way to relate it to other feature structures. Under the assumption that a lexical rule can in principle be typed and resides in the lexicon, this type ought to be a subtype of *lexical-sign* according to Pollard & Sag [36] and ought to have exactly the three top-level attributes PHON, SYN, and SEM—but this isn't the case.

[22]One can quibble about our choice of terminology here. Given the possibility of interleaving processing in the way we describe, it might seem as if what we are calling INTERNAL lexical rules are rather more external than other construals would have it. Or one can insist on the standard construal of derivation as a mapping from lexical entries to lexical entries, which implies that the term INTERNAL lexical rule is misleading, since this insistence effectively equates the notion 'lexical rule' with that of an external mapping, always taking a set of feature structures and yielding a feature structure. Our own preference for the term 'internal lexical rule' arises because we see the lexicon in general as constituted by a set of feature structure descriptions—with no fundamental distinction between lexical rules and lexical entries. Lexical rules are simply a particular kind of feature structure description.

ill-formed structures (to fail to parse or generate them). A treatment of this kind will be presented here for the field of DERIVATION. Another approach, which is also distinguished by its use of internal lexical rules, can be found in the ELU system [40].[23]

4.2 Our Treatment of Derivation

In HPSG-I linguistic knowledge about word formation is encoded through a family of lexical rules, which are not feature structures, but rather essentially external operators working on feature structures. This (for us) unsatisfactory view appears even more questionable given the view of most linguists that form and meaning are much harder to describe for sentences and phrases than for words. If this is the case, one may ask, why does HPSG treat word formation via external lexical rules rather than in a purely feature-based way? Why not formulate RULES and PRINCIPLES for word grammar similar to those stated by Pollard & Sag [36] for phrasal and sentential grammar? We think, there are at least three replies this question might provoke:

1. HPSG is a theory capable only of capturing the form and meaning of sentences in feature structure descriptions, but incapable of describing morphotactical aspects of language.

2. Trying to handle lexical structures (morphotactics) in terms of feature structure descriptions (i.e., via rules and principles) only leads to inefficient implementations.

3. HPSG is a conglomerate of different formalisms and theories (cf. [42], p. 1, and [36], p. 1), saying little or nothing about morphotactics. In the theories from which HPSG borrows, morphotactics were stated in form of external lexical rules. HPSG assumed this view somewhat nonreflectively, because HPSG's primary purpose is the description of phrasal and sentential syntax and semantics.

We're convinced that the first thesis is simply WRONG, and that the second one is (at the moment) probably true (because procedural implementations of lexical rules can be very efficient), while the third statement is definitely correct. Summing up, we think it's a promising task to approach DERIVATION purely in terms of feature structure descriptions—just in the spirit of HPSG. Recall the two equations in [36], p. 147,

$$(2) \qquad\qquad UG = P_1 \sqcap ... \sqcap P_n$$

$$(3) \qquad\qquad English = P_1 \sqcap ... \sqcap P_{n+m} \sqcap (L_1 \sqcup ... \sqcup L_p \sqcup R_1 \sqcup ... \sqcup R_q)$$

These may be understood in the following way: universal grammar (UG) consists of a set of principles $P_1, ..., P_n$, whose conjunction (or unification) must hold true of every structure in every language. In addition, a given language may impose the language-specific constraints $P_{n+1}, ..., P_{n+m}$. Finally, the grammar requires that every structure instantiate some lexical entry $L_1, ..., L_p$, or phrasal pattern (rule) $R_1, ..., R_q$—since a structure need satisfy only one of the lexical or rule descriptions in order to be a well-formed phrase, these constraints obtain disjunctively. A language is then just the set of structures which simultaneously conform not only to all the principles, both universal and language-specific, but also to at least one of the lexical or phrasal descriptions. These fundamental equations define an HPSG grammatical theory for phrases and sentences, and we propose to apply a similar methodology to derivation, relying extensively on RULES, PRINCIPLES, and unification-based INHERITANCE (for an explanation of (2) and (3), cf. Pollard & Sag, [36], p. 147).

In contrast to inflection, derivation cannot rely on NAIVE inheritance alone. Here, 'naive' means that a word like the German *weglaufen* is defined by inheriting (unifying) all the properties from the prefix *weg-*, the verb *laufen*, plus additional idiosyncratic properties of the new complex lexeme, i.e.,

$$(4) \qquad\qquad weglaufen = [\,weg\,] \sqcap [\,laufen\,] \sqcap [\,.\,.\,.\,.\,.\,.\,].$$

ELU's treatment of derivation (cf. [40], p. 218) is done in such a way, and it may be a reasonable tack to take for the treatment at hand, that of the German separable prefixes. Applied generally, an approach like this leads to several insurmountable problems:
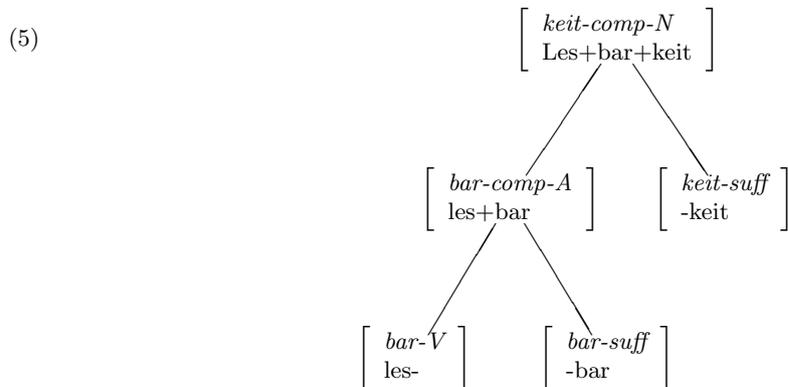
- If we relied on naive inheritance as the (sole) descriptive means, it would seem impossible to explain how the iteration of derivational processes could ever lead to different results. If *anti-* (or take the German *vor-*) is a derivational prefix, and its effect on a stem is described via inheritance, then the

---

[23]ELU treats inflection as well as derivation by means of pure inheritance. We are convinced, however, that this approach is not strong enough for derivation (cf. below).

effect of inheriting it should be the same, whether there are one, two, or more instances of the SAME prefix in a word because unification is IDEMPOTENT and inheritance defines itself through unification. Thus a complex word like *anti-missile* (or *Vor+version*) would be predicted to be the same as *anti-anti-missile* (or *Vor+vor+version*).[24] Likewise, such an approach is not capable of explaining the INDIRECT recursion occurring in complex compounds such as *institu+$\underline{t}$ion+al+isa+$\underline{t}$ion*.

- Sole reliance on naive inheritance leaves little opportunity to explain the hierarchical structure often found in morphology, e.g., the difference in bracketing one finds in complex words containing at least two affixes, e.g., *[un- [do -able]]* as opposed to *[[un- do] -able]*. Because inheritance is associative and monotonic (in the absence of overwriting), other mechanisms must be at play. Naive inheritance seems incapable of accounting for any structure, let alone ambiguous hierarchical structure.

- Simple examination of derivational results suggests that treating all of them via naive inheritance from a single lexeme will lead to unwieldy lexicons: a form such as German *Ableit+bar+keit* (derivability) would seem to require that verbal, adjectival, and nominal paradigms be found as heirs of the single lexeme (recall that we dealt with this above by modeling it via mapping from lexeme to lexeme).

- It turns out also that there are technical problems connected with the treatment of derivation as inheritance. These may be summarized, albeit cryptically, in the following way: we should prefer that the result of a category-changing derivational process, e.g., the process which derives *derive+able* from *derive* and *-able*, be a full-fledged member of the target category (of the derivational process)—in this case, the class of adjectives. Now, if the derivational process is modeled by naive inheritance only, then *derive+able* ought to inherit from the class of verbs (through *derive*), as well. It is easy to continue this line of reasoning further (consider *derive+abil+ity*) to see how this sort of explanation leads one to the postulation of lexemes of dubious lineage, inheriting from too many ancestors (this point is essentially just the converse of the last).

Treating DERIVATION in our approach will lead to complex morphs (e.g., words) consisting of a head daughter HEAD-MORPH and a complement daughter COMP-MORPH (e.g. (5)). The task of the 'morphological-daughters' feature is to encode morphological structure, similarly to how HEAD-DTR and COMP-DTRS do this on the phrasal level (cf. [36]). This is in analogy to the HPSG formulation of phrase structure in features, yielding tree structures.[25]

(5)

$$
\begin{bmatrix} keit\text{-}comp\text{-}N \\ \text{Les+bar+keit} \end{bmatrix}
$$

$$
\begin{bmatrix} bar\text{-}comp\text{-}A \\ \text{les+bar} \end{bmatrix} \qquad \begin{bmatrix} keit\text{-}suff \\ \text{-keit} \end{bmatrix}
$$

$$
\begin{bmatrix} bar\text{-}V \\ \text{les-} \end{bmatrix} \qquad \begin{bmatrix} bar\text{-}suff \\ \text{-bar} \end{bmatrix}
$$

We include this as an example of the hierarchical structure whose analysis is beyond the descriptive reaches of NAIVE inheritance. The objections to the description of derivation in terms of naive inheritance do not apply here, since, e.g., tree adjunction is not idempotent—so that, e.g., *Vorversion* may be distinguished from *Vorvorversion*; tree adjunction generates hierarchical structures (evident here), and, as we shall see, it distinguishes inheritance (sharing properties) from the requirements that sublexemes come from particular word classes or types (so that the tree structure above cannot be interpreted to mean that the

---

[24]Permitting iteration of derivational prefixes only to a certain depth (which seems *prima facie* plausible since, e.g., words such as German *Vor+vor+vor+version* are questionable), will solve this problem, if every element of the finite set of complex prefixes is coded as a lexical entry. But this attempt at repair is (i) extremely unsatisfying theoretically and (ii) incomplete, because the depth of composition is a subjective measure.

[25]BINARY trees (together possibly with unary ones for ∅ derivation) seem to suffice, at least for derivation. Of course, this is an assumption that will be put to the test in applications of this lexical work.

noun *Lesbarkeit* is in any sense a verb of the same type as *lesen* or an adjective of the same type as *lesbar*). The hierarchy here is a PART-OF hierarchy in contrast to the inheritance hierarchy, which constitutes an IS-A hierarchy. The distinction is crucial: the parts of a complex word do not bequeath their properties to the words derived from them.

In general, the head daughter is a bound morpheme of type *affix*, while the complement daughter is of type *word* and is free (cf. Fig. 2). HEAD-MORPH and COMP-MORPH are put together under the label MORPHS (cf. DTRS). With these assertions in mind, we postulate, in analogy to the phrasal 'rules' in HPSG (cf. [36], pp. 149), the following morphological rule schema:

$$(6) \qquad \text{[LEX +]} \; \longrightarrow \; \text{H, C[LEX +]}$$

or more formally as a typed feature structure ('≡' is used for definitional expressions):

$$(7) \qquad \text{MHCR} \; \equiv \; \begin{bmatrix} complex \\ \text{SYN|LOC|LEX +} \\ \text{MORPHS} \begin{bmatrix} morph\text{-}head\text{-}struct \\ \text{HEAD-MORPH } [\mathit{affix}] \\ \text{COMP-MORPH } [\mathit{part\text{-}of\text{-}speech}] \end{bmatrix} \end{bmatrix}$$

Just as Pollard & Sag proclaim universal as well as language-specific principles, we will define four 'principles', which are consistent with our linguistic data and specified as typed implications. We don't suppose that these principles will not be overturned by wider ranges of analyses, but we do suppose that they illustrate how the HPSG style of analysis can be extended to word-internal structure. The formulation of the principles presupposes that the underlying feature logic (along the lines of Kasper & Rounds [27], [39]; cf. the section on feature structures and HPSG) is extended by adding FUNCTIONAL DEPENDENCIES (functionally dependent values; for a motivation, cf. Pollard & Sag [36], pp. 48-49; for a formal definition, cf. Reape [37], pp. 73ff).

All morphological HEAD features, as well as (morphological) SUBCATEGORIZATION will be defined, for simplicity, under the path SYN|LOC.[26] PHON will be replaced by MORPH|FORM and *headed-structure* is replaced by *morph-head-struct* (morphologically headed structure) which has at least the attributes HEAD-MORPH and COMP-MORPH. The symbol *complex* (complex word) corresponds to the TYPE of the same name in the subsumption lattice (cf. Fig. 2). The CONSTITUENT ORDER PRINCIPLE ([36], pp. 169) was taken over directly from HPSG (MCOP).[27]

$$(8) \qquad \text{MCOP} \; \equiv \; \left\{ \begin{array}{l} \begin{bmatrix} complex \\ \text{MORPHS } [morph\text{-}head\text{-}struct] \end{bmatrix} \; \Longrightarrow \\ \begin{bmatrix} complex \\ \text{MORPH|FORM \; order-constituents}(\boxed{1}) \\ \text{MORPHS } \boxed{1} \end{bmatrix} \end{array} \right.$$

Likewise for derivation, the formulation of the HEAD FEATURE PRINCIPLE in HPSG ([36], pp. 58) is used directly (MHFP), and only certain attributes and type names were altered. Among other things, MHFP is responsible for deducing the category of the new word from the category of the head daughter.

$$(9) \qquad \text{MHFP} \; \equiv \; \left\{ \begin{array}{l} \begin{bmatrix} complex \\ \text{MORPHS } [morph\text{-}head\text{-}struct] \end{bmatrix} \; \Longrightarrow \\ \begin{bmatrix} complex \\ \text{SYN|LOC|HEAD } \boxed{1} \\ \text{MORPHS|HEAD-MORPH|SYN|LOC|HEAD } \boxed{1} \end{bmatrix} \end{array} \right.$$

[26]There are good reasons to introduce, at least for affixes, an (additional) morphological subcategorization feature under the path MORPH, but we shall not pursue this here. This approach is investigated in Krieger [33].

[27]The function order-constituents in MCOP has to be sensitive to the type of its argument. If the argument is of type *morph-head-struct* (see below), the function is being applied in derivation instead of working on the sentence level. Thus, in contrast to the version of order-constituents in [36], our version of order-constituent is overloaded with respect to its argument—we employ an AD HOC POLYMORPHISM. Cf. Cardelli & Wegner [10]. Alternatively, we could specify a second function, order-morph-constituents.

The SEMANTICS PRINCIPLE may be taken in its simplest form and slightly modified ([36], pp. 99): the semantics of the mother is equal to the semantics of the head daughter.

$$(10) \qquad \text{MSP} \equiv \left\{ \begin{array}{l} \begin{bmatrix} complex \\ \text{MORPHS} \;\; [\,morph\text{-}head\text{-}struct\,] \end{bmatrix} \implies \\[4pt] \begin{bmatrix} complex \\ \text{SEM} \;\; \boxed{1} \\ \text{MORPHS|HEAD-MORPH|SEM} \;\; \boxed{1} \end{bmatrix} \end{array} \right.$$

The use of binary trees and the head-complement structure for derivation leads to a SUBCATE-GORIZATION PRINCIPLE which looks (and is) completely different from the one proposed in HPSG-I for phrases ([36], pp. 71). Identifying the values of MORPHS|COMP-MORPH and MORPHS|HEAD-MORPH|SYN|LOC|SUBCAT in MSCP (cf. (11)) guarantees that the head takes the right complement and binds it. In addition, the function construct-subcat assembles the subcategorization information of the new morphological phrase. The elements of the subcategorization list under path SYN|LOC|SUBCAT are essentially those of the complement—however, construct-subcat regroups them, perhaps omitting some of them (cf. examples). The result of this modification depends not only on the type of the attribute MORPHS|HEAD-MORPH, but also on the sort of the complement morpheme:

$$(11) \qquad \text{MSCP} \equiv \left\{ \begin{array}{l} \begin{bmatrix} complex \\ \text{MORPHS} \;\; [\,morph\text{-}head\text{-}struct\,] \end{bmatrix} \implies \\[4pt] \begin{bmatrix} complex \\ \text{SYN|LOC|SUBCAT} \;\; \text{construct-subcat}(\boxed{1}) \\ \text{MORPHS} \;\; \boxed{1} \begin{bmatrix} \text{HEAD-MORPH|SYN|LOC|SUBCAT} \;\; \boxed{2} \\ \text{COMP-MORPH} \;\; \boxed{2} \end{bmatrix} \end{bmatrix} \end{array} \right.$$

Although Pollard & Sag [36] strictly type the attributes of feature structures in general, they do not explicitly state that PRINCIPLES as well as RULES may also be regarded as types. But we may interpret them as types which have to satisfy the SUBSUMPTION relation only.[28] In taking this step, one has to integrate them consistently into the subsumption lattice (cf. Fig. 2).

With respect to equation (3), we extend the set of principles and the set of rules by adding MCOP, MHFP, MSP, MSCP, and MHCR. Finally, in typing the antecedents (of the implications), we must take care, since not every principle can be combined with every rule or lexical entry.

Because only morphological head-complement structures are examined in this paper, equation (3) allows us to unify the types associated with (the right-hand sides of) MCOP, MHFP, MSP, MSCP, and MHCR (call the result HCR&Ps), and to regard this feature structure as a restriction for all feature structures belonging to this new (conjunctive) type. All complex morphs (morphs having the attribute MORPHS) must satisfy this type restriction, i.e., must be of the type *HCR&Ps*.

$$(12) \qquad HCR\&Ps \;=\; MHFP \sqcap MSCP \sqcap MSP \sqcap MCOP \sqcap MHCR$$

$$(13) \qquad \text{HCR\&Ps} \equiv \begin{bmatrix} \text{MORPH|FORM} \;\; \text{order-constituents}(\boxed{1}) \\ \text{SYN|LOC} \begin{bmatrix} \text{LEX} \;\; + \\ \text{SUBCAT} \;\; \text{construct-subcat}(\boxed{1}) \\ \text{HEAD} \;\; \boxed{2} \end{bmatrix} \\ \text{SEM} \;\; \boxed{3} \\ \text{MORPHS} \;\; \boxed{1} \begin{bmatrix} \text{HEAD-MORPH} \begin{bmatrix} \text{SYN|LOC} \begin{bmatrix} \text{SUBCAT} \;\; \boxed{4} \\ \text{HEAD} \;\; \boxed{2} \end{bmatrix} \\ \text{SEM} \;\; \boxed{3} \end{bmatrix} \\ \text{COMP-MORPH} \;\; \boxed{4} \end{bmatrix} \end{bmatrix}$$

Trying to encode rules and principles explicitly as elements of a type subsumption lattice (inheritance network) along the lines of HPSG ([36], Ch. 8), requires a REWRITING step: Because of their implicative

---

[28]Principles constrain existing types and so must be interpreted as supertypes. In translating a principle—usually expressed as a conditional—into a type, we only use the the right side of the conditional, the consequent. For a motivation, see below.

nature, we cannot state principles DIRECTLY as types—we must rewrite them. 'Rewriting' means first, that only the right side (the consequent) of an implication will be regarded as a type. Second, in order to obtain the force of the antecedent, the type associated with the conjunctive feature structure representing the consequent has to be integrated into the 'right' position in the lattice (cf. *HCR&Ps* in Fig. 2), where 'right' is determined by *taking care that the subsumption relation holds*.[29] Even an equation like (3), containing lots of implications, can then be compiled to form an inheritance hierarchy, consisting only of conjunctive feature types.[30]

The idea of reducing implications to conjunctive types will lead us directly to the structure of the (type/class) subsumption lattice (cf. Fig. 2). Notice that, regarding the laws of feature algebras, we're allowed to 'multiply out' information stored in certain normal forms. This additional step of transformation is necessary to construct hierarchies like those one shown in Fig. 2 and 4.

In the following, we will further motivate and exemplify our approach to derivation by applying it to examples of (morphological) SUFFIXATION and PREFIXATION.

4.3 *-bar* and *-keit* Suffixation

The treatment of German *-bar* (and also of *-keit*) suffixation is interesting from different points of view and presents severe problems, which can, however, be adequately solved in our approach:

sporadic applicability *-bar* suffixes many verbs, but not all.

partial regularity Many *-bar* derivatives have regular forms and irregular semantics.

category change *-bar* suffixation changes (syntactic) category: `Verb` $\leadsto$ `Adjective`.

subcategorization change The subcategorization list of `Verb+`*bar* changes: the arity is that of the verb minus 1; the semantic argument positions in the scope of the *-bar* semantics, on the other hand, do NOT change.

Starting with a verb like the German *lesen* (to read), where *-bar* suffixation is regular, we may construct a possible lexicon entry with respect to the inheritance network of Fig. 2.

$$(14) \qquad lesen \equiv \begin{bmatrix} bar\text{-}V \\ \text{MORPH} \quad \begin{bmatrix} \text{STEM} \ \text{``les''} \\ \text{PARADIGM} \ [\,\dots\,] \end{bmatrix} \\ \text{SYN|LOC} \quad \begin{bmatrix} \text{SUBCAT} \ < \ \dots, (\text{NP}_{\boxed{2}}), \ \text{NP}_{\boxed{1}} > \\ \text{LEX} \ + \\ \text{HEAD|MAJ} \ \text{V} \end{bmatrix} \\ \text{SEM} \quad \begin{bmatrix} \text{RELN} \ read' \\ \text{SOURCE} \ \boxed{1} \\ \text{THEME} \ \boxed{2} \end{bmatrix} \end{bmatrix}$$

Notice that although *lesen* is syntactically classified as a verb (this is the import of the feature specification [SYN|LOC|HEAD|MAJ   V]), more specifically, it is an instance of the class *bar-V* (verbs that may combine with *-bar*). Note also that we employ here the LEXEME *lesen* rather than, e.g., the infinitive in *lesen*'s paradigm—this is compatible with the fact that only the stem *les-* is found in the derived word.

Moving now to *-bar*, we regard *-bar* (cf. (15)) as the HEAD of the morphological complex with category ADJECTIVE (A); it may function as a head even though it fails to appear as a FREE word. Instead, it occurs only as a BOUND morpheme (instance of the class *bar-suff*; cf. Fig. 2). As a result of the HEAD FEATURE PRINCIPLE the mother obtains automatically the category of the head daughter—and this is exactly what we want, since *les+bar* (*readable*) is an adjective. The head-complement rule, the subcategorization principle and the specification of SYN|LOC|SUBCAT to be an (underspecified) instance

---

[29]In general, there's only ONE right position—*the most general position at which the subsumption relation holds.* But this is only true, if we assume a (subsumption) lattice where subsumption is STRICT, i.e., where it is not possible to have two different types standing in a subsumption relation, even though their denotation is the same.

[30]The rewriting step is subtle in that it moves information from object-language implicational statements into restrictions in the type hierarchy which forms the skeleton of the interpretation. On the one hand, because of general laws of interpretation for feature logics, we have the following inference for *Ante*, *Conseq* feature structure terms: from the principle $Ante \Rightarrow Conseq$, we know that $[\![Conseq]\!] \supseteq [\![Ante]\!]$. On the other hand, the principles always ADD information to a feature structure description to which they are applied, so that *Ante* always subsumes *Conseq*, i.e., $[\![Ante]\!] \supseteq [\![Conseq]\!]$. This leads to an effective identification of *Ante* and *Conseq* which is realized in the type hierarchy.
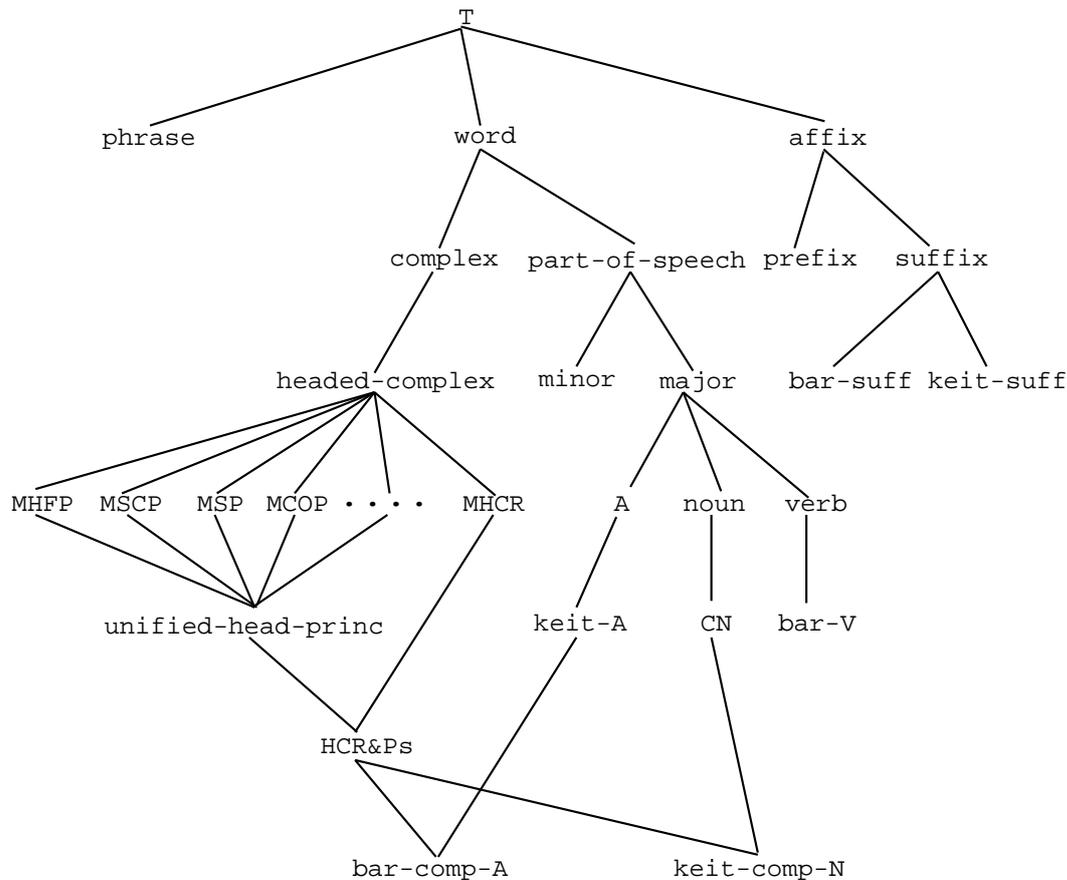
Figure 2: Structure of the inheritance network in case of *-bar* and *-keit* suffixation, including morphological principles and rules. Note that we additionally impose LOCAL constraints on certain classes, especially on *bar-comp-A* and *keit-comp-N*; for motivation, see text. Note further that, although the class of adjectives formed using *-bar* inherits from *A* (adjective) and from *HCR&Ps*, it does NOT inherit from either of its component morphs—*bar-V* or *bar-suffix*.

of *bar-V* additionally guarantee that *-bar* only combines with *-bar* verbs. Note too, that the value of the attribute LEX in (15) is UNSPECIFIED.[31]

Semantically, *-bar* functions as a modal operator, working on the propositional semantics of *lesen* 'read' to create a proposition asserting the possibility of reading. We note here the co-specification between the semantics of the subcategorized element and the value of the SCOPE attribute in the modal proposition. These assumptions lead us to postulate the following structure for *-bar*:

$$
(15) \qquad \text{bar} \equiv
\begin{bmatrix}
\textit{bar-suff} \\
\text{MORPH|FORM ``bar''} \\[4pt]
\text{SYN|LOC}
\begin{bmatrix}
\text{LEX} \\
\text{HEAD|MAJ A} \\
\text{SUBCAT bar-V}_{\boxed{1}}
\end{bmatrix} \\[12pt]
\text{SEM}
\begin{bmatrix}
\text{OPERATOR } \diamond \\
\text{SCOPE } \boxed{1}
\end{bmatrix}
\end{bmatrix}
$$

The entries for *lesen* and *-bar* together with the head-complement rule and the morphological principles

---

[31] Under the assumption of Carpenter's "total well-typing" [12], it may be useful to drop the attribute LEX in (15).

permit us therefore to construct a well-formed feature structure for *lesbar*, and also to reject ill-formed feature structures, so that we can show that (16) is the predicted structure. This meshing of mechanisms ensures that *lesbar* has the right internal structure. The function order-constituents, for instance, determines on the basis of the values of HEAD|MORPH and COMP|MORPH (more exactly, on the basis of the types restricting the attributes) that it has to perform a concatenation (&) (cf. 16). Additionally, the semantics principle is responsible for the semantics of *lesbar* coming directly from the head daughter *-bar*, while *-bar* takes the complete semantics of *lesen* to fill its attribute SCOPE.

$$
\text{lesbar} \equiv
$$

$$
(16) \quad
\begin{bmatrix}
\textit{bar-comp-A} \\
\text{MORPH|FORM } \boxed{1} \& \boxed{2} \\
\text{SYN|LOC } \begin{bmatrix} \text{LEX } + \\ \text{SUBCAT construct-subcat(} \boxed{5} \text{)} \\ \text{HEAD } \boxed{3} \end{bmatrix} \\
\text{SEM } \boxed{4} \\
\text{MORPHS } \boxed{5} \begin{bmatrix}
\text{HEAD-MORPH} \begin{bmatrix} \text{MORPH|FORM } \boxed{2} \text{ "bar"} \\ \text{SYN|LOC } \begin{bmatrix} \text{SUBCAT } \boxed{6}[\ ]\boxed{7} \\ \text{HEAD } \boxed{3} [\text{ MAJ A }] \end{bmatrix} \\ \text{SEM } \boxed{4} \begin{bmatrix} \text{OPERATOR } \diamond \\ \text{SCOPE } \boxed{7} \end{bmatrix} \end{bmatrix} \\
\text{COMP-MORPH } \boxed{6} \begin{bmatrix} \text{MORPH|STEM } \boxed{1} \text{ "les"} \\ \text{SYN|LOC|LEX } + \\ \text{SEM } \boxed{7} \end{bmatrix}
\end{bmatrix}
\end{bmatrix}
$$

At this point, we have to clarify the complex interaction between the SUBCATEGORIZATION PRINCIPLE and the SEMANTICS PRINCIPLE. The scope on which *-bar* semantically operates is the semantics of *lesen*. The SOURCE role of *lesen* (cf. (14)), whose value is identical to the semantics of the SUBJECT of *lesen*, won't be filled by a phrase licensed by *lesbar* directly, so that it is possible that the SOURCE role in *lesbar* is unfilled. This occurs when no agentive *von* phrase occurs in construction with the adjective, but the attribute is present even where the value is not. In this case, the SOURCE role contains the semantics of an underspecified NP, and the proposition within which it occurs holds whenever there is some value for the NP semantics for which the proposition holds (cf. Flickinger and Nerbonne [18] for a similar treatment of the semantics of the *for* phrase licensed by *easy* adjectives and Karttunen [26] for an analysis of the semantics of the passive *by* phrase along these lines). The intention behind this approach can be appreciated in a concrete example: the sentence *Das Buch ist lesbar* 'The book is readable' doesn't EXPLICITLY state for whom it is possible to read the book. Instead, the reader is mentioned only IMPLICITLY, so that the filler of SOURCE role of *lesbar* might be suspected from extralingustic context, or from the preceding or following discourse, but is not specified directly by the sentence itself.[32] The subcat list of *lesbar* therefore does not include the subject of *lesen*, at least not as an obligatory complement. The object of *lesen* fills the same role filled by the subject of *lesbar*, of course. This is exactly what the function construct-subcat has to accomplish, taking over all other entries from the subcat list of *lesen* to build the right subcategorization list for *lesbar* (cf. (16)).

We provide a single sketch to suggest the possibility of more involved hierarchical structure. In order to construct the word *Les+bar+keit* (*readability*) out of *lesbar*, we have to specify the entry for the suffix *-keit* and the *keit-suff* class (cf. Fig. 2), i.e., it is necessary to state the idiosyncratic properties of *-keit*.

$$
(17) \quad \text{keit} \equiv
\begin{bmatrix}
\textit{keit-suff} \\
\text{MORPH|FORM "keit"} \\
\text{SYN|LOC } \begin{bmatrix} \text{HEAD|MAJ N} \\ \text{SUBCAT keit-A}\boxed{1} \end{bmatrix} \\
\text{SEM } \begin{bmatrix} \text{VAR } \boxed{2} \\ \text{RESTRICTION } \boxed{1} \begin{bmatrix} \text{PRED } \dots \\ \text{EVENT } \boxed{2} \end{bmatrix} \end{bmatrix}
\end{bmatrix}
$$

---

[32] By contrast, there may be a syntactic binding in examples such as *Ich finde das Buch lesbar* 'I find the book readable'.

By means of the morphological principles and the head-complement rule we may now build an entry for *Lesbarkeit* in the same way shown above (for *lesbar*) with the morphological constituent structure shown in (5).

Among *-bar* verbs such as *lesen*, having perfectly regular *-bar* adjectives (i.e., complex adjectives, containing *-bar* as their head, e.g., *lesbar*), there are others whose derived adjectives are partially irregular, for example with respect to their semantics. As an additional complication, some *-bar* adjectives of these verbs are provided with an additional regular, but non-standard reading. Take for instance the German verb *essen* (to eat):

$$(18) \qquad essen \equiv \begin{bmatrix} bar\text{-}V \\ \text{MORPH|FORM} \quad \text{``essen''} \\ \text{SYN|LOC|SUBCAT} \quad < \ldots, (\text{NP}_{\boxed{2}}), \text{NP}_{\boxed{1}} > \\ \text{SEM} \quad \begin{bmatrix} \text{RELN} \quad eat' \\ \text{SOURCE} \quad \boxed{1} \\ \text{THEME} \quad \boxed{2} \end{bmatrix} \end{bmatrix}$$

The non-standard (semantically regular) reading of *eßbar* can be built in a regular way by means of the mechanisms described above, taking *essen* and *-bar* to form a complex word.

$$(19) \qquad eßbar^{non-stand} \equiv \begin{bmatrix} bar\text{-}comp\text{-}A \\ \text{SEM} \quad \boxed{1} \begin{bmatrix} \text{OPERATOR} \quad \diamond \\ \text{SCOPE} \quad \begin{bmatrix} \text{RELN} \quad eat' \\ \text{SOURCE} \quad \ldots \end{bmatrix} \end{bmatrix} \\ \text{MORPHS|HEAD-MORPH|SEM} \quad \boxed{1} \end{bmatrix}$$

The standard reading of *eßbar* on the other hand is 'edible' (the property of an object which can SAFELY be eaten). Constructing the standard reading (with irregular semantics) for *eßbar* can be done in our approach in two different ways:

1. We do NOT regard *eßbar* as an instance of the class *bar-comp-A*; instead, *eßbar* is entered separately into the lexicon. We then have to specify at least that the semantics of (20) is different from that of (19), although the MORPH and SYN properties seem to remain the same. A treatment of this kind leads us to the question of whether the feature structure (20) actually will have MORPH daughters—since no use need be made of the structure.

2. The semantics of (19) (the entry which was built regularly) is modified by using OVERWRITING, DEFAULT UNIFICATION or other nonmonotonic mechanisms to enforce the standard reading. In this case, *eßbar* (21) belongs to the class *bar-comp-A*, because all other properties remain the same. We would follow Bouma [7] in the use of default unification (as a basis for default inheritance).

$$(20) \qquad eßbar^{stand} \equiv \begin{bmatrix} ?\text{-}A \ \wedge \ \neg bar\text{-}comp\text{-}A \\ \text{SEM} \quad \begin{bmatrix} \text{RELN} \quad safely\text{-}eat' \\ \text{SOURCE} \quad \ldots \end{bmatrix} \\ \text{MORPHS} \quad ??? \end{bmatrix}$$

$$eßbar^{stand'} \equiv eßbar^{non-stand} \oplus \begin{bmatrix} \text{SEM} \quad \begin{bmatrix} \text{RELN} \quad safely\text{-}eat' \\ \text{SOURCE} \quad \ldots \end{bmatrix} \end{bmatrix}$$

$$(21) \qquad = \begin{bmatrix} bar\text{-}comp\text{-}A \\ \text{SEM} \quad \boxed{1} \begin{bmatrix} \text{RELN} \quad safely\text{-}eat' \\ \text{SOURCE} \quad \ldots \end{bmatrix} \\ \text{MORPHS|HEAD-MORPH|SEM} \quad \neg \ \boxed{1} \end{bmatrix}$$

The advantage of the second approach is that regular properties of partially regular derivations need not be specified redundantly, as would be the case in the first approach. The use of default specifications thus obtains the same advantages in DERIVATION that Flickinger et al. [19] and Evans & Gazdar [16] have shown in word-class definitions. Defaults, together with the possibility of overwriting defaults in more specific definitions may turn out to be even more important in connection with the analysis of derivational

relationships, since these are notoriously irregular in morphological form, syntactic feature assignment, and semantics (cf. Toman's book-length study on *-bar* adjectives [47] for ample illustration).

The typed approach to *-bar* suffixation allows us to prevent ill-formed *-bar* adjectives; e.g., we have to rule out the combination of *haben* (to have) together with *-bar*. This is very easy to achieve under the assumption that *haben* doesn't belong to the *-bar* verb class *bar-V*, but instead to another class (say *?-V*), thus preventing *haben* from combining with *-bar*—therefore *hab+bar* is disallowed.

$$(22) \qquad \text{haben} \equiv \left[\begin{array}{l} \text{?-V } \wedge \text{ } \neg\text{bar-V} \\ \ldots\ldots \\ \ldots\ldots \end{array} \right]$$

It is nevertheless possible to construct *handhab+bar* 'manageable' out of *handhaben* 'to handle, manage', since *haben* and *handhaben* are distinct lexemes. By explicitly encoding *handhaben* as an entry of type *bar-V*, we can move to a legal description of *handhabbar*.

$$(23) \qquad \text{handhaben} \equiv \left[\begin{array}{l} \text{bar-V} \\ \ldots\ldots \\ \ldots\ldots \end{array} \right]$$

$$(24) \qquad \text{handhabbar} \equiv \left[\begin{array}{l} \text{bar-comp-A} \\ \ldots\ldots \\ \ldots\ldots \end{array} \right]$$

The structure of the class hierarchy (cf. Fig. 2) ultimately leads us to a treatment of suffixation, esp. *-bar* and *-keit* suffixation (and also of prefixation in general), where the whole process can be described within the framework of UNIFICATION-BASED INHERITANCE REASONING. On what grounds are we allowed to state such a thesis? At first sight, this statement seems to stand in contrast with the claim made above, that NAIVE inheritance is not enough. But we do not rely on naive inheritance as the only mechanism. So we turn now to an examination of why this is so.

We noted earlier that *les+bar* and *Les+bar+keit* are legal lexemes because they satisfy all principles whose left sides they match (implying that they have to meet the right sides too), and because they are composed out of lexicon entries by means of rules. In doing realistic parsing or generation, we might assume additional CONTROL MACHINERY outside of the grammar/lexicon, which uses principles and rules to accept or reject, or alternatively, to generate well-formed (complex) phrases.

Because we regard principles as well as rules as types, equation (3) allows us to employ the laws of feature algebras to construct new types (call them PRECATEGORIES), which are subsumed by all principles having a more general left side and by at least one rule (cf. equations (12) and (13)). Complex words/morphemes like *lesbar* on the other hand will then be subsumed by such precategories.

It is now easy to see that the processes described up to now can be represented entirely via inheritance of a sophisticated kind (effectively constraint resolution). This very interesting observation is motivated as follows: it is possible to define new legal complex word classes by inheriting from precategories as well as from simple lexical categories (cf. subtypes of *part-of-speech* in Fig. 2) and by stating additional local constraints for the class in question. Looking at Fig. 2, *bar-comp-A* (complex adjectives with head daughter *-bar*) and *keit-comp-N* (complex nouns with head daughter *-keit*) are classes of such a kind.

Let's have a closer look at *bar-comp-A* and *keit-comp-N*: *bar-comp-A* inherits from *HCR&Ps* and *A*, but also enforces idiosyncratic constraints, which have to be satisfied by words that are members of this class:

$$(25) \qquad \begin{array}{l} \text{bar-comp-A } = \text{ HCR\&Ps } \wedge \text{ A } \wedge \\ \left[\begin{array}{ll} \text{MORPHS} & \left[\begin{array}{l} \text{HEAD-MORPH } [\textit{ bar-suff}] \\ \text{COMP-MORPH } [\textit{ bar-V}] \end{array} \right] \end{array} \right] \end{array}$$

It's very important to constrain HEAD-MORPH and COMP-MORPH to be of type *bar-suff* resp. *bar-V* respectively,[33] in order to get the right feature structure for *bar-comp-A*. We also require that the adjective class *A* is associated with the following feature structure.

---

[33]Strictly speaking: The value of HEAD-MORPH is a fully expanded instance of type *bar-suff*, whereas COMP-MORPH is bound to an underspecified instance of type *bar-V*, because for instance the value of MORPH|FORM is unspecified.

$$(26) \qquad \text{A} \equiv \left[\begin{array}{l} \text{MORPH} \ \ldots\ldots \\ \text{SYN|LOC} \ \left[\begin{array}{l} \text{LEX} \ + \\ \text{MOD} \ < \ \ldots, \ \text{NP} \ > \\ \text{HEAD|MAJ} \ \text{A} \end{array} \right] \\ \text{SEM} \ \ldots\ldots \end{array} \right]$$

Since furthermore *HCR&Ps* (cf. (13)) is also associated with a feature structure, it's not difficult to construct the prototypical feature structure for *bar-comp-A* by unifying all the information. But once this is achieved, we may construct an entry for *lesbar* by creating an INSTANCE of the class *bar-comp-A* and stating that the complement daughter of this instance is *lesen*, i.e. COMP-MORPH must have as value a feature structure equal to that of the lexeme *lesen* (cf. (14)).

$$(27) \qquad \text{lesbar} = \text{bar-comp-A} \ \wedge \ [\ \text{MORPHS|COMP-MORPH} \ \textit{lesen}\ ]$$

Notice that the feature structure for (27) corresponds to the one for *les+bar* (cf. (16)) provided earlier. In entirely the same fashion, we might INSTANTIATE feature structures for new words like *Les+bar+keit*, which belong to the class *keit-comp-N*. For that purpose, we have to define the class *CN*, which *keit-comp-N* inherits from (cf. Fig. 2).

$$(28) \qquad \text{CN} \equiv \left[\begin{array}{l} \text{MORPH} \ \ldots\ldots \\ \text{SYN|LOC} \ \left[\begin{array}{l} \text{LEX} \ + \\ \text{SUBCAT} \ < \ \ldots, \ \{\text{Det, PosP}\} \ > \\ \text{HEAD|MAJ} \ \text{N} \end{array} \right] \\ \text{SEM} \ \ldots\ldots \end{array} \right]$$

With these definitions in mind, we're able to state the dependence of the complex word class *keit-comp-N* on *HCR&Ps* and *A*, in perfect analogy to equation (25):

$$(29) \qquad \begin{array}{l} \text{keit-comp-N} \ = \ \text{HCR\&Ps} \ \wedge \ \text{CN} \ \wedge \\ \left[\ \text{MORPHS} \ \left[\begin{array}{l} \text{HEAD-MORPH} \ [\ \textit{keit-suff}\ ] \\ \text{COMP-MORPH} \ [\ \textit{keit-A}\ ] \end{array} \right] \ \right] \end{array}$$

We may then represent a feature structure like *Lesbarkeit* by instantiating *keit-comp-N* and imposing a local constraint on this instance:

$$(30) \qquad \text{Lesbarkeit} = \text{keit-comp-N} \ \wedge \ [\ \text{MORPHS|COMP-MORPH} \ \textit{lesbar}\ ]$$

The restriction that COMP-MORPH must be of type *keit-A* (cf. (29)) also allows that COMP-MORPH may be an instance of type *bar-comp-A* defined earlier (cf. (25)) because *keit-A* is a supertype of *bar-comp-A* (cf. Fig. 2).[34]

---

[34]What we said up to now isn't the whole truth. There's an additional restriction we're faced with: given what we have said up till now, nothing prevents us from creating instances which are UNDERSPECIFIED with respect to certain attributes.Such instances do not represent real words and therefore must be forbidden.Take for instance *CN*, the class of common nouns. We might create an instance without specifying the value for MORPH|FORM. Although this instance would be of class *CN*, it couldn't be used by any speaker. Trying to build an instance of type *word* (*lexical-sign*; the most general type of the lexical subsumption hierarchy), which is only constrained to possess the attributes PHON, SYN, and SEM according to Pollard & Sag, would be an extreme case of this shortcoming. This observation holds for PHRASAL SIGNS too, because it is possible to generate sentences without any phonological content, when assuming lexical entries (instances) which are empty with respect to their PHON attribute. As these examples suggest, the possibility of creating underspecified instances depends on there being incorrect grammatical specifications. There are at least two NON-MONOTONIC approaches in order to repair this defect:

1. We introduce a special type, say *undefined*, whose extension is a unique constant (call it NONE). Those attributes we require to be filled at run time (instantiation time) are assigned the type restriction ¬ *undefined* and the contradictory value NONE at definition time. Now, if we carry out a type check at run time, we're able to recognize whether the critical attributes have been assigned real values (by overwriting NONE with a different value), i.e. not to be of type *undefined*.

2. We classify the relevant attributes with a special atom called ANY, which functions in the same way as the ANY works in Kay's FUG: ANY may successfully unify with every object, except ⊥ (Bottom); i.e., the semantics of ANY is that of ⊤ (Top). But when the instantiation of a word is done, all ANYs have to be

This last point should only be seen as a remark to practitioners working on computational lexicons. To enforce, for instance, that elements of the lexicon MUST have their PHON attributes filled, one can use the mechanisms discussed in the footnote.

At the beginning of this section we listed four analytical problems for the description of *-bar* adjectives. During the course of this section we have proposed solutions to these which we summarize here:

Problem	Solution via internal lexical rules
sporadic applicability	use type restrictions
partial regularity	apply non-mon mechanisms or introduce additional classes
category change	treat affix as head of morphological complex
subcategorization change	employ functional dependencies

Before closing this section, we would like to note that the derivational view of passive, which we promised to sketch in the introduction, may be developed straightforwardly on the basis of the analysis of *-bar* sketched here. In particular, the class of verbs involved here is very nearly the same, and the effects on subcategorization (via construct-subcat) identical.

4.4   *Vor-* Prefixation

In this section we investigate the phenomenon of PREFIXATION, focusing for further depth on a specific prefix, namely *vor-*.[35] What prefixes and suffixes have in common is that they serve as heads in our simple head-complement approach. *Vor-* prefixation is in many respects different from *-bar* suffixation and has special properties that makes it interesting for expository purposes:

sporadic applicability *Vor-* prefixes many nouns (e.g. *Vorversion*, *Vorgaben*, *Vorzelt*, *Vorzimmer* and *Vorabend*), but not all.

partial regularity Many *Vor-* derivatives have regular morphological forms and irregular semantics.

category constant The (syntactic) category of the *Vor-* derivative does not change.

subcategorization constant The subcategorization list of the derived complex word (*Vor*+Noun) is taken over from the complement, the noun and does not undergo any changes.

iterability The prefix *Vor-* can be applied iteratively.

Let's examine a noun that may combine with *Vor-* to form a complex noun, viz., the German *Version* 'version':

$$
(31) \qquad \text{Version} \equiv
\begin{bmatrix}
\textit{vor-N} \\
\text{MORPH|FORM}\ \ \text{``Version''} \\
\text{SYN|LOC}\ \begin{bmatrix} \text{SUBCAT}\ \ <\ldots, \{\text{Det}, \text{PosP}\}> \\ \text{LEX}\ + \\ \text{HEAD|MAJ}\ \ \text{N} \end{bmatrix} \\
\text{SEM}\ \begin{bmatrix} \text{PRED}\ \ \text{version}' \end{bmatrix}
\end{bmatrix}
$$

Trying to encode *Vor-* is a bit harder, because *Vor-* not only works on nouns, but also on certain verbs (e.g., *vorgehen*, *vorarbeiten*, or *vorlaufen*). In order to represent this fact, we again make use of distributed disjunctions, which were employed in Section 3 above to encode inflectional paradigms.

---

removed (must be unified 'out'), because ANY henceforth behaves like $\perp$. (Dörre & Eisele, [15], pp. 18, give a formalization of ANY in terms of so-called *meta-constraints*.)

There will be a third possibility of repair, if the underlying (feature) logic allows us to state that certain (under-specified) types (classes) CANNOT be instantiated. Instead, these classes only serve as choice points in our linguistic ontology—reflecting the distinction between REGULARITIES, SUBREGULARITIES, and EXCEPTIONS and enabling a finer granularity of lexical knowledge.

[35]As we mentioned above, we have taken the prefix *Vor-* as an example to show how certain phenomena can be handled in our approach. The assumption that *Vor-* functions semantically as an operator, working on the semantics of the noun (cf. (32)), is of course not an in-depth analysis and may not be useful in real applications. There are other prefixes like *Anti-* or *Ur-* having similar properties, but their semantics is even more complicated.
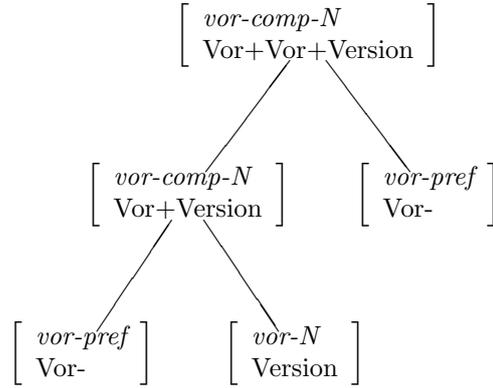
$$
\begin{bmatrix}
\textit{vor-comp-N} \\
\text{Vor+Vor+Version}
\end{bmatrix}
$$

$$
\begin{bmatrix}
\textit{vor-comp-N} \\
\text{Vor+Version}
\end{bmatrix}
\qquad
\begin{bmatrix}
\textit{vor-pref} \\
\text{Vor-}
\end{bmatrix}
$$

$$
\begin{bmatrix}
\textit{vor-pref} \\
\text{Vor-}
\end{bmatrix}
\qquad
\begin{bmatrix}
\textit{vor-N} \\
\text{Version}
\end{bmatrix}
$$

Figure 3: Derviational morphotactics defined through lexical type specifications.

$$
(32) \qquad \text{Vor} \equiv
\begin{bmatrix}
\textit{vor-pref} \\
\text{MORPH|FORM "Vor"} \\
\text{SYN|LOC}
\begin{bmatrix}
\text{HEAD|MAJ } \{\$_1 \ N \ , \ V \} \\
\text{SUBCAT } \{\$_1 \ \text{vor-N}_{\boxed{1}} , \ \text{vor-V}_{\boxed{2}} \}
\end{bmatrix} \\
\text{SEM}
\begin{bmatrix}
\text{OPERATOR vor}' \\
\text{SCOPE } \{\$_1 \ \boxed{1} , \ \boxed{2} \}
\end{bmatrix}
\end{bmatrix}
$$

It is important to understand the intention behind the use of the distributed disjunction in *Vor-*: if *Vor-* combines with a *Vor-* noun, it will be classified as a noun, but if it binds a *Vor-* verb, it creates a verb. Moreover, the head feature principle takes care that the mother of the morphological phrase will be assigned the same category that the prefix *Vor-* bears. That's the main reason why recursion is possible—the new word, e.g., *Vor+Version*, will again be classified as a noun and could then combine with a new *Vor-* (cf. Figure 3) in the same way as described before (actually it is now a COMPLEX noun with internal structure; cf. Fig. 4).

The value of SCOPE under path MORPHS|HEAD-MORPH|SEM now will be assigned by means of structure sharing, coming directly from the semantics of the value of SYN|LOC|SUBCAT, no matter which value of the distributed disjunction is taken (cf. (32)). Finally, by virtue of the semantics principle, *Vorversion* will get its semantics from its head daughter *Vor-*. construct-subcat is again responsible for constructing the right subcategorization list for *Vorversion*: because *Vor-* is the head morph, construct-subcat can detect that the value of SYN|LOC|SUBCAT has to be equal to the entire subcat list of the complement. With these things in mind, we may now construct, with the assistance of the above mentioned principles and the head-complement rule, an admissible feature structure for *Vorversion*, which has the following form:

(33) Vorversion ≡

$$
\begin{bmatrix}
\textit{vor-comp-N} \\
\text{MORPH|FORM} \; \boxed{1} \; \& \; \boxed{2} \\
\text{SYN|LOC} \begin{bmatrix} \text{LEX} \; + \\ \text{SUBCAT} \; \text{construct-subcat}(\boxed{5}) \\ \text{HEAD} \; \boxed{3} \end{bmatrix} \\
\text{SEM} \; \boxed{4} \\[2em]
\text{MORPHS} \; \boxed{5} \begin{bmatrix}
\text{HEAD-MORPH} \begin{bmatrix} \textit{vor-pref} \\ \text{MORPH|FORM} \; \boxed{1} \; \text{``Vor''} \\ \text{SYN|LOC} \begin{bmatrix} \text{SUBCAT} \; \boxed{6} [\;]_{\boxed{7}} \\ \text{HEAD} \; \boxed{3} \, [\, \text{MAJ} \; \text{N} \,] \end{bmatrix} \\ \text{SEM} \; \boxed{4} \begin{bmatrix} \text{OPERATOR} \; \text{vor}' \\ \text{SCOPE} \; \boxed{7} \end{bmatrix} \end{bmatrix} \\[2em]
\begin{matrix} \text{COMP-} \\ \text{MORPH} \end{matrix} \; \boxed{6} \begin{bmatrix} \textit{vor-N} \\ \text{MORPH|FORM} \; \boxed{2} \; \text{``Version''} \\ \text{SYN|LOC} \begin{bmatrix} \text{LEX} \; + \\ \text{SUBCAT} \; < \ldots, \{\text{Det} \, , \text{PosP}\} > \end{bmatrix} \\ \text{SEM} \; \boxed{7} \end{bmatrix}
\end{bmatrix}
\end{bmatrix}
$$

It may be useful to split up the entry for *Vor-* (cf. (32)), distributing its semantics among two feature structures—one (Vor[1]), which combines only with nouns, another one (Vor[2]), which binds instead verbs. But this kind of representation is rather a matter of style.

(34) Vor[1] ≡
$$
\begin{bmatrix}
\textit{vor-pref} \\
\text{MORPH|FORM} \; \text{``Vor''} \\
\text{SYN|LOC} \begin{bmatrix} \text{HEAD|MAJ} \; \text{N} \\ \text{SUBCAT} \; \text{vor-N}_{\boxed{1}} \end{bmatrix} \\
\text{SEM} \begin{bmatrix} \text{OPERATOR} \; \text{vor}' \\ \text{SCOPE} \; \boxed{1} \end{bmatrix}
\end{bmatrix}
$$

(35) Vor[2] ≡
$$
\begin{bmatrix}
\textit{vor-pref} \\
\text{MORPH|FORM} \; \text{``Vor''} \\
\text{SYN|LOC} \begin{bmatrix} \text{HEAD|MAJ} \; \text{V} \\ \text{SUBCAT} \; \text{vor-V}_{\boxed{1}} \end{bmatrix} \\
\text{SEM} \begin{bmatrix} \text{OPERATOR} \; \text{vor}'' \\ \text{SCOPE} \; \boxed{1} \end{bmatrix}
\end{bmatrix}
$$

We described *-bar/-keit* suffixation above by means of unification-based inheritance; once we assume an inheritance network such as Fig. 4, we may analyze *Vor-* prefixation (including recursion) similarly.

Analogous to (25) and (29), we may state the right definitions for *vor-comp-N* and *vor-comp-V* with respect to Fig. 4.

(36)
$$
\text{vor-comp-N} \; = \; \text{HCR\&Ps} \; \wedge \; \text{vor-N} \; \wedge \\
\begin{bmatrix} \text{MORPHS} \begin{bmatrix} \text{HEAD-MORPH} \; [\, \textit{vor-pref} \,] \\ \text{COMP-MORPH} \; [\, \textit{vor-N} \,] \end{bmatrix} \end{bmatrix}
$$

(37)
$$
\text{vor-comp-V} \; = \; \text{HCR\&Ps} \; \wedge \; \text{vor-V} \; \wedge \\
\begin{bmatrix} \text{MORPHS} \begin{bmatrix} \text{HEAD-MORPH} \; [\, \textit{vor-pref} \,] \\ \text{COMP-MORPH} \; [\, \textit{vor-V} \,] \end{bmatrix} \end{bmatrix}
$$

The iterated application of *Vor-* will be guaranteed by using the RECURSIVE type definition of *vor-comp-N*: the value of the attribute MORPHS|COMP-MORPH in *vor-comp-N* is required to be of type *vor-N*. Because *vor-N* subsumes *vor-comp-N*, we're allowed in particular to require that the value of MORPHS|COMP-MORPH should be an instance of type *vor-comp-N*, and this corresponds to a potentially infinite iteration of *Vor-*. But how do we block infinite recursion in cases of concrete words? Only
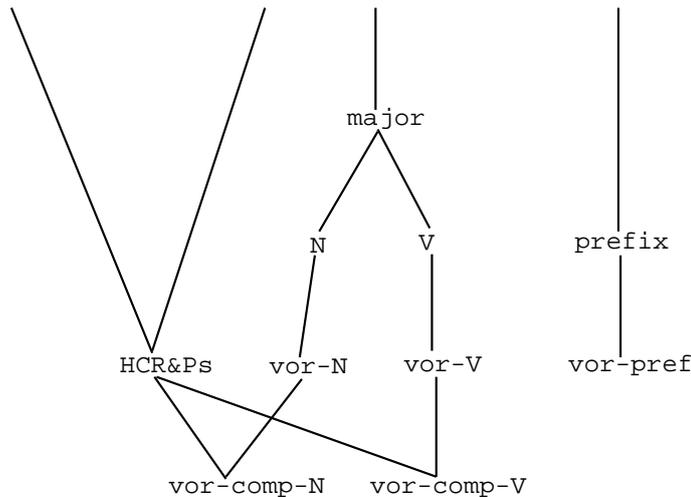
Figure 4: Structure of the inheritance network in case of *Vor-* prefixation, regarding the principles and the rule. Note, that we additionally impose LOCAL constraints on certain classes, especially on *vor-comp-N* and *vor-comp-V*.

instances of minimal type *vor-N* will stop parsing or generation, because those instances don't have any internal constituent structure (no MORPHS attribute), i.e., there's no way to expand them further.[36] It is the indirect self-reference that is responsible for the recursive nature of *vor-comp-N*. If we now still require that the value of MORPHS.COMP-MORPH not be underspecified, the aim of describing *Vor-* prefixation by inheritance only is fully realized.

Constructing an entry for *Vorversion* is done in a trivial way by instantiating *vor-comp-N* and by imposing an additional restriction on that instance, namely that the complement daughter must hold a feature structure representing *Version*.

(38) Vorversion = vor-comp-N $\land$ [MORPHS|COMP-MORPH *Version*]

5   Summary and Conclusions

In this section we summarize the results of the present study, pointing out areas which we have yet to investigate, but which seem pertinent or promising; the further investigation of these areas offers the best practical means to advances in this approach to computational lexicology.

The results of the present study may be viewed as follows: feature-based formalisms (such as PATR-II or the HPSG formalism) have been successful in the description of SYNTAGMATIC grammatical relations—the relations between the various syntactic parts of an utterance token. The present study attempts to demonstrate that the feature description languages developed for this purpose may also be applied fruitfully to PARADIGMATIC relations—the relations between words and their common alternatives in utterances.[37] We have examined proposals here for the representation of inflectional and derivational relations. These relations are purely lexical, since they may never result in syntactic relations, so that a distinctly lexical status accrues to them even in a highly "lexicalized theory" such as HPSG. This is the sense in which the present contribution claims to develop further the theory of the LEXICON for feature-based theories of language. It is of course clear to us that several aspects of this theory—most clearly its syntactic content—have been under development for some time, but the larger theoretical picture had not been clarified.

We have attempted to provide that clarification, so that our proposals here have thus been programmatic, but we provide concrete elaborations in two central areas of paradigmatic relations, inflection and

---

[36]When we say *an instance of minimal type vor-N*, we exclude instances of types more specific than *vor-N*. This corresponds roughly to CLASSIFICATION in KL-ONE-like knowledge representation systems. By making a distinction between class and instance, and assuming totally well-typed feature structures, this goal is easily reached.

[37]We appreciate Carl Pollard's suggesting this contrast to us.

derivation. Our proposal for inflection may be seen as a variant of one first proposed in DATR: we characterize the inflectional variants of a lexeme as alternative (disjunctive) realizations. The basic insight of DATR is easily accommodated within the language of feature structure descriptions. Alternative realizations of lexemes—paradigms—are represented using the technical tool of distributed disjunctions (although there are several equivalent means of representation). Our proposal for derivation may be seen as an application of the HPSG treatment of syntactic structure in feature structure formalism. Just as HPSG characterizes phrase structure rules via descriptions of the output phrases created by the rule, so we propose characterizing (derivational) word formation rules via recursive constraints on complex lexemes—those which linguists would regard as created by the "rule". This contrasts with the usual treatment in feature-based theories, which construes derivational rules (in fact, normally ALL lexical rules) as mappings within the algebra of feature structures. The latter proposal relies on subsidiary functions or relations to characterize lexical rules, while our own characterization remains within the language of feature-structure description. Our proposal is probably preferable in direct proportion to the degree to which derivational structure employs mechanisms which feature formalisms describe well—inheritance, typing, and shared structure. We suggest that the inheritance exploited in structured lexicons finds very apt application in derivation as well.

We do not imagine that this paper represents a mature presentation of what a feature-based lexicon ought to be capable of. In particular, we continue to puzzle over several areas: the correct representation of compounding; the treatment of idioms ; the most attractive analysis of so-called "zero-derivation" (e.g., the relation between verbal participles and the adjectives derived from them); the proper interface to allomorphy—both theoretically and practically; the role of morpheme-based generalizations; and the question of flexible access (for both parsing and generation) to lexical structure. We regard these as challenging issues for future work in feature-based lexical analysis.

The further investigation of these areas, together with continuing work on elaborations and alternatives to the analyses suggested here, offers the best practical means to advances in this approach to computational lexicology.

References

[1] J.H. Allen and H.B.Greenough. *New Latin Grammar*. Ginn and Company, Boston, 1903.

[2] Stephen R. Anderson. Inflection. In Michael Hammond and Michael Noonan, editors, *Theoretical Morphology*, pages 23–43. Academic Press, Orlando, 1988.

[3] Stephen R. Anderson. Morphological theory. In Frederick J. Newmeyer, editor, *Linguistics: The Cambridge Survey*, volume 1, pages 146–91. Cambridge University Press, Cambridge, 1988.

[4] Emmon Bach. Categorial grammars as theories of language. In Richard T. Oehrle, Emmon Bach, and Deirdre Wheeler, editors, *Categorial Grammars and Natural Language Structures*, pages 17–34. Reidel, Dordrecht and Boston, 1988.

[5] Rolf Backofen, Lutz Euler, and Günter Görz. Towards the integration of functions, relations and types in an AI programming language. In *Proc. of GWAI-90*, Berlin, 1990. Springer.

[6] Steven Bird. Prosodic morphology and constraint-based phonology. Research Paper EUCCSRP-38, Centre for Cognitive Science, University of Edinburgh, 1990.

[7] Gosse Bouma. Defaults in unification grammar. In *Proc. of the 28th Annual Meeting of the ACL*, pages 165–172. Association for Computational Linguistics, 1990.

[8] Joan Bresnan, editor. *The Mental Representation of Grammatical Relations*. MIT Press, Cambridge, Mass., 1982.

[9] Jonathan Calder. Paradigmatic morphology. In *Proc. of the 5th European ACL*, 1988.

[10] Luca Cardelli and Peter Wegner. On understanding types, data abstraction, and polymorphism. *ACM Computing Surveys*, 17(4):471–522, 1985.

[11] Bob Carpenter. *The Logic of Typed Feature Structures*. Number 32 in Tracts in Theoretical Computer Science. Cambridge University Press, Cambridge, 1992.

[12] Bob Carpenter, Carl J. Pollard, and Alex Franz. The specification and implementation of constraint-based unification grammar. In *Proc. of the Second International Workshop on Parsing Technology*, Cancun, 1991.

[13] William Cook, Walt Hill, and Peter Canning. Inheritance is not subtyping. In *ACM SIGACT-SIGPLAN Symposium on Principles of Programming Languages (POPL)*, 1990.

[14] Jochen Dörre and Andreas Eisele. Determining consistency of feature terms with distributed disjunctions. In Dieter Metzing, editor, *Proc. of GWAI-89 (15th German Workshop on AI)*, pages 270–279, Berlin, 1989. Springer-Verlag.

[15] Jochen Dörre and Andreas Eisele. A comprehensive unification-based grammar formalism. Technical Report Deliverable R3.1.B, DYANA, Centre for Cognitive Science, Edinburgh, 1991.

[16] Roger Evans and Gerald Gazdar. The DATR papers. Technical Report SCRP 139, School of Cognitive and Computing Sciences, University of Sussex, 1990.

[17] Daniel Flickinger. *Lexical Rules in the Hierarchical Lexicon*. PhD thesis, Stanford University, 1987.

[18] Daniel Flickinger and John Nerbonne. Inheritance and complementation: A case study of *easy* adjectives and related nouns. *Computational Linguistics*, 19(3):269–309, 1992.

[19] Daniel Flickinger, Carl Pollard, and Thomas Wasow. Structure-sharing in lexical representation. In *Proc. of the 25th ACL*, 1985.

[20] Gerald Gazdar. Linguistic applications of default inheritance mechanisms. In Peter Whitelock, Mary McGee Wood, Harald L. Somers, Rod Johnson, and P.Bennett, editors, *Linguistic Theory and Computer Applications*. Academic Press, London, 1987.

[21] Gerald Gazdar. An introduction to DATR. In Roger Evans and Gerald Gazdar, editors, *The DATR Papers*, number SCRP 139 in Cognitive Science Research Reports, pages 1–14. School of Cognitive and Computing Sciences, University of Sussex, 1990.

[22] Gerald Gazdar, Ewan Klein, Geoffrey Pullum, and Ivan Sag. *Generalized Phrase Structure Grammar*. Harvard University Press, 1985.

[23] Jack Hoeksema. *Categorial Morphology*. Garland, New York, 1985.

[24] Mark Johnson. *Attribute Value Logic and the Theory of Grammar*. Center for the Study of Language and Information, Stanford, 1988.

[25] Ronald Kaplan and Joan Bresnan. Lexical-functional grammar: A formal system for grammatical representation. In Joan Bresnan, editor, *The Mental Representation of Grammatical Relations*, pages 173–281. MIT Press, Cambridge, Mass, 1982.

[26] Lauri Karttunen. D-PATR: A development environment for unifiaction-based grammars. Technical Report CSLI-86-61, Center for the Study of Language and Information, Stanford University, 1986.

[27] Robert T. Kasper and William C. Rounds. A logical semantics for feature structures. In *Proc. of the 24th Annual Meeting of the Association for Computational Linguistics*, pages 257–266, Columbia University, 1986.

[28] Andreas Kathol. Verbal and adjectival passives in german. In *MIT Working Papers in Linguistics*, volume 14. MIT, 1991.

[29] Martin Kay. Functional unification grammar: A formalism for machine translation. In *Proc. of COLING*, pages 75–78, 1984.

[30] Martin Kay. Parsing in functional unification grammar. In David Dowty and Lauri Karttunen, editors, *Natural Language Parsing*. Cambridge University Press, Cambridge, England, 1985.

[31] James Kilbury, Petra Naerger, and Ingrid Renz. DATR as a lexical component for PATR. In *Proc. of the 6th Meeting of the European Chapter of the Association for Computational Linguistics*, 1991.

[32] Kimmo Koskenniemi. Two-level model for morphological analysis. In *Proc. of IJCAI*, pages 683–685, 1983.

[33] Hans-Ulrich Krieger. Lexicon-driven derivation in hpsg. In John Nerbonne, Klaus Netter, and Carl Pollard, editors, *German Grammar in HPSG*, pages xxx–yyy. CSLI, Stanford, to appear, 1992.

[34] P.H. Matthews. *Inflectional Morphology: A Theoretical Study Based on Aspects of Latin Verb Conjugation*. Cambridge University Press, Cambridge, England, 1972.

[35] P.H. Matthews. *Morphology*. Cambridge University Press, Cambridge, England, 1974.

[36] Carl Pollard and Ivan Sag. *Information-Based Syntax and Semantics, Vol.I*. CSLI, Stanford, 1987.

[37] Mike Reape. An introduction to the semantics of unification-based grammar formalisms. Technical Report R3.2.A, DYANA, University of Edinburgh, 1991.

[38] Graeme D. Ritchie, Stephen G. Pulman, Alan W. Black, and Graham Russell. A computational framework for lexical description. *Computational Linguistics*, 13(3-4):290–307, 1987.

[39] William C. Rounds and Robert T. Kasper. A complete logical calculus for record structures representing linguistic information. In *Proc. of the 15th Annual Symposium of the on Logic in Computer Science*, Cambridge, 1986.

[40] Graham Russell, John Carroll, and Susan Warwick. Multiple default inheritance in a unification-based lexicon. In *Proc. of the 29th Annual Meeting of the Association for Computational Linguistics*, 1991.

[41] Jerry Sadock. Autolexical syntax: A proposal for the treatment of noun incorporation and similar phenomena. *Natural Language and Linguistic Theory*, 3:379–439, 1985.

[42] Ivan Sag and Carl Pollard. Head-driven phrase structure: An informal synopsis. Technical Report CSLI-87-89, Center for the Study of Language and Information, Stanford University, 1987.

[43] Stuart Shieber. *An Introduction to Unification-Based Approaches to Grammar*. Center for the Study of Language and Information, Stanford University, 1986.

[44] Stuart Shieber, Hans Uszkoreit, Fernando Pereira, Jane Robinson, and Mabry Tyson. The formalism and implementation of PATR-II. In *Research on Interactive Acquisition and Use of Knowledge*. Artificial Intelligence Center, SRI International, Menlo Park, Calif., 1983.

[45] Koenraad De Smedt. Using object-oriented knowledge representation techniques in morphology and syntax programming. In *Proc. of the 1984 European Conference on Artificial Intelligence*, pages 181–184, 1984.

[46] Gert Smolka. A feature logic with subsorts. Technical Report 33, WT LILOG–IBM Germany, 1988.

[47] Jindrich Toman. *Wortsyntax: Eine Diskussion ausgewählter Probleme deutscher Wortbildung*. Niemeyer, Tübingen, 1983.

[48] Harald Trost. The application of two-level morphology to non-concatenative german morphology. In *Proc. of COLING*, pages 371–376, 1990.

[49] Arnold M. Zwicky. How to describe inflection. In *Proc. of the 11th Annual Meeting of the Berkeley Linguistics Society*, pages 372–86, 1985.

[50] Arnold M. Zwicky. Inflectional morphology as a subcomponent of grammar. In *Proc. of the 3rd International Morphology Meeting*, 1988.