

Coping With Derivation in a Morphological Component *

Harald Trost

Austrian Research Institute for Artificial Intelligence
Schottengasse 3, A-1010 Wien
Austria
email: harald@ai.univie.ac.at

Abstract

In this paper a morphological component with a limited capability to automatically interpret (and generate) derived words is presented. The system combines an extended two-level morphology [Trost, 1991a; Trost, 1991b] with a feature-based word grammar building on a hierarchical lexicon. Polymorphemic stems not explicitly stored in the lexicon are given a compositional interpretation. That way the system allows to minimize redundancy in the lexicon because derived words that are transparent need not to be stored explicitly. Also, words formed ad-hoc can be recognized correctly. The system is implemented in CommonLisp and has been tested on examples from German derivation.

1 Introduction

This paper is about words. Since *word* is a rather fuzzy term we will first try to make clear what *word* means in the context of this paper. Following [di Sciullo and Williams, 1989] we discriminate two senses. One is the *morphological word* which is built from morphs according to the rules of morphology. The other is the *syntactic word* which is the atomic entity from which sentences are built according to the rules of syntax.

*Work on this project was partially sponsored by the Austrian Federal Ministry for Science and Research and the "Fonds zur Förderung der wissenschaftlichen Forschung" grant no.P7986-PHY. I would also like to thank John Nerbonne, Klaus Netter and Wolfgang Heinz for comments on earlier versions of this paper.

These two views support two different sets of information which are to be kept separate but which are not disjunctive. The syntactical word carries information about category, valency and semantics, information that is important for the interpretation of a word in the context of the sentence. It also carries information like case, number, gender and person. The former information is basically the same for all different surface forms of the syntactic word¹ the latter is conveyed by the different surface forms produced by the inflectional paradigm and is therefore shared with the morphological word.

Besides this shared information the morphological word carries information about the inflectional paradigm, the stem, and the way it is internally structured. In our view the lexicon should be a mediator between these two views of word.

Traditionally, the lexicon in natural language processing (NLP) systems is viewed as a finite collection of syntactic words. Words have stored with them their syntactic and semantic information. In the most simple case the lexicon contains an entry for every different word form. For highly inflecting (or agglutinating) languages this approach is not feasible for realistic vocabulary sizes. Instead, morphological components are used to map between the different surface forms of a word and its canonical form stored in the lexicon. We will call this canonical form and the information associated with it *lexeme*.

There are problems with such a static view of the lexicon. In the open word classes our vocabulary is potentially infinite. Making use of derivation and compounding speakers (or writers) can and do always create new words. A majority of these words

¹For some forms like the passive PPP some authors assume different syntactic features. Nevertheless they are derived regularly, e.g., by lexical rules.

are invented on the spot and may never be used again. Skimming through real texts one will always find such ad-hoc formed words not to be found in any lexicon that are nevertheless readily understood by any competent reader. A realistic NLP system should therefore have means to cope with ad-hoc word formation.

Efficiency considerations also support the idea of extending morphological components to treat derivation. Because of the regularities found in derivation a lexicon purely based on words will be highly redundant and wasting space. On the other hand a large percentage of lexicalized derived words (and compounds) is no longer transparent syntactically and/or semantically and has to be treated like a monomorphemic lexeme. What we do need then is a system that is flexible enough to allow for both a compositional and an idiosyncratic reading of polymorphemic stems.

The system described in this paper is a combination of a feature-based hierarchical lexicon and word grammar with an extended two-level morphology. Before describing the system in more detail we will shortly discuss these two strands of research.

2 Inheritance Lexica

Research directed at reducing redundancy in the lexicon has come up with the idea of organizing the information hierarchically making use of inheritance (see, e.g. [Daelemans *et al.*, 1992; Russell *et al.*, 1992]).

Various formalisms supporting inheritance have been proposed that can be classified into two major approaches. One uses defaults, i.e., inherited data may be overwritten by more specific ones. The default mechanism handles exceptions which are an inherent phenomenon of the lexicon. A well-known formalism following this approach is DATR [Evans and Gazdar, 1989].

The major advantage of defaults is the rather natural hierarchy formation it supports where classes can be organized in a tree instead of a multiple-inheritance hierarchy. Drawbacks are that defaults are computationally costly and one needs an interface to the sentence grammar which is usually written in default-free feature descriptions.

Although the term default is taken from knowledge representation one should be aware of the quite different usage. In knowledge representation defaults are used to describe uncertain facts which may or may not become explicitly known later on.² Exceptions in the lexicon are of a different nature because they form an a priori known set. For any word it is

²An example for the use of defaults in knowledge representation is an inference rule like *Birds typically can fly*. In the absence of more detailed knowledge this allows me to conclude that Tweety which I only know to be a bird can fly. Should I later on get the additional information that Tweety is a penguin I must revoke that conclusion.

known whether it is regular or an exception.³ The only motivation to use defaults in the lexicon is that they allow for a more concise and natural representation.

The alternative approach organizes classes in a multiple-inheritance hierarchy without defaults. This means that lexical items can be described as standard feature terms organized in a type hierarchy (see, e.g., [Smolka, 1988; Carpenter *et al.*, 1991]). The advantages are clear. There is no need for an interface to the grammar and computational complexity is lower.

At the moment it is an open question which of the two approaches is the more appropriate. In our system we decided against introducing a new formalism. Most current natural language systems are based on feature formalisms and we see no obvious reason why the lexicon should not be feature-based (see also [Nerbonne, 1992]).

While inheritance lexica—concerned with the syntactic word—have mainly been used to express generalizations over classes of words the idea can also be used for the explicit representation of derivation. In [Nerbonne, 1992] we find such a proposal. What the proposal shares with most of the other schemes is that not much consideration is given to morphophonology. The problem is acknowledged by some authors by using a function *morphologically append* instead of pure concatenation of morphs but it remains unclear how this function should be implemented.

The approach presented here follows this line of research in complementing an extended two-level morphology with a hierarchical lexicon that contains as entries not only words but also morphs. This way morphophonology can be treated in a principled way while retaining the advantages of hierarchical lexica.

3 Two-Level Morphology

For dealing with a compositional syntax and semantics of derivatives one needs a component that is capable of constructing arbitrary words from a finite set of morphs according to morphotactic rules. Very successful in the domain of morphological analysis/generation are finite-state approaches, notably two-level morphology [Koskenniemi, 1984]. Two-level morphology deals with two aspects of word formation:

Morphotactics: The combination rules that govern which morphs may be combined in what order to produce morphologically correct words.

Morphophonology: Phonological alterations occurring in the process of combination.

Morphotactics is dealt with by a so-called continuation lexicon. In expressiveness that is equivalent to a finite state automaton consuming morphs.

³We do not consider language acquisition here.

Morphophonology is treated by assuming two distinct levels, namely a lexical and a surface level. The lexical level consists of a sequence of morphs as found in the lexicon; the surface level is the form found in the actual text/utterance. The mapping between these two levels is constrained by so-called two-level rules describing the contexts for certain phonological alterations.

An example for a morphophonological alteration in German is the insertion of *e* between a stem ending in a *t* or *d*, and a suffix starting with *s* or *t*, e.g., 3rd person singular of the verb *arbeiten* (to work) is *arbeitet*. In two-level morphology that means that the lexical form *arbeit+st* has to be mapped to surface *arbeitet*. The following rule will enforce just that mapping:

- (1) $+e \Leftrightarrow \{d, t\} - \{s, t\}$;

A detailed description of two-level morphology can be found in [Sproat, 1992, chapter 3].

In its basic form two-level morphology is not well suited for our task because all the morphosyntactic information is encoded in the lexical form. When connected to a syntactic/semantic component one needs an interface to mediate between the morphological and the syntactic word. We will show in in chapter 5 how our version of two-level-morphology is extended to provide such an interface.

4 Derivation in German

Usually, in German derived words are morphologically regular.⁴ Morphophonological alterations are the same as for inflection only the occurrence of umlaut is less regular. Syntax and semantics on the other hand are very often irregular with respect to compositional rules for derivation.

As an example we will look at the German derivational prefix *be-*. This prefix is both very productive and considered to be rather regular. The prefix *be-* produces transitive verbs mostly from (intransitive) verbs but also from other word categories. We will restrict ourselves here to all those cases where the new verb is formed from a verb. In the new verb the direct object role is filled by a modifier role of the original verb while the original meaning is basically preserved. One regularly formed example is *bearbeiten* derived from the intransitive verb *arbeiten* (to work).

- (2) $[Maria]_{SUBJ}$ *arbeitet* $[an\ dem\ Papier]_{POBJ}$.
Mary works on the paper.
- (3) $[Maria]_{SUBJ}$ *bearbeitet* $[das\ Papier]_{OBJ}$.

Skimming through [Wahrig, 1978] we find 238 en-

⁴Most exceptions are regularly inflecting compound verbs derived from an irregular verb, e.g., *handhaben* (to manipulate) a regular verb derived from the irregular verb *haben* (to have).

tries starting with prefix *be-*. 91 of these can be excluded because they cannot be explained as being derived from verbs. Of the remaining 147 words about 60 have no meaning that can be interpreted compositionally.⁵ The remaining ones do have at least one compositional meaning.

Even with those the situation is difficult. In some cases the derived word takes just one of the meanings of the original word as its semantic basis, e.g., *befolgen* (to obey) is derived from *folgen* in the meaning to obey, but not to follow or to ensue:

- (4) *Der Soldat folgt [dem Befehl]_{IOBJ}*.
The soldier obeys the order.
- (5) *Der Soldat befolgt [den Befehl]_{OBJ}*.
- (6) *Der Soldat folgt [dem Offizier]_{IOBJ}*.
The soldier follows the officer.
- (7) **Der Soldat befolgt [den Offizier]_{OBJ}*.

In other cases we have a compositional as well as a non-compositional reading, e.g., *besetzen* derived from *setzen* (to set) may either mean to set or to occupy.

What is needed is a flexible system where regularities can be expressed to reduce redundancy while irregularities can still easily be handled.

5 The Morphological Component X2MORF

X2MORF [Trost, 1991a; Trost, 1991b] that forms the basis of our system is a morphological component based on two-level morphology. X2MORF extends the standard model in two way which are crucial for our task. A feature-based word grammar replaces the continuation class approach thus providing a natural interface to the syntax/semantics component. Two-level rules are provided with a morphological filter restricting their application to certain morphological classes.

5.1 Feature-Based Grammar and Lexicon

In X2MORF morphotactics are described by a feature-based grammar. As a result, the representation of a word form is a feature description. The word grammar employs a functor argument structure with binary branching.

Let us look at a specific example. The (simplified) entry for the noun stem *Hand* (hand) is given in fig.1.

To form a legal word that stem must combine with an inflectional ending. Fig.2 shows the (simplified) entry for the plural ending. Note that plural formation also involves umlaut, i.e., the correct surface

⁵About half of them are actually derived from words from other classes like *befehlen* (to order) which is clearly derived from the noun *Befehl* (order) and not the verb *fehlen* (to miss).

MORPH:	CAT: <i>N</i>
	PARAD: <i>e-plural</i>
	UMLAUT: <i>binary</i>
	PHON: <i>hAnd</i>
STEM:	<i><hand></i>

Figure 1: Lexical entry for *Hand* (preliminary)

form is *Hände*. As we will see later on this is what the feature UMLAUT is needed for.

MORPH:	CAT: <i>N</i>	
	NUM: <i>pl</i>	
	CASE: { <i>nom gen acc</i> }	
PHON:	<i>+e</i>	
STEM:	1	
ARG:	MORPH:	CAT: <i>N</i>
		PARAD: <i>e-plural</i>
		UMLAUT: <i>+</i>
		STEM: 1

Figure 2: Lexical entry for suffix *e* (preliminary)

Combining the above two lexical entries in the appropriate way leads to the feature structure described in fig.3.

MORPH:	CAT: <i>N</i>	
	NUM: <i>pl</i>	
	CASE: { <i>nom gen acc</i> }	
PHON:	<i>+e</i>	
STEM:	1 <i><hand></i>	
ARG:	MORPH:	CAT: <i>N</i>
		PARAD: <i>e-plural</i>
		UMLAUT: <i>+</i>
		PHON: <i>hAnd</i>
STEM:	1	

Figure 3: Resulting feature structure for *Hände*

5.2 Extending Two-level Rules with Morphological Contexts

X2MORF employs an extended version of two-level rules. Besides the standard phonological context they also have a morphological context in form of a feature structure. This morphological context is unified with the feature structure of the morph to which the character pair belongs. This morphological context serves two purposes. One is to restrict the application of morphophonological rules to suitable morphological contexts. The other is to enable the

transmission of information from the phonological to the morphological level.

We can now show how umlaut is treated in X2MORF. A two-level rule constrains the mapping of *A* to *ä* to the appropriate contexts, namely where the inflection suffix requires umlaut:

(8) $A:\ddot{a} \Leftrightarrow - ; [\text{MORPH: } [\text{HEAD: } [\text{UMLAUT: } +]]]$

The occurrence of the umlaut *ä* in the surface form is now coupled to the feature UMLAUT taking the value *+*. As we can see in fig.3 the plural ending has forced the feature to take that value already which means that the morphological context of the rule is valid.

Reinhard [Reinhard, 1991] argues that a purely feature-based approach is not well suited for the treatment of umlaut in derivation because of its idiosyncrasy. One example are different derivations from *Hand* (*hand*) which takes umlaut for plural (*Hände*) and some derivations (*händisch*) but not for others (*handlich*). There are also words like *Tag* (*day*) where the plural takes no umlaut (*Tage*) but derivations do (*täglich*). Reinhard maintains that a default mechanism like DATR is more appropriate to deal with umlaut.

We disagree since the facts can be described in X2MORF in a fairly natural manner. Once the equivalence classes with respect to umlaut are known we can describe the data using a complex feature UMLAUT⁶ instead of the simple binary one. This complex feature UMLAUT consists of a feature for each class, which takes as value *+* or *-* and one feature *value* for the recording of actual occurrence of umlaut:

UMLAUT:	VALUE: <i>binary</i>
	PL-UML: <i>binary</i>
	LICH-UML: <i>binary</i>
	ISCH-UML: <i>binary</i>

The value of the feature UMLAUT|VALUE is set by the morphological filter of the two-level rule triggering umlaut, i.e., if an umlaut is found it is set to *+* otherwise to *-*. The entries of those affixes requiring umlaut set the value of their equivalence class to *+*. Therefore the relevant parts of the entries for *-lich* and *-isch* look like $[\text{UMLAUT: } [\text{LICH-UML: } +]]$ and $[\text{UMLAUT: } [\text{ISCH-UML: } +]]$ because both these endings normally require umlaut.

As we have seen above the noun *Hand* comes with umlaut in the plural (*Hände*) and the derived adjective *händisch* (*manually*) but (irregularly) without umlaut in the adjective *handlich* (*handy*). In fig.4 we show the relevant part of the entry for *Hand* that produces the correct results. The regular cases are

⁶In our simplified example we assume just 3 classes (for plural, derivation with *-lich* and *-isch*). In reality the number of classes is larger but still fairly small.

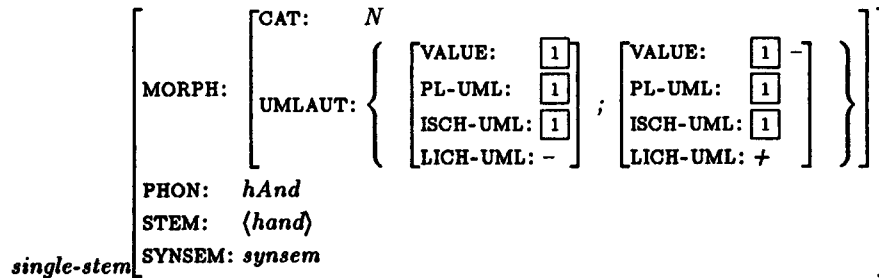


Figure 4: Lexical entry for *Hand* (final version)

taken care of by the first disjunct while the exceptions are captured by the second.

The first disjunct in this feature structure takes care of all cases but the derivation with *-lich*. The entries for plural (see fig.5) and *-isch* come with the value + forcing the VALUE feature also to have a + value. The entry for *-lich* also comes with a + value and therefore fails to unify with the first disjunct. Suffixes that do not trigger umlaut come with the VALUE feature set to -.

The second disjunct captures the exception for the *-lich* derivation of *Hand*. Because of requiring a - value it fails to unify with the entries for plural and *-isch*. The + value for *-lich* succeeds forcing at the same time the VALUE feature to be -.

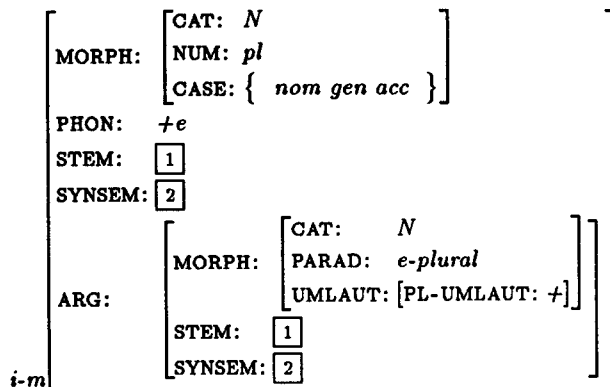


Figure 5: Lexical entry for suffix *e* (final version)

This mechanism allows us to describe the umlaut phenomenon in a very general way while at the same time being able to deal with exceptions to the rule in a simple and straightforward manner.

5.3 Using X2MORF directly for derivation

Regarding morphotactics and morphophonology there is basically no difference between inflection and derivation. So one could use X2MORF as it is to cope with derivation. Derivation particles are word-forming heads [di Sciullo and Williams, 1989] that have to be complemented with the appropriate (sim-

ple or complex) stems. Words that cannot be interpreted compositionally anymore have to be regarded as monomorphemic and must be stored in the morph lexicon.

Such an approach is possible but it poses some problems:

- The morphological structure of words is no more available to succeeding processing stages. For some phenomena just this structural information is necessary though. Take as an example the partial deletion of words in phrases with conjunction (*Ein- und Verkauf*).
- The compositional reading of a derived word cannot be suppressed⁷, even worse, it is indistinguishable from the correct reading (remember the *befehlen* example).
- Partial regularities cannot be used anymore to reduce redundancy.

Therefore we have chosen instead to augment X2MORF with a lexeme lexicon and an explicit interface between morphological and syntactic word.

6 System Architecture

Logically, the system uses two different lexica. A *morph lexicon* contains all the morphs, i.e., monomorphemic stems, inflectional and derivational affixes. This lexicon is used by X2MORF. A *lexeme lexicon* contains the lexemes, i.e. stem morphs and derivational endings (because of their word-forming capacity). The lexical entries contain the lexeme-specific syntactic and semantic information under the feature SYNSEM.

These two lexica can be merged into a single type hierarchy (see fig.6) where the morph lexicon entries are of type *morph* and lexeme lexicon entries of type *lexeme*. *Single-stems* and *deriv-morphs* share the properties of both lexica.

⁷One could argue that the idea of preemption is incorrect anyway and that only syntactic or semantic restrictions block derivation. While this may be true in theory at least for practical considerations we will need to be able to block derivation in the lexicon.

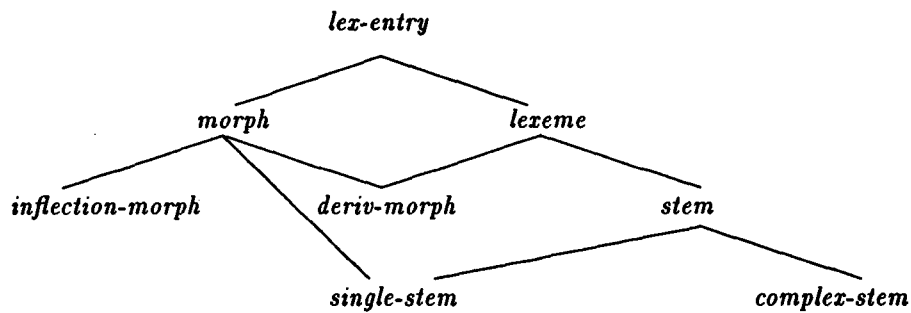


Figure 6: Part of the type lattice of the lexicon

Since we have organized our lexica in a type hierarchy we have already succeeded in establishing an inheritance hierarchy. We can now impose any of the structures proposed in the literature (e.g., [Krieger and Nerbonne, 1991; Russell *et al.*, 1992]) for hierarchical lexica on it, as long as they observe the same functor argument structure of words crucial to our morphotactics.

Why are we now in a better situation than by using X2MORF directly? Because complex stems are no morphs and therefore inaccessible to X2MORF. They are only used in a second processing stage where complex words can be given a non-compositional reading. To make this possible the assigning of compositional readings must also be postponed to this second stage. This is attained by giving derivation morphs in the lexicon no feature SYNSEM but stating the information under FUNCTOR|SYNSEM instead.

In the first stage X2MORF processes the morphotactic information including the word-form-specific morphosyntactic information making use of the morph lexicon. The result is a feature-description containing the morphotactic structure and the morphosyntactic information of the processed word form. What has also been constructed is a value for the STEM feature that is used as an index to the lexeme lexicon in the second processing stage.⁸

In the second stage we have to discriminate between the following cases:

- The stem is found in the lexeme lexicon. In case of a monomorphemic stem processing is completed because the relevant syntactic/semantic information has already been constructed during the first stage. In case of a polymorphemic stem the retrieved lexical entry is unified with the result of the first stage, delivering the lexicalized interpretation.

⁸Inflectional endings do not contribute to the stem. Also, allomorphs like irregular verb forms share a common stem.

- The stem is *not* found in the lexeme lexicon. In that case a compositional interpretation is required. This is achieved by unifying the result of stage one with the feature structure shown in fig.7 This activates the SYNSEM information of the functor—which must be either an inflection or a derivation morph. In case of an inflection morph nothing really happens. But for derivation morphs the syntactic/semantic information which has already been constructed is bound to the feature SYNSEM. Then the process must recursively be applied to the argument of the structure. Since all monomorphemic stems and all derivational affixes are stored in the lexeme lexicon this search is bound to terminate.

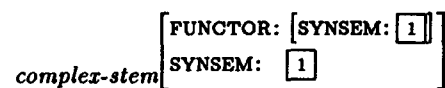


Figure 7: Default entry in the lexeme lexicon

How does this procedure account for the flexibility demanded in section 4. By keeping the compositional syntactic/semantic interpretation local to the functor during morphological interpretation the decision is postponed to the second stage. In case there is no explicit entry found this compositional interpretation is just made available.

In case of an explicit entry in the lexeme lexicon there is a number of different possibilities, among them:

- There are just lexicalized interpretations.
- There is a compositional as well as a lexicalized interpretation.
- The compositional interpretation is restricted to a subset of the possible semantics of the root.

The entries in the lexeme lexicon can easily be tailor-made to fit any of these possibilities.

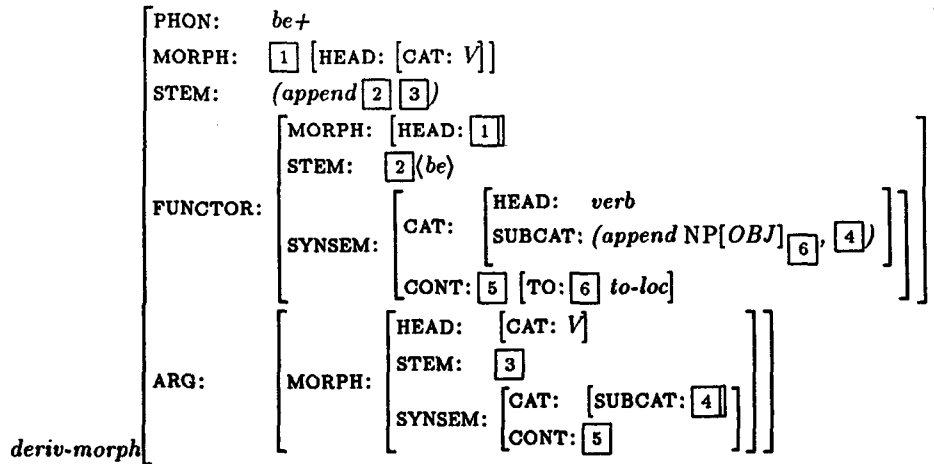


Figure 8: Lexical entry for the derivational prefix *be-*

7 A Detailed Example

We will now illustrate the workings of the system using a few examples from section 4. The first example describes the purely compositional case. The verb *betreten* (*to enter*) can be regularly derived from *treten* (*to enter*) and the suffix *be-*. The sentences

(9) *Die Frau tritt [in das Zimmer]POBJ.*
The woman enters the room.

(10) *Die Frau betritt [das Zimmer]OBJ.*

are semantically equivalent. The prepositional object of the intransitive verb *treten* is transformed into a direct object making *betreten* a transitive verb. A number of verbs derived by using the particle *be-* follows this general pattern. Figure 8 shows—a simplified version of—the lexical entry for *be-*.

The SYNSEM feature of the functor contains the modified syntactic/semantic description. Note that the lexical entry itself contains no SYNSEM feature. When analyzing a surface form of the word *betreten* this functor is combined with the feature structure for *treten* (shown in fig.9) as argument.

At that stage the FUNCTOR|SYNSEM feature of *be-* is unified with the SYNSEM feature of *treten*. But there is still no value set for the SYNSEM feature. This is intended because it allows to disregard the composition in favour of a direct interpretation of the derived word. In our example we will find no entry for the stem *betreten* though. We therefore have to take the default approach which means unifying the result with the structure shown in fig.7.

Up to now our example was overly simplified because it did not take into account that *treten* has a second reading, namely *to kick*. The final lexical entry for *treten* is shown in fig.10.

But this second reading of *treten* cannot be used for deriving a second meaning of *betreten*:

(11) *Die Frau tritt [den Hund]OBJ.*
The woman kicks the dog.

(12) **Die Frau betritt [den Hund]OBJ.*

We therefore need to block the second compositional interpretation. This is achieved by an explicit entry for *betreten* in the lexeme lexicon which is shown in fig.11.

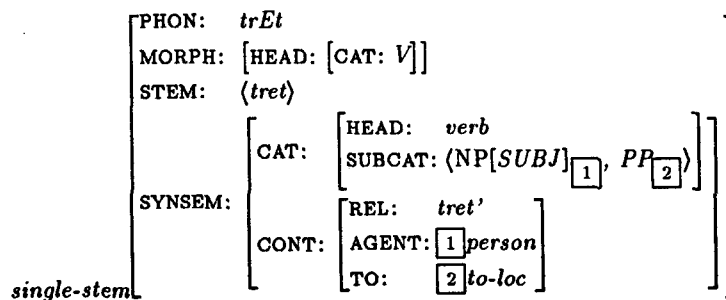


Figure 9: Lexical entry for verb *treten* (preliminary version)

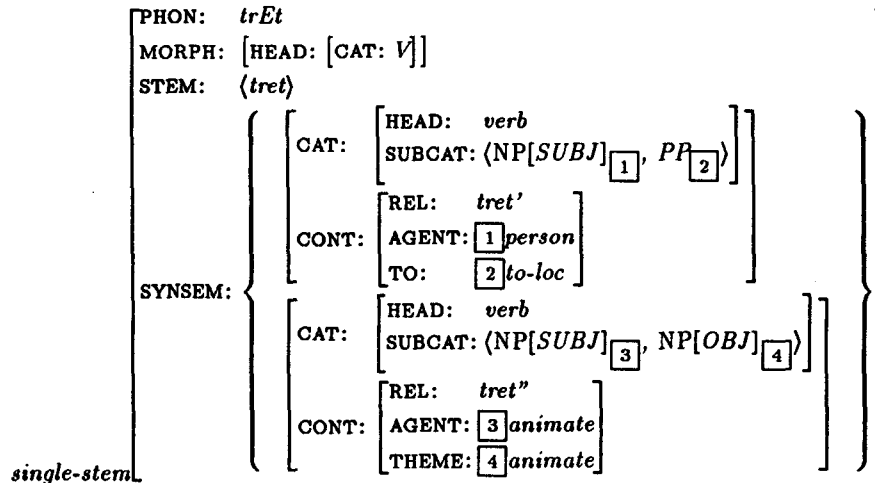


Figure 10: Lexical entry for *treten* (final version)

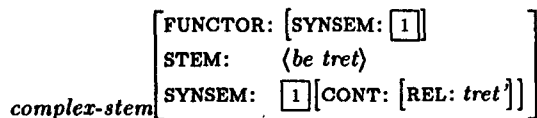


Figure 11: Entry for *betreten* in the lexeme lexicon

We now get the desired results. While both readings of *treten* produce a syntactic/semantic interpretation in the first stage the incorrect one is filtered out by applying the lexeme lexicon entry for *betreten* in the second stage.

8 Conclusion

In this paper we have presented a morphological analyzer/generator that combines an extended two-level morphology with a feature-based word grammar that deals with inflection as well as derivation. The grammar works on a lexicon containing both morphs and lexemes.

The system combines the main advantage of two-level morphology, namely the adequate treatment of morphophonology with the advantages of feature-based inheritance lexica. The system is able to automatically deduce a compositional interpretation for derived words not explicitly contained in the system's lexicon. Lexicalized compounds may be entered explicitly while retaining the information about their morphological structure. That way one can implement blocking (suppressing compositional readings) but is not forced to do so.

References

- [Backofen *et al.*, 1991] Rolf Backofen, Harald Trost, and Hans Uszkoreit. Linking Typed Feature Formalisms and Terminological Knowledge Representation Languages in Natural Language Front-Ends. In *W. Bauer, editor. Proceedings GI Kongress Wissensbasierte Systeme 1991*, Springer, Berlin, 1991.
- [Carpenter *et al.*, 1991] Bob Carpenter, Carl Pollard, and Alex Franz. The Specification and Implementation of Constraint-Based Unification Grammars. In *Proceedings of the Second International Workshop on Parsing Technology*, pages 143–153, Cancun, Mexico, 1991.
- [Daelemans *et al.*, 1992] Walter Daelemans, Koeraad De Smetd, and Gerald Gazdar. Inheritance in Natural Language Processing. *Computational Linguistics* 18(2):205–218, June 1992.
- [Evans and Gazdar, 1989] Roger Evans and Gerald Gazdar. Inference in DATR. In *Proceedings of the 4th Conference of the European Chapter of the ACL*, pages 66–71, Manchester, April 1989. Association for Computational Linguistics.
- [Heinz and Matiasek, 1993] Wolfgang Heinz and Johannes Matiasek. Argument Structure and Case Assignment in German. In *J. Nerbonne, K. Netter, and C. Pollard, editors. HPSG for German*, CSLI Publications, Stanford, California, (to appear), 1993.
- [Koskenniemi, 1984] Kimmo Koskenniemi. A General Computational Model for Word-Form Recognition and Production. In *Proceedings of the 10th International Conference on Computational Linguistics*, Stanford, California, 1984. International Committee on Computational Linguistics.
- [Krieger and Nerbonne, 1991] Hans-Ulrich Krieger and John Nerbonne. Feature-Based Inheritance Networks for Computational Lexicons. DFKI

- Research Report RR-91-31, German Research Center for Artificial Intelligence, Saarbrücken, 1991.
- [Nerbonne, 1992] John Nerbonne. Feature-Based Lexicons: An Example and a Comparison to DATR. DFKI Research Report RR-92-04, German Research Center for Artificial Intelligence, Saarbrücken, 1992.
- [Reinhard, 1991] Sabine Reinhard. *Adäquatheitsprobleme automatenbasierter Morphologiemodelle am Beispiel der deutschen Umlautung*. Magisterarbeit, Universität Trier, Germany, 1990.
- [Russell *et al.*, 1992] Graham Russell, Afzal Ballim, John Carroll, and Susan Warwick-Armstrong. A Practical Approach to Multiple Default Inheritance for Unification-Based Lexicons. *Computational Linguistics*, 18(3):311–338, September 1992.
- [di Sciullo and Williams, 1989] Anna-Maria di Sciullo and Edwin Williams. *On the Definition of Word*. MIT Press, Cambridge, Massachusetts, 1987.
- [Sproat, 1992] Richard Sproat. *Morphology and Computation*. MIT Press, Cambridge, Massachusetts, 1992.
- [Smolka, 1988] Gerd Smolka. A Feature Logic with Subsorts. LILOG-Report 33, IBM-Germany, Stuttgart, 1988.
- [Trost, 1991a] Harald Trost. Recognition and Generation of Word Forms for Natural Language Understanding Systems: Integrating Two-Level Morphology and Feature Unification. *Applied Artificial Intelligence*, 5(4):411–458, 1991.
- [Trost, 1991b] Harald Trost. X2MORF: A Morphological Component Based on Two-Level Morphology. In *Proceedings of the 12th International Joint Conference on Artificial Intelligence*, pages 1024–1030, Sydney, Australia, 1991. International Joint Committee on Artificial Intelligence.
- [Wahrig, 1978] Gerhard Wahrig, editor. *dtv Wörterbuch der deutschen Sprache*. Deutscher Taschenbuch Verlag, Munich, Germany, 1978.