Paradigm-Based Derivational Morphology

James Kilbury
Seminar für Allgemeine Sprachwissenschaft
Heinrich-Heine-Universität Düsseldorf
Universitätsstr. 1, 4000 Düsseldorf 1, Germany
e-mail: Kilbury@ze8.rz.uni-duesseldorf.de

Abstract

The paper sketches an approach to derivational morphology that is based on the notion of the paradigm and provides new possibilities for an integrated treatment of inflection and derivation. The principal innovation lies in the use of cross-subcategorization to describe derivational combinations. The notion of a derivational closure is also introduced. Advantages of the approach for computational morphology involve both the representation and the processing of derivational information. Primary attention is directed at derivational morphotactics.

Das Papier umreißt einen Ansatz in der Morphologie abgeleiteter Formen, der auf dem Begriff des Paradigmas beruht und neue Möglichkeiten für eine integrierte Behandlung der Flexion und Derivation eröffnet. Die wichtigste Innovation liegt in der Verwendung einer gegenseitigen Subkategorisierung, um Ableitungskombinationen zu beschreiben. Auch der Begriff einer derivationellen Hülle wird eingeführt. Vorteile des Ansatzes für die computerlinguistische Morphologie beziehen sich sowohl auf die Repräsentation wie auch die Verarbeitung derivationeller Informationen. Der Schwerpunkt der Aufmerksamkeit wird auf die derivationelle Morphotaktik gerichtet.

1 Introduction

The aim of this paper is to sketch an approach to derivational morphology that is based on the notion of the paradigm.[1] While recent studies reflect a renewed interest in the latter (cf Calder 1990), most follow grammatical tradition in applying paradigms primarily to the domain of inflection. The major exception is Gibbon (1991, 1992), whose conception of morphological paradigms, while more general than ours and developed independently, is close to that of this paper.

In addition to providing new possibilities for an integrated treatment of inflection and derivation, the approach we present here offers advantages for computational morphology with respect to both the representation and the processing of derivational information. While these techniques may prove also to be applicable in the domain of compositional morphology, this will not be discussed. Likewise, the treatment of morphophonemic or morphographemic alternations will only be briefly mentioned in this paper. Thus, the focus of attention is directed at derivational morphotactics, the arrangement of morphological elements in derived forms. Examples will be taken chiefly from German.

---

[1] Much of the material of this paper has been presented orally on a number of previous occasions, including the 1990 DGfS meeting in Saarbrücken and workshops in Bielefeld and Bochum in 1991. My particular thanks for discussion and suggestions go to Dafydd Gibbon, Ewan Klein, Petra Naerger, Ingrid Renz, and Richard Wiese, who, however, are not responsible for remaining errors. The work was supported by the DFG with a grant for the project "Simulation of Lexical Acquisition" (Ki 374/1).

An adequate approach to derivational morphotactics must meet a number of criteria. While it must capture the combinatory potential of affixes in order to model derivational productivity, it must also characterize derived lexemes as lexicalized combinations of elementary morphological units. Linguistic adequacy requires simple representations with minimal redundancy (cf Kiparsky 1982: 25), and the model must furthermore capture generalizations and explicitly describe the structure of the lexicon as an integrated whole, including the principles according to which individual lexical entries are *addressed* or located within the lexicon.[2] Computational adequacy demands representations and algorithms allowing for the efficient storage and processing of derivational information.

This is not the place to review discussions of morphology in general or the arguments presented for or against the lexicalist hypothesis (cf e.g. Spencer 1991) in particular. Much of the linguistic discussion as to whether derived forms should be "entered as wholes" in the lexicon or "derived by rule" is too vague to allow interpretation and evaluation within the context of computational linguistics. Whatever special assumptions a particular theory of language may make, it clearly must take account of the fact that derived forms such as German *Un-zu-ver-läss-ig-keit* 'unreliability', *Eigen-heit* 'peculiarity', or *Ent-eign-ung* 'expropriation' are combinations of elementary morphological units but also that such combinations may be *lexicalized* and associated with information not predictable from that of the constituent parts. In practice, computational approaches to derivation have tended to ignore one or the other aspect of the problem, either treating derived forms as rule-generated and entirely transparent, or else recording them individually and failing to use derivational generalizations to minimize redundancy in the representations. In view of the novelty of nonmonotonic devices in representation languages for lexical information (cf Evans/Gazdar 1990), it hardly comes as a surprise that most approaches have fallen victim to this dilemma, since, as can be seen below, the purely monotonic combination of derivational information necessarily leads to such a restriction of the possible solutions.

In contrast, our approach seeks to reconcile the apparent conflict between productive patterns and lexicalized combinations which constitutes the central problem of derivational morphology. The approach provides an efficient technique for addressing lexicalized derived forms indirectly and captures extension of the lexicon with new vocabulary in a natural fashion.

2 Representation of derivational information

We assume a morpheme-based lexicon containing a single entry for each root and affix morpheme. Affixes comprise bound morphemes of closed classes, whereas roots belong to open classes and may be free. Lexicalized derived forms are lemmatized under their constituent root morpheme so as to avoid direct addressing under the full forms. This is accomplished within a unification-based approach to grammar (cf Shieber 1986) making extensive use of structure sharing and cross reference between entries.

Although the direct orthographic or phonological addressing of morphemes is not the central concern of this paper, we will sketch the techniques employed, which are now largely conventional in computational morphology, in order to show the overall structure of the lexicon. Lexical entries for morphemes are associated with nodes in a discrimination network, the edges of which are labeled with orthographic or phonological segments, or complex representations thereof. Following the techniques of finite-state

[2] Of course, linguistic adequacy also requires a detailed account of many special phenomena which are not discussed here. The simple model of derivational morphotactics assumed in this paper is intended as a formal basis for extensions in future work.

morphology developed in particular by Kay (1983) and Koskenniemi (1983), the string representing the surface form of an item is matched with a path in the network leading to the node bearing the lexical entry of the form. Depending on the particular descriptive techniques chosen for a given language, the surface string is transduced in parallel with one or more underying strings (cf Kay 1987) encoding the lexical address of the entry. Alternatively, the surface alternants of a morpheme may be encoded in individual paths leading to a single node bearing the lexical entry, so that surface forms need not be transduced with underlying representations; in this case the discrimination network constitutes a graph rather than a tree. Clearly, the transduction technique is appropriate to capture *automatic* orthographic or phonological alternations (cf Hockett 1958, Kilbury 1976), which involve no grammatical conditioning but are forced by the phonological structure of the language, while grammatically conditioned alternations such as those involving suppletive forms like English *go* and *went* can best be handled with distinct paths. Moreover, we follow current practice (cf Trost 1990) in extending the original framework of two-level morphology to allow for unification operations to restrict the application of transduction rules or to label individual transitions in the discrimination network. Allomorphs encoded in alternative paths of surface segments leading to the same morpheme entry can thus be distinguished by different feature structures built up during traversal of the distinct paths.

Final-state nodes, which correspond to morphemes, bear lexical representations. These may take the form of feature structures, descriptions of feature structures consisting of Boolean expressions over path equations and templates, or DATR descriptions of feature structures (cf Kilbury/Naerger/Renz 1991). We assume in any case that a feature structure is associated directly or indirectly with each morpheme entry. The particular structuring of information within this feature structure will of course depend on the particular theoretical framework chosen. Whatever this may be, the representation will include information analogous to that involving subcategorization at the sentence-syntactic level to express the combinatory potential of the morpheme. The latter may involve general classes or individual lexical items, just as transitive verbs in general combine with a direct object to form a verb phrase or English *kick* combines specifically with *the bucket* to build a particular phraseolexeme with opaque semantics.

Since derivational structures in German can generally be viewed as *binary* combinations of stem and affix morphemes (cf Wiese 1988 and the *unary* rule stated below for conversion), it is convenient to represent German derivational morphotactics within the framework of categorial grammar. Work done since the development of PATR-II (cf Shieber 1986) has shown how categorial grammars can be encoded in unification-based formalisms (cf e.g. Uszkoreit 1986), and these techniques can be adopted for the representation of derivational information.

In one major respect, however, our representations constitute a substantial departure from conventional practice in categorial grammar. Fundamental to the latter is the distinction between *functor* and *argument* categories, whereby the latter may be *basic* or *complex*. We instead assign complex categories to all morphemes, both stems and affixes, and then formulate functional application rules in terms of a cross-subcategorization between the morphemes.

Affixes subcategorize for stem classes with their *arg(ument)* specification and describe the resulting derived stem with their *val(ue)* specification. They serve as functors encoding the syntactic and semantic information of transparent derivational constructions and thus constitute the repository of information about productive derivational processes.

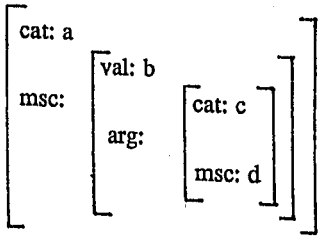
Roots and derived stems are likewise subcategorized, but for *particular* affixes rather than *classes* of affixes. In contrast to the affix representations, which encode the transparent and functionally determined information about derived stems, the stem representations encode precisely that information which has been

lexicalized and is not predictable from general information in the affix representations.[3] Therefore, the latter, together with templates that may be used in their definition, express linguistic generalizations about derivation and help to minimize redundancy in the lexicon.

The representation assigned to a derived stem inherits information from the *val* specifications of both the affix and the constituent stem representations, but the inheritance from the latter is *strict* and can be modelled with conventional unification, whereas that from the affix representation is *defeasible* or *nonmonotonic* (cf Kilbury, to appear). Since the lexicalized information stored under roots captures irregularity (such as semantic opacity and phonological idiosyncracy like lexical exceptions to umlaut), it must have precedence over the *default* information of affixes that characterizes transparent constructions, but the information from both sources must be combined in order to reconcile these conflicting aspects of derivation. Any formalization that takes account of these considerations must make use of devices for dealing with nonmonotonicity in lexical description like those that have recently been proposed (cf e.g. Evans/Gazdar 1990 and Bouma 1990), but which particular such devices are chosen is of secondary importance in the present paper. To simplify the formal exposition below, we will ignore technical questions involving reentrancy and simply use *overwriting* as presented by Shieber (1986: 60), where *A* => *B* may be roughly understood as meaning that feature structures *A* and *B* are to be unified, but if any specifications give rise to conflicts that would cause normal unification to fail, then the overwriting succeeds anyway, and the specifications of *B* win out over those of *A*.

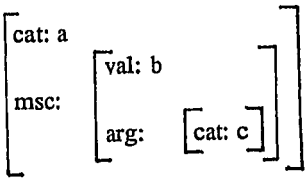
Before schematic rules for derivation can be formulated that are comparable to those for functional application in conventional categorial grammar, a detail involving the notion of cross-subcategorization must be clarified. We introduce the attribute *msc* for *morphological subcategorization* as opposed to syntactic subcategorization represented with the attribute *subcat*. As its value *msc* receives a feature structure with categorial specifications for *arg* and *val* or else the atom *none*. Given feature structures *S* and *A* for a constituent stem and affix, repectively, the *arg* specification of *A* expresses subcategorization for some *S'* such that *S'* subsumes *S*, and the *arg* specification of *S* expresses subcategorization for *A*. As it stands, this leads to cyclic feature structures. While these may in fact be desirable in the present context, we wish to avoid them for the sake of conventional implementations of PATR-II or similar formalisms. As a notational aid we therefore introduce a function Ψ such that, for any feature structure *F* containing a specification of morphological subcategorization α, $\Psi(F)$ subsumes *F* and contains all the information of the latter *except* any further morphological subcategorization specification nested within α. Thus, given the feature structure *F* represented in (1), $\Psi(F)$ is the feature structure represented in (2):

(1)



[3] Exceptionally, fossilized suffixes such as *-st* in *Dien-st* 'service' may be subcategorized for particular stems in order to capture a semantic relationship to the root, here *dien* V 'serve', and to *block* the production of nonoccurring forms like **Dien-ung*.

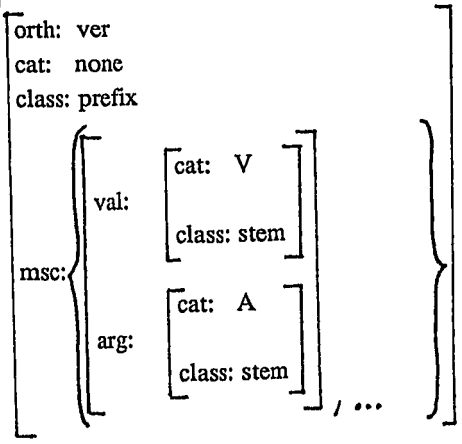
(2)



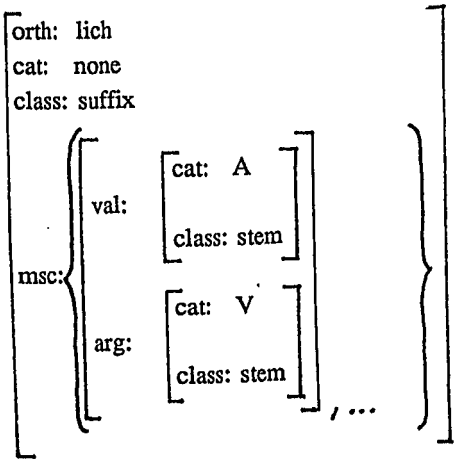
We then can formulate the cross-subcategorization of *S* and *A* so that *S* subcategorizes for $\Psi(A)$ and *A* for $\Psi(S')$.

Simplified lexical entries for the German morphemes *ver-*, *-lich*, and *wirk* (prefix, suffix, and root, respectively) are given in (3), where braces indicate disjunction:

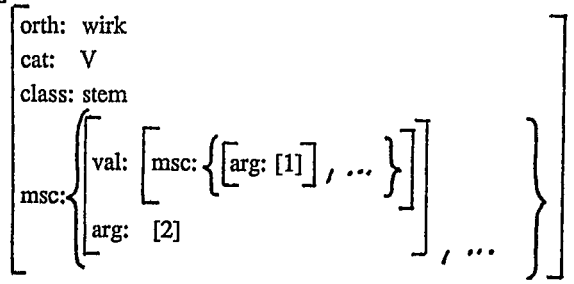
(3) [1]



[2]



[3]



Application rules for prefixation and suffixation are stated in (4):

(4) Rule {prefixation}
RESULT --> PREFIX STEM :
<PREFIX class> = prefix
<PREFIX msc arg> = Ψ(STEM)
<PREFIX msc val> = RESULT
<STEM class> = stem
<STEM msc arg> = Ψ(PREFIX)
<STEM msc val> => RESULT.

Rule {suffixation}
RESULT --> STEM SUFFIX :
<SUFFIX class> = suffix
<SUFFIX msc arg> = Ψ(STEM)
<SUFFIX msc val> = RESULT
<STEM class> = stem
<STEM msc arg> = Ψ(SUFFIX)
<STEM msc val> => RESULT.

An analogous *unary* application rule for *conversion* (to derive e.g. a nominal stem from the verbal stem *ess-* 'eat' without affixation) is formulated in (5):

(5) Rule {conversion}
RESULT --> STEM :
<STEM class> = stem
<STEM msc arg> = none
<STEM msc val> = RESULT.

As the above representations show,[4] the morphological subcategorization is specified with ***both*** the attributes *val* and *arg*, rather than the latter alone. This apparent departure from the conventional notion of subcategorization is necessary to capture dependencies between the subcategorized sister and mother categories and constitutes the key to our innovation in formal representation. The category assigned to a

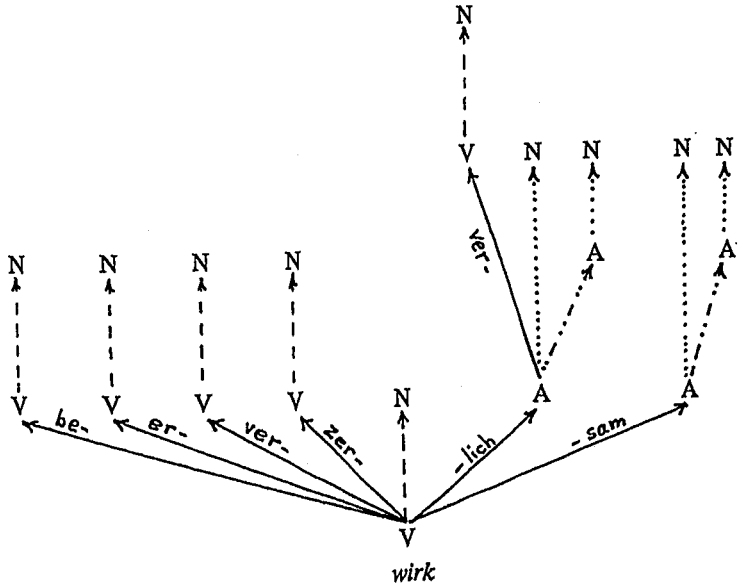
[4] In this formulation the nonmonotonicity seems to be extended from the description of the lexicon to analysis. This can, however, be avoided in a reformulation that locates the nonmonotonicity in template definitions for the morpheme classes. That is, the rules of (4) can themselves be viewed as generalizations about the lexicon.

derived stem is also subcategorized, so that morphological subcategorization involves a *recursive nesting* of *msc* specifications.

Since morphological subcategorization is recursively structured in this manner, all the complex stems derived from a given root (i.e. from a morphologically simple stem) are represented within the morphological subcategorization of the latter. We can call this set of lexemes R^* derived from a root R its *derivational closure*, which may in principle encompass an infinite number of potential forms but only a finite number of actual, lexicalized forms associated at least partially with idiosyncratic information that cannot be derived transparently from the given construction. This captures the notion of a *word family*, which is familiar within traditional lexicology but has not been treated systematically in synchronic structural linguistics.

A substantial subset of the lexicalized derivational closure of the German verbal root *wirk* 'work, operate, have effect on' is shown in (6):

(6)



-ung _ _ _ _
-keit .....
un- .._.._..

Except for the root node, which corresponds to the morphological root of the family, nodes of the tree represent the lemmata for derived stems located under *val* specifications. Edges correspond to *arg* specifications for the subcategorized affixes. Thus, only the lemma of the root is addressed directly in the orthographic or phonological discrimination network, whereas the lemmata of derived stems are addressed indirectly, via the primary address of the root together with the morphological subcategorization of the latter.

A node is needed for each lexicalized derived stem as a receptacle of idiosyncractic information, but all functionally determined, transparent information is inherited from the representations of the affixes, which are each recorded once. This radically reduces the amount of information represented under a root. In the

extreme case, a node of the derivational closure may bear no idiosyncratic information at all and simply attest the fact that a particular derived form happens to occur rather than not to occur, although the construction in which the form stands is entirely transparent.

The derivational closures of two distinct roots may be similar or even isomorphic in structure. In this case the information common to both can be represented in *templates*, from which the entries then inherit and which themselves build an inheritance network. In this way the amount of information stored in an individual derivational closure is still further reduced. Such templates characterizing a set of concrete derivational closures constitute *derivational paradigms* parallel to the inflectional paradigms of inflectional morphology. Inheritance relations between derivational paradigms can be captured with representation languages such as DATR (cf Evans/Gazdar 1990), so that generalizations about derivational structure can be explicitly formulated which, to our knowledge, have not even been informally discussed up to now in the linguistic literature.

3 Parsing morphologically derived forms

Corresponding to the indirect addressing of derived lexemes in our representations, parsing the forms can be modelled as involving two analytic stages. In the first stage, the surface form is parsed into a *regular expression,* familiar from the theory of finite-state automata (FSAs) and regular formal languages. Although its precise form is undoubtedly language-specific, we assume that for each language a general schema for derived forms can also be stated as a regular expression. For German this general schema can be stated as in (7):

(7) $\text{Prefix}^* \ \text{Root} \ \text{Suffix}^* \ (\text{Ending}_1) \ (\text{Ending}_2)$

That is, a derived form in German consists of zero or more prefixes followed by a single root, which in turn is followed by zero or more derivational suffixes and at most two inflectional endings.[5] In the course of the left-to-right decomposition into a regular expression, the prefixes of a derived form are pushed onto a stack (first-in, last-out) and the suffixes into a queue (first-in, first-out). After this first stage of parsing, the innermost affixes are both directly accessible. Once the root has been parsed, the node in the discrimination network has been found at which the derivational closure is represented in which the lemma for the derived form is embedded.

In the second stage of parsing, innermost affixes are successively taken from the stack and queue, and the path of morphological subcategorization specifications in the root entry is traversed in parallel. The lemma for the derived stem is found when both devices are empty.

In the case of lexicalized derived stems, each POP operation is determined by the *msc* specification of the current stem, although the principle of cross-subcategorization also ensures compatibility of the stem with the subcategorized affix. If the *msc* specification of the current stem does not call for one of the current TOP affixes, then either (1) the derived form is nonlexicalized and novel, and the further analysis is driven by the *msc* specifications of the affixes, or (2) apparent structural ambiguity in the derived form led to an incorrect POP operation, and backtracking must be initiated, or (3) the form is ungrammatical and cannot be built up on the basis of the morphological subcategorization information of the constitutent

[5] Two inflections occur e.g. in the form *Brett-er-n* 'boards' (dative plural). Sequences of affixes are in fact more highly constrained in German than the Kleene star suggests, but this need not be taken into account here.

morphemes. Internal sandhi (i.e. morphological alternation) phenomena are checked in parallel with the assembly of the derived stem. In case (1) the number of lexicalized affixation steps that precede shift of assembly control to the affix subcategorization determines the degree of *transparency* of the derived stem.

A special case (4) arises when the root cannot be parsed, i.e. when there is no orthographic or phonological address in the discrimination network that records an entry for the root. The search for the path addressing a root runs in parallel with a finite-state automaton defining the orthographic or phonological structure of root morphemes. This permits the morphological parser to postulate a form for roots not recorded in the lexicon. The decomposition into potential derivational suffixes and inflectional endings is then continued, and the constituent structure of the unknown derived stem is assembled according to the *msc* specifications of the affixes as in case (1) above.

Note that both stages of the analysis are nondeterministic and may find alternative solutions. Since the morphological parser attempts to find a maximal morphemic decomposition, the first stage will first incorrectly identify a prefix *ver-* in *Versifizierung* 'versification' before backtracking to find the correct decomposition (cf Black et al. 1991). This appears to us to embody a correct modelling of linguistic competence. Likewise, assembly in the second stage of analysis will identify structurally ambiguous stems to which more than one internal constituent structure can be assigned.[6]

While the two-stage model of analysis just outlined is useful for expository purposes, implementations would undoubtedly be based on a simplification combining the stages. After a partial-decomposition stage in which the potential prefixes are isolated and pushed onto a stack, the further decomposition into morphs and the assembly to derived stems would proceed together. Assuming that the cyclicity of phonological rules presents no problems for splitting off the prefixes, this modified strategy permits a direct morphological analysis of derived forms from left to right.

Although the representations described here were chosen primarily with regard to analysis, it should be clear that our approach can be extended to synthesis and in fact *includes* the latter as a part of analysis since complex forms are reassembled during parsing. A full treatment of synthesis would additionally require that lemmata be accessible on the basis of semantic and pragmatic information. Note that the treatment of blocking mentioned above in footnote 3 is important for generation.

For the sake of completeness we briefly summarize our treatment of inflection. While morphs constituting inflectional endings are isolated in the decomposition stage, the representation of these endings differs from that of other affixes and of roots because we assume that the former are not subcategorized and have no morpheme-like nodes bearing lexical representations. Instead, each root and derived stem, if subject to inflection,[7] is marked for an *inflection class* or *paradigm* implemented as a *continuation class* as in Koskenniemi (1983). Within the FSAs implementing inflectional paradigms, grammatical markings on the edges in paths associated with morphs found in the decomposition stage serve to distinguish and identify the grammatical function of the endings. This captures the fact that the inflectional endings of languages like German cannot usefully be analyzed in isolation and that their grammatical features must be defined within a system of oppositions building a paradigm. In the case of derived stems, the inflection class will generally be inherited from a constituent affix. Likewise, syncretism within and relations between paradigms are captured with inheritance mechanisms.

---

[6] Although globally unambiguous, the verbal stem *ver-wirk-lich* is assembled nondeterministically because the root *wirk-* is subcategorized for both *ver-* and *-lich*, so that local structural ambiguity arises.

[7] Note that this allows us to represent stems which exceptionally cannot be inflected but which are needed for other derived forms. *Back-formations* arise when the representations of such stems, which by default should be capable of inflection, are systematically simplified to allow inflection.

4 Conclusion

We have outlined an approach to representation and analysis for derivational morphology that rests crucially on a notion of cross-subcategorization and the nonmonotonic combination of information. The derivational closure of a root consists of the family of lexicalized forms derived from it. Taken together, the techniques presented allow us formally to capture the relations between productive and transparent derivational constructions, on the one hand, and lexicalized and opaque information, on the other. Furthermore, unknown derived lexical items, for which either an affixation or the root itself is not lexicalized, can be handled without the introduction of any special devices except a general description of the orthographic or phonological structure of roots, which is independently desirable. Although morphological composition involves further complications that have not been touched on in this paper, the application of the techniques to this area will be the subject of future investigation.

References

Black, Alan W. / van de Plassche, Joke / Williams, Briony (1991) Analysis of Unknown Words through Morphological Decomposition, *Proceedings of the 5th EACL Conference*, 101-106.

Bouma, Gosse (1990) Defaults in Unification Grammar, *Proceedings of the 28th ACL Conference*, 165-172.

Calder, Jonathan (1989) Paradigmatic Morphology, *Proceedings of the 4th EACL Conference*, 58-65.

Evans, Roger / Gazdar, Gerald (eds) (1990) *The DATR Papers: February 1990* (= *Cognitive Science Research Paper* 139). School of Cognitive and Computing Sciences, University of Sussex, Brighton, England.

Gibbon, Dafydd (1991) Lexical Signs and Lexicon Structure: Phonology and Prosody in the ASL-Lexicon (ASL-Memo-20-91/UBI). University of Bielefeld.

Gibbon, Dafydd (1992) ILEX: A Linguistic Approach to Computational Lexica, 32-53, Ursula Klenk (ed), *Computatio Linguae*. Stuttgart: Franz Steiner Verlag.

Hocket, Charles F. (1958) *A Course in Modern Linguistics*. New York: MacMillan.

Kay, Martin (1983) When Meta-rules are not Meta-rules, K. Sparck-Jones and Y. Wilks (eds), *Automatic Natural Language Processing*. Chichester: Ellis Horwood.

Kay, Martin (1987) Nonconcatenative Finite-State Morphology, *Proceedings of the 3rd EACL Conference*, 2-10.

Kilbury, James (1976) *The Development of Morphophonemic Theory*. Amsterdam: Benjamins.

Kilbury, James (to appear) Strict Inheritance and the Taxonomy of Lexical Types in DATR.

Kilbury, James / Naerger, Petra / Renz, Ingrid (1991) DATR as a Lexical Component for PATR, *Proceedings of the 5th EACL Conference*, 137-142.

Kiparsky, Paul (1982) Lexical Morphology and Phonology, I.-S. Yang (ed), *Linguistics in the Morning Calm*, 3-91. Seoul: Hanshin.

Koskenniemi, Kimmo (1983) Two-level Model for Morphological Analysis, *Proceedings of IJCAI-83*, 683-685.

Shieber, Stuart M. (1986) *An Introduction to Unification-based Approaches to Grammar* (= *CSLI Lecture Notes* 4). Stanford, Calif.: CSLI.

Spencer, Andrew (1991) *Morphological Theory*. Oxford and Cambridge, MA: Blackwell.

Trost, Harald (1990) The Application of Two-level Morphology to Non-concatenative German Morphology, *Proceedings of COLING-90, Vol. 2*, 371-376.

Uzskoreit, Hans (1986) Categorial Unification Grammars, *Proceedings of COLING '86*, 187-194.

Wiese, Richard (1988) *Silbische und lexikalische Phonologie*. Tübingen: Niemeyer.