

PARADIGMATIC MORPHOLOGY

Jonathan Calder
University of Edinburgh
Centre for Cognitive Science
2 Buccleuch Place
Edinburgh
Scotland
EH8 9LW

ABSTRACT

We present a notation for the declarative statement of morphological relationships and lexical rules, based on the traditional notion of *Word and Paradigm* (cf Hockett 1954). The phenomenon of blocking arises from a generalized version of Kiparsky's (1973) *Elsewhere Condition*, stated in terms of ordering by subsumption over paradigms. Orthographic constraints on morphemic alternation are described by means of *string equations* (Siekman 1975). We indicate some criticisms to be made of our approach from both linguistic and computational perspectives and relate our approach to others such as Finite-State Morphology (Koskenniemi 1983), DATR (Gazdar and Evans 1989) and object-oriented morphophonemics (de Smedt 1984, Daelemans 1988). Finally, we discuss the questions of whether a system involving string equations allows a reduction to finite-state techniques.

1 Introduction

A common assumption in linguistics is that the phonological, morphological and orthographic statements are most appropriately phrased in a fundamentally procedural way, (see for example Hoeksma and Janda 1988). Morphological analysis under the rubric of finite-state morphology (Koskenniemi 1983) has arguably tended to support the view that morphological alternation is best described by stating procedures for the destructive alteration of orthographic units. At the very least, it appears to have led to the view that morphological descriptions should be restricted to those with an immediate interpretation in terms of the operations of finite-state transducers.

In this paper, we present a notation for the

declarative statement of morphological relationships and lexical rules, based on the traditional notion of *Word and Paradigm* (WP, Hockett 1954, see also Anderson 1982). The phenomenon of blocking arises from a generalized version of Kiparsky's (1973) *Elsewhere Condition*, stated in terms of ordering by subsumption over paradigms. Orthographic constraints on morphemic alternation are described by means of *string equations* (Siekman 1975).

We will first give a brief introduction to string equations and the other formal devices used in our model, namely lexical entries and rules, grammatical properties and paradigms. We give example paradigms and show how our interpretation of paradigms leads to the phenomenon of blocking. We will then indicate some criticisms to be made of our approach from both linguistic and computational perspectives. We discuss relations between our proposals and the approaches of Finite-State Morphology (FSM, Koskenniemi 1983), DATR (Gazdar and Evans 1989) and object-oriented morphophonemics (de Smedt 1985, Daelemans 1988). One important question in the light of current work in morphology is whether a system involving string equations allows a reduction to finite-state techniques. We review some possible answers to this question.

2 Components of the Model

2.1 String Equations and String Unification

This introduction is based on Siekman (1975). A string σ is a sequence of elements drawn from a finite alphabet C combined by the associative operator $+$, representing the concatenation of strings

A *string specification* or *string form* is a sequence possibly containing variables drawn from the set of variables V , disjoint from C . Omission of the operator $+$ increases legibility as shown in the right hand column of (1) which gives examples of strings (a,b) and string specifications (c-e) and where lower case alphabets represent elements of C and upper case alphabets elements of V .

- | | | | |
|-----|----|-----------|-------|
| (1) | a. | w+a+l+k+s | walks |
| | b. | s+ö+r | sör |
| | c. | A | A |
| | d. | W+s | Ws |
| | e. | k+V+t+V+b | kVtVb |

String specifications are *partial descriptions* of strings. As with the standard use of unification in computational linguistics (Shieber et al 1983, Pereira 1987), we may take two partial descriptions to describe the same object. We use *string equations* to represent this situation. The examples in (2) show string equations and the assignments of values to variables which satisfy the equations.

- | | | |
|-----|-----|-------------------------|
| (2) | a. | walks = Ws |
| | a'. | W/walk |
| | b. | sörAk = XYZYW |
| | b'. | A/ö, Y/ö, X/s, Z/r, W/k |
| | c. | kVtWb = CiDaE |
| | c'. | V/i, W/a, C/k, D/t, E/b |

The operation which determines the assignments of values to variables (equivalently, determines the most general unifying substitution for a given problem) is *string unification*. While no complete, terminating algorithm for the general case of string unification exists, the class of problems described by Siekmann (*op. cit.*, section 4.3.3.2) as $P_{0.5}$, that is where repeated variables are only permitted on one side of an equation, are decidable and have only finitary substitutions (see also Siekmann 1984). Whether or not an equation falls within $P_{0.5}$ is easily determined. The examples in (2) and in the rest of this paper all fall within this class. We will refer to the result of applying a unifying substitution to either side of a string equation as the *unification* of the two

string specifications in question.

The relation of subsumption defines a partial ordering on string specifications. A string specification S subsumes another S' ($S \sqsubseteq S'$) if all ground instances (i.e. instances that contain no variables) of S' are also instances of S . Equivalently, $S \sqsubseteq S'$ if the unification of S and S' is S' ($S' \sqcup_S S = S'$). If S and S' are inconsistent, $S \sqcup_S S'$ is undefined.

It is worth noting that the use of string unification is widespread in the field of automatic theorem proving, as an extension of standard resolution techniques and typically as an instance of what Plotkin (1972) terms "building in equational theories", that is the extension of standard unification algorithms by axioms such as associativity and the development of a normal form for the resulting theory (Plotkin *op.cit.* p74).

2.2 Lexical entries

A *lexical entry* $S:P$ associates a string S (i.e. S contains no variables) and a set of grammatical properties P . We will here treat grammatical properties as atomic. (Their ultimate interpretation is intended to be similar to that of *templates* in PATR-II (Shieber et al 1983)). A *lexical specification* $\sigma:\phi$ subsumes another $\sigma':\phi'$ iff $\sigma \sqsubseteq \sigma'$ and $\phi \leq \phi'$ where \leq represents the partial ordering over sets defined by the relation of set inclusion. (In other words, $\sigma':\phi'$ contains at least as much orthographic and grammatical information as $\sigma:\phi$). The lexicon consists of a finite set of lexical entries.

2.3 Lexical rules

A *lexical rule* is a triple $\langle \text{Name}, \text{IS:IP}, \text{OS:OP} \rangle$, representing a mapping between a set of "input" properties IP and a set of "output" properties OP . The interpretation of a rule with respect to grammatical properties is as follows:

- (3) Given a lexical item $S:P$, and a lexical rule, $\langle \text{LR}, \text{IS:IP}, \text{OS:OP} \rangle$, as before, $L R$ relates P to another set of properties P' (its "output") in the following way:
- $$P' = (P \setminus \text{IP}) \cup \text{OP}$$
- where $\text{OP} \leq P'$ and $\text{IP} \leq P$.

\setminus is set complement. Equivalently $P = (P' \setminus \text{OP})$

$\cup IP^1$. The use of set complement allows a general *ceteris paribus* statement². That is, properties not mentioned in the rule are unchanged. The relationship between the string specifications IS and OS is mediated by a *paradigm*.

2.4 Paradigms

A paradigm is a quadruple, $\langle \text{Name}, \sigma:\phi, [\text{LR}_1 \dots \text{LR}_n], [\text{S}_1 \dots \text{S}_n] \rangle$, $n \geq 1$, which relates string forms σ and S_i via the lexical rule LR_i under conditions ϕ where the set of string variables in σ and S_i are not disjoint. S_i is a *derived string form*. Any variables in S_i also occur within σ (this restriction will be reformulated shortly). **Name** is the (unique) name of the paradigm. $\sigma:\phi$ is the lexical specification of the paradigm. (Alternatively, it is the underspecified *word* whose behaviour the paradigm describes). The interpretation of a paradigm is given in (4).

- (4) If a paradigm $\langle \text{Name}, \sigma:\phi, \text{LR}, \text{S} \rangle$ is applicable, lexical items S:P and $\text{S}_i:\text{P}'$ are related by lexical rule $\langle \text{LR}_i, \text{IS:IP}, \text{OS:OP} \rangle$, with P, P' , as in (3).

For a paradigm $\pi \langle \text{Name}, \sigma:\phi, \text{LR}, \text{S} \rangle$ to be applicable to a lexical item S:P , two conditions must hold:

- (5) a. $\sigma \sqsubseteq \text{S}$ and $\phi \leq \text{P}$.
 b. There is no paradigm $\pi' \langle \text{Name}', \sigma':\phi', \text{LR}', \text{S}' \rangle$ such that $\sigma \sqsubseteq \sigma', \phi \leq \phi', \sigma' \sqsubseteq \text{S}$ and $\phi' \leq \text{P}$.

(5a) requires that the lexical specification of the paradigm subsume the lexical item. (5b) requires that there be no paradigm whose lexical specification is more specific than that of π which is also applicable to the lexical item. The effect of (5b) is to enforce a generalized *Elsewhere Condition* (Kiparsky 1973), under which a morphological operation is only allowed if there is no more specific statement which also holds. This also captures the notion of nested

¹In fact, for this relationship to hold, we have to add the following restrictions over the properties mentioned in the rule and lexical item: $\text{P} \cap \text{OP} \leq \text{IP}$ and symmetrically $\text{P}' \cap \text{IP} \leq \text{OP}$. We thank Marc Moens for this observation.

²Related proposals are made by Flickinger (1987, ch. 5).

generalizations or *stratification* (Flickinger et al 1985 Gazdar and Evans 1988). Given a lexicon containing a finite number of lexical entries, paradigms and lexical rules, the set of *lexical items* is defined as the closure of the lexicon under the application of lexical rules mediated by the paradigms. Nothing in the basic formalism we propose constrains this closure to be finite or requires that the set of lexical items contain the set of lexical entries. Note that the restrictions we have imposed on lexical entries and variables in derived strings imply that, in the lexical items that result from the application of paradigms, there will be no string variables.

3 Abbreviatory conventions

We introduce three abbreviatory conventions, the first two trivial, the third less so. First, as lexical rules always make reference to input and output string forms and these forms can be determined by reference to the paradigm that relates them, we do not need to state string forms in lexical rules. This leads to the concrete syntax (IP, OP as before):

- (6) `lexical_rule(Name, IP \rightarrow OP)`

The second convention allows us to state more complex constraints on string forms. In a paradigm $\langle \pi, \sigma:\phi, \dots \rangle$, we allow arbitrary equations over string forms to be included in ϕ , including negative and disjunctive constraints³, and a syntax for allowing the expression of character classes (effectively these are just a special case of disjunctive constraint). This allows statements of the following kind, where $+$ represents string concatenation, \leq represents an inequality between strings and PROLOG conventions for marking variables are followed.

- (7) `Stem+Affix = Word,`
`Stem = Prefix+C+V,`
`V \in vowels, C \in consonants, V \neq y, Affix = d`

(7) might be taken to describe the behaviour of vowel-final verbs under affixation of the past tense morpheme. Note that statements which do not constrain the value of σ do not take part in the calculation of subsumption relations over paradigms.

³We make the restriction that any such constraints do not contain variables.

We also have to revise our restrictions on the occurrences of variables in derived string forms given in 2.4. Any variable in a derived string form S_i must either occur in the string specification σ of the paradigm or must be equated, directly or indirectly, to some form consisting of variables drawn solely from σ and ground forms.

The third convention is considerably more complex and effects a rapprochement between our scheme and those of default logics for lexical description (Gazdar and Evans 1988, Flickinger 1987) and object-oriented morphophonemics (Daelemans 1988). Given a paradigm $\langle \pi, \sigma: \phi, LR, S \rangle$, if there is only one directly subsuming paradigm $\langle \pi', \sigma': \phi', LR', S' \rangle$, and for some i , $LR_i = LR'_i$ and $\sigma \cup \sigma' \rightarrow S_i = S'_i$ (i.e. we would get the same result if we used either π or π'), we are allowed to omit the references to LR_i and S_i in π . In other words, we allow the inheritance of a string form and associated lexical rule from the more general paradigm. In the case of n directly subsuming paradigms, the same convention applies if $\sigma \cup \sigma_1 \dots \cup \sigma_n \rightarrow S_i = S'_{i,1} \dots = S'_{i,n}$. That is,

multiple inheritance must be consistent. In the case where this condition doesn't hold, we allow the equivalence of inheritance by delegation (Daelemans *op. cit.*), so enforcing orthogonality of inheritance (Touretzky 1986); corresponding to some LR_i , we allow S_i to be of the form $\text{paradigm}(\pi'')$, where π'' is the name of a paradigm, such that $\pi'' \sqsubseteq \pi$, with the interpretation that the paradigm π'' relates σ and S_i via LR_i .

4 Example paradigms and lexical rules

We are now in a position to give some example paradigms. These have the concrete syntax:

(8) `table(Name, String:Properties, LR, S)`

where **Properties** may also include string equations and **LR** and **S** are lists of names of lexical rules and of string forms respectively, subject to the conventions described above.

In Table 1 below, (9) is the most general paradigm for English verbs. (10), (11), and (12) are instances of that paradigm under (5b) above. (12) is also an instance of (11).

-
- (9) `table(verb, Verb:[verb, base, Past = Verb+ed],
 [base 3sg non3sg past_participle past passive progressive]
 [Verb Verb+s Verb Past Past Past Verb+ing])`
- (10) `table(verb_age, age:[verb, base],
 [progressive]
 [ageing])`
- (11) `table(verb_strong, S+in+C:[verb, base,strong, Past=S+un+C, C= {g,k}],
 [past_participle past passive]
 [Past S+an+C Past])`
- (12) `table(verb_bring, bring:[verb, base,strong, Past = brought],
 [past_participle past passive]
 [Past Past Past])`
-

Table 1: Example paradigms

The abbreviatory mechanisms allow us roughly the same amount of compaction for these descriptions as found in Daelemans' (1988) approach. Lexical rules that might be associated with such paradigms are:

(13) lexical_rule(3sg,
[verb, base] → [verb, finite, 3sg])

lexical_rule(non3sg,
[verb, base] → [verb, finite, non3sg])

5 Criticisms of the framework

There is one major criticism of our approach from a linguistic point of view, namely that in cases such as "ageing", there is no reason why the form of the participle ending "ing" should be the same in all verbal paradigms. Likewise we cannot make the generalizations that passive and past participle forms are identical in every verbal paradigm and that the orthographic behaviour of the verbal singular affix and nominal plural affix is identical. Defining subsumption on the basis of lexical specifications of paradigms alone leads to a very simple statement of the conditions of inheritance of derived string forms, but disallows the possibility of inheritance of partial derived string forms. The restriction of inheritance by delegation or stipulation to subsuming paradigms, while natural, is not motivated by more general considerations.

This problem becomes much more obvious and acute in analysing non-concatenative morphology, as in Semitic (McCarthy 1981). It is not the intercalation of the consonantal roots and vocalic melodies which leads to difficulties, as this is easily described in our framework. Rather, the problem lies in having to choose which of the root and melody should be expressed as the word with which a paradigm is associated. On the one hand, traditional grammar would suggest that the consonantal root has some claim to this status. However, there are clearly relationships between the vocalic melodies which indicate syntactic regularities on the basis of Ablaut (McCarthy 1981, p403), and these regularities cannot be captured if we choose the consonantal root as the paradigmatic word and disallow inheritance of partial derived string forms. In any case, such regularities should presumably be stated independently of any

roots with which they happen to combine.

The above criticism provides an interesting illumination of Matthews' (1974, p163) claim that different styles of morphological analysis are required by different language types, word and paradigm morphology being particularly suited to inflecting rather than isolating and agglutinating languages. Current work is investigating how we may alter some of the assumptions in the definitions in section 2, to allow for some degree of parametricity in the languages that such systems may describe.

A second criticism is both computational and linguistic. String unification is a very powerful operation and, while it is arguable that strings in our sense and orthographic constructs are indeed the same kind of object, one may justifiably have reservations about introducing string equations into linguistic description. The resolution of this point awaits further work on the formal properties of finitary theories. We return to this point in our discussion of phonological theory below. The computational aspects of this problem will have less force if we can show that there is a reduction from descriptions involving string unification to some less powerful mechanism such as finite state transduction. This point is also discussed further below.

6 Comparison with other frameworks

That our approach is more general than standard FSM is clear from the fact that string variables can represent an unbounded amount of orthographic material. In this way, we can, for example, model unbounded reduplication. The problematic cases of Finnish ambifixation described by Karttunen (1983 p181, citing Koskeniemi) are handled straightforwardly, although this raises immediate questions about the formalism's interpretation in terms of finite-state transduction, discussed shortly. Görz and Paulus's (1988) algorithm for the location of stem vowels in German which undergo Ablaut can be reconstructed declaratively. Our approach also solves the problem noted by Bear (1988) of the overgenerality of rules in an FSM system. He introduces the mechanism of a negative rule feature to prevent the application of an epenthesis rule which would otherwise insert an "e" before the suffix "s" in

the case of words such as "banjo" and "tango". The need for negative rule features and their quasiprocedural implications are avoided in the system we propose. The following paradigm correctly states the facts and will apply only in the case of those items that violate the subgeneralization concerning nominal plurals in "o".

(14) table(piano,
 S:[noun, singular,
 S = {piano, piccolo, ...}],

 [singular, plural]
 [S, S+s])

Free variation, such as that between "o" and "oe" forms in words like "cargo", is treated analogously by allowing the derived string specification to be $S+\{e, \emptyset\}+s$ in this case (where \emptyset represents the empty string), although this obviously fails to capture the fact that the alternants are precisely those found in the most closely related paradigms. Finally, having the level of properties over which to state generalizations means that our lexical representations of strings are not cluttered with non-realizable diacritic characters whose function is simply to trigger morphological processes (cf Bear 1988).

The main advance we would claim for our system is that we have provided a calculus for orthographic forms, bringing the treatment of orthography within the same kind of logical framework now accorded to our treatment of semantic and syntactic objects. The fully declarative interpretation of our system and the similarity of statements within it to work by Daelemans (1988) offer a way of giving a formal treatment of object-oriented morphophonemics.

Finally, recent work by Gazdar and Evans (1988) may offer techniques for extending the formalism to more complex grammatical descriptions, in line with many current views on the nature of syntactic categories as recursively structured objects. Current work is examining this possibility.

7 Reduction to finite-state transducers

While the declarative nature of our system means that it is not tied to a particular computational interpretation, the fact that the closure of the lexicon

may be infinite argues in favour of an interpretation that does not presuppose computation of that closure, and considerations of efficiency in favour of one in terms of finite-state transduction (Koskenniemi 1983)⁴. It is immediately clear that heavy restrictions have to be placed on systems involving string equations in order to have any chance of a reduction to finite-state transduction at all. One difficulty will lie in representing paradigm subsumption induced by non-orthographic properties. The assumption that there is only a finite set of such properties would make this problem easier. (We assume the subsumption relations engendered by orthographic properties are readily handled by the standard notion of priority of transition in FSTs). A much graver problem lies in eliminating non-finite-state constructions. Example (15) below exhibits (at least) context-freeness:

(15) table(weak_verb,
 S+en:[verb, weak, root],

 [... , past_participle, ...]
 [... , ge+S+et, ...])

Here any string may be substituted for S. If we allow *past participle* to represent the null lexical rule, we may generate the string set $(ge)^n S(et)^n$ which is clearly homomorphic to the context-free language $a^n b^n$. Similar demonstrations can be given of context-sensitive⁵ and even more powerful languages. In order for the reduction to go through in this case, we should have to demonstrate the finitariness of substitutions for S. This is clearly impossible in the general case. The kinds of restriction to be imposed would include at least the prohibition of copying over unbounded domains and of affixation at both margins.

Under the optimistic assumption that appropriate restrictions can be found, we might proceed by computing tree-structured lexicons on the basis of lexical entries and of affixes introduced under paradigms. Continuation classes, and possibly further

⁴This section and the following have benefited greatly from discussions with Mike Reape.

⁵Such as the string specification WW , where W ranges over strings from some alphabet and which clearly shows context-sensitivity (Aho and Ullman 1972, p198).

partitions and duplications of the lexicons, could be computed on the basis of the properties mentioned in paradigms and lexical rules. Information from the distribution of derived forms with respect to string specifications in paradigms could then be used to construct the FSMs that mediate surface and lexical tapes. Problematic cases might only be handled by indicating an ordering over paradigms where context-freeness is implicated, effectively indicating that those paradigms may only apply to non-derived forms and allowing incompleteness in the computation of the closure of the lexicon.

The above paragraphs are mostly speculation. As Gazdar (1985) notes, it is not certain that morphological phenomena in natural language are best characterized by finite-state devices. Depending on one's view of the data cited by Culy (1985) and the ambifixation cases mentioned above, the formal power of our framework might be interpreted as a virtue rather than a vice and future work should also look at introducing (at least) context-free devices into our computational interpretation of morphology. Unsurprisingly, this is an area for further research.

8 Implementation

All aspects of the system described above have been implemented, primarily by Mike Reape. The implementation of lexical rules differs somewhat from the presentation given here, in that we allow the association of a PATR-II style lexical rules with the lexical rules we describe above and, as such, lexical rules may perform arbitrary mappings over feature structures. The work considerably extended published string unification algorithms to handle identity and arbitrary constraints over string forms. The system is a subcomponent of a polytheoretic lexicon system (Calder and te Lindert 1987, cf. also Zaenen 1988) and is currently being used to generate English lexical entries for use with Unification Categorical Grammar (Zeevat et al 1986, Calder et al 1986). The system generates lexical entries off-line; the lexicons used to date have provably finite closures under the application of lexical rules. Current work is focused on the computational interpretation of paradigms and on descriptive work in languages other than English.

ACKNOWLEDGEMENTS

This work is supported by ESPRIT project P393: ACORD "The Construction and Interrogation of Knowledge Bases using Natural Language Text and Graphics". Thanks are due in particular to Mike Reape and also to Ewan Klein, Claire Gardent, Enric te Lindert, Marc Moens, Antonio Sanfilippo and Henk Zeevat for discussions and criticism of previous proposals. All errors are of course my own.

References

- Aho, A. V. and Ullman, J. D. (1972). *The theory of Parsing, Translation, and Compiling Volume 1: Parsing* Englewood Cliffs Prentice-Hall.
- Anderson, S. R. (1982) "Where's Morphology" *Linguistic Inquiry* 13.4 pp571-612
- Bear, J. (1988) "Morphology and two-level rules and negative rule features". In *Proceedings of the 12th International Conference on Computational Linguistics and the 24th Annual Meeting of the Association for Computational Linguistics*, Budapest Hungary, 22-27 August, 1988, pp28-31
- Calder, J. and te Lindert, E. (1987) "The Protolexicon: Towards a High-Level Language for Lexical Description". In Klein E. and Benthem, J. (eds.) *Categories Polymorphism and Unification*, pp355-370 Edinburgh and Amsterdam: Centre for Cognitive Science, University of Edinburgh and Institute for Language, Logic and Information, University of Amsterdam.
- Calder, J., Moens, M. and Zeevat, H. (1986) A UCC Interpreter. ESPRIT PROJECT 39: ACORD; Deliverable T2.6, Centre for Cognitive Science, University of Edinburgh
- Culy, C. (1985) "The complexity of the vocabulary of Bambara" *Linguistics and Philosophy* 8 pp345-351.
- Daelemans, W. (1988) A Model of Dutch Morphophonology and its Applications. A Memo No. 88-8, Artificial Intelligence Laboratory, Brussels, May, 1988.
- Flickinger, D., Pollard, C. and Wasow, T. (1985) Structure-Sharing in Lexical Representation In *Proceedings of the 23rd Annual Meeting of the Association for Computational Linguistics*, University of Chicago Chicago, Illinois, July, 1985, pp262-267.
- Flickinger, D. (1987) *Lexical Rules in the Hierarchical Lexicon*, PhD Thesis, Stanford University.
- Gazdar, G. J. M. (1985) "Finite State Morphology

- Linguistics 23.4. pp597-607
- Gazdar, G. J. M. and Evans, R. (1989) "Inference in DATR" in *Proceedings of the Fourth Conference of the European Chapter of the Association For Computational Linguistics*, UMIST, April 1989, Morristown, NJ: ACL.
- Görz, G. and Paulus, D.(1988) "A finite state approach to German verb morphology" In *Proceedings of the 12th International Conference on Computational Linguistics and the 24th Annual Meeting of the Association for Computational Linguistics*, Budapest, Hungary, 22-27 August, 1988, pp212-215
- Hockett, C. F. (1954) Two Models of Grammatical Description. *Word*, 10, pp210-311. Also in Joos, M. (ed.), *Readings in Linguistics I*, University of Chicago Press, 1957, pp386-399.
- Hoeksma, J. and Janda, R.D. (1988) Implications of Process-Morphology for Categorical Grammar, in R. Oehrle, E. Bach and D. Wheeler (eds.) *Categorical Grammars and Natural Language Structures*, 199-247 *Studies in Linguistics and Philosophy*, D. Reidel, Dordrecht.
- Karttunen, L. (1983) "KIMMO: A general morphological processor". *Texas Linguistic Forum*, 22, 165-186.
- Karttunen, L. and Wittenburg, K. (1983), " A two-level morphological analysis of English", *Texas Linguistic Forum*, 22, 217-228
- Kiparsky, P. (1973) "'Elsewhere" in Phonology". In Anderson, S. R. and Kiparsky, P. (eds.) *A Festschrift for Morris Halle*. Holt, Rinehart and Winston
- Koskenniemi, K. (1983) "Two-level morphology: A general computational model for word-form recognition and production". Publication 11, Department of General Linguistics, University of Helsinki, Helsinki, 1983.
- McCarthy, J. (1981) "A Prosodic Theory of Nonconcatenative Morphology" *Linguistic Inquiry* 12 373-418.
- Matthews, P. H. (1974) *Morphology*, Cambridge, Cambridge University Press.
- Pereira, F. C. N. (1987) "Grammars and Logics of Partial Information" *Proceedings of the International Conference on Logic Programming*, Melbourne, Australia
- Plotkin, G. D. (1972) "Building-in Equational Theories" in D Michie and B Meltzer (eds.) *Machine Intelligence 7*, Edinburgh University Press.
- Shieber, S., Uszkoreit, H., Pereira, F.C.N., Robinson, J.J., and Tyson, M. (1983) "The Formalism and Implementation of PATR-II" in B. Grosz and M. Stickel (eds.) *Research on Interactive Acquisition and Use of Knowledge* SRI International, Menlo Park pp39-79.
- Siekmann, J. H. (1975) String-unification, part 1. Essex University ms.
- Siekmann, J. H. (1984) Universal Unification. In Shostak, R. H. (ed.) *Proceedings of the Seventh International Conference on Automated Deduction*, Napa, California, May, 1984, pp1-42. *Lecture Notes in Computer Science*, Springer-Verlag.
- de Smedt, K. (1984) "Using Object-Oriented Knowledge-Representation Techniques in Morphology and Syntax Programming", in O'Shea, T. (ed.) *ECAI '84: Advances in Artificial Intelligence*. North Holland.
- Touretzky, D. (1986) *The Mathematics of Inheritance Systems*, Morgan Kaufmann, Los Altos
- Zaenen, A. (1988) Lexical Information in LFG – an overview. Unpublished ms.
- Zeevat, H., Klein, E., and Calder, J. (1987) "Introduction to Unification Categorical Grammar" in Haddock, N.J., Klein, E. and Morrill, G. (eds.) *Categorical Grammar, Unification Grammar, and Parsing*, Edinburgh Working Papers in Cognitive Science 1, pp195-222