Syllable-based Morphology

Lynne J. Cahill*
School of Cognitive and Computing Sciences
University of Sussex
Brighton BN1 9QH
England

Abstract

This paper presents a language for the description of morphological alternations which is based on syllable structure. The justification for such an approach is discussed with reference to examples from a variety of languages and the approach is compared to Koskenniemi's two-level account of morphonology.

Keywords: morphology, phonology, syllables.

1 Introduction

The field of computational morphology was revolutionized by the work of Kimmo Koskenniemi (1983, 1984), whose two-level model of morphonology has been used for the description of several languages, including English, French, Finnish and Japanese. It is attractive computationally, being based on finite state transducers. However, we shall argue that, although FSTs are good at mapping strings of symbols onto strings of symbols, morphological representation is tree-like rather than string-like.

Work on computational morphology can roughly be divided into Koskenniemi-type models, which give an account of phonological (or orthographical) alternations, but which pay less attention to morphological aspects, such as inheritance; and inheritance-type models, which do the opposite – ignoring phonological aspects.

The latter includes lexical representation languages which provide as output objects like:

1. (suffix_er (umlaut Buch)) [after Evans & Gazdar, 1989a, p.67]

2. ... is realized by the suffixation of /en/ [Zwicky, 1985, p.374]

3. VERB → past tense suffix: +te or +de [de Smedt, 1984, p.183]

4.
```
(>>REFERENT
   (APPEND
     (APPEND [STRING "werk"])
        OF PAST-PARTICIPLE-SUFFIX))
      OF PAST-PARTICIPLE-PREFIX))
```

[Daelemans, 1987, p.40]

Work of this kind thus opens a gap which needs to be filled – namely a means of defining such morphological functions as umlaut and suffixation.

The aim of the work described here is to provide a formal language for describing (in phonological terms) those morphological alternations that are to be found in natural languages.

Section 2 will discuss the approach to morphology being advocated here, in particular contrasting it with Koskenniemi's approach, and explaining why concepts from non-linear phonologies may be useful to morphology. Section 3 will present the language itself, and some examples of how it may be applied to natural language alternations will be discussed in Section 4.

2 The Approach

Koskenniemi's model divides the morphological processor into two elements – the FST which handles (mor)phonological alternations by matching lexical and surface forms, and a system of mini-lexicons, which handle the concatenation of morphemes to make words. While this system has been shown to work with a number of languages, and while its computational simplicity makes it very attractive, such a model forces one to make a rather radical distinction between, say, suffixation and a modification function such as umlaut. Furthermore, the model is concerned with strings of segments – that is, it does not allow one to make any reference to entities above the level of the segment (syllables, feet, etc.), except by the use of rather awkward boundary diacritics. This is in contrast with much recent linguistic work on phonology, which has rediscovered suprasegmental concepts, such as syllables, metrical feet and tone

*Supported by a grant from IBM (UK) Scientific Centre, Winchester.

groups. (see e.g. Liberman & Prince, 1976, Durand,1986.)

That there is some interaction between the processes of phonology and those of morphology can be readily seen by looking at an example from Matthews (1974). The Latin verb alternations 'scribo' (I write) and 'scripsi' (I wrote) apparently involve both. The addition of the '-s-' in the past tense form is purely morphological, while the alteration of the /b/ to a /p/, while it could again be morphological, could also be accounted for by natural phonological processes which operate in other morphological environments, as a case of voicing assimilation, with the voicing feature of the /b/ assimilating to that of the /s/ to give /p/.

With this in mind, it is no surprise that the types of functions which can occur in morphology are generally similar to phonological processes and often require reference to at least a subset of those concepts required for the description of phonological phenomena. (Indeed, many morphological functions are phonological functions which have become "fossilized" in the morphology after the disappearance of the phonological conditioning context.) Koskenniemi's model was strictly segmental and used only monadic phones and phonemes, rather than feature bundles. The present work proposes that tree structures to the level of the syllable are required for morphological description, as well as feature bundles. The motivation comes from examples such as the English stressed syllable shift as in pairs like /kon'vikt/–/'konvikt/, and pairs distinguished by a single feature, such as the voicing feature in /reli:f/–/reli:v/.

The present work aims to provide a language for defining morphological functions using tree-structures, feature bundles and concepts from non-linear phonologies. It assumes the existence of a lexical representation language like those in Section 1 above, although the exact nature of that language has no bearing on the language presented here.

3 The Language

MOLUSC[1], the formal language presented here, is a declarative language based on the concept of the syllable and hence it embodies the claim that all the functions used in morphology can be defined in terms of syllables and subsyllabic constituents. A syllable structure as in Figure 1 is assumed, where the onset and coda consist of consonants and the peak consists of vowels. (The onset and coda may optionally be empty.)

The existence of a rhyme constituent is not entirely uncontroversial (see Clements and Keyser, 1983),

[1] Morphological Operations Language Using Syllable-based Constructs

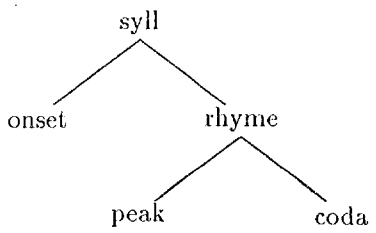

Figure 1: Syllable structure

but is nonetheless widely accepted by linguists and its role in morphological description can be seen by looking at the English verb alternations, "bring" – "brought", "think" – "thought" etc., in which the rhyme constituent in the past tense form is always /ɔɔ/ (assuming that the /t/ is a past tense suffix).

A stem[2] consists of a sequence of syllables, without any further structure between the level of syllable and stem. A disyllabic stem, such as /begin/ would therefore have the structure in Figure 2 (over).

It should be noted that this analysis is distinct from most non-linear phonologies, which postulate further levels of structure between the syllable and the stem, such as metrical feet. We maintain that these further levels of analysis are not necessary for morphological description, although this is not to deny their role in phonological description.

With this kind of structure, we still need some way of referring to particular syllables within the stem, or segments within the peak, onset and coda. We achieve this by means of a simple numbering convention, where +N refers to the Nth element from the left, and –N refers to the Nth element from the right. In addition, +0 refers to the pre-initial position with respect to a sequence of nodes, and –0 refers to the post-final position. This latter convention is intended primarily for pre- and suffixation.

MOLUSC contains one basic operation, substitution. Conceptually, this means that affixation functions are regarded as substitutions of null elements with non-null elements, and deletion functions ("subtraction" in Matthews, 1974) vice versa. A substitution is expressed in the following form:

$$[\text{LHS} \Rightarrow \text{RHS}]$$

where LHS is an expression which specifies the node of a tree which is to be replaced and RHS is an expression which specifies the subtree which is to replace it. The above function template consists of a single rule (LHS $\Rightarrow$ RHS), each rule defining a substitution. All alternations are defined in terms of

[2] We use the word stem to refer to any word-form; this is not intended to carry any implications about the nature or role of the object in question, but is merely a relatively neutral word for referring to objects above the level of the syllable.

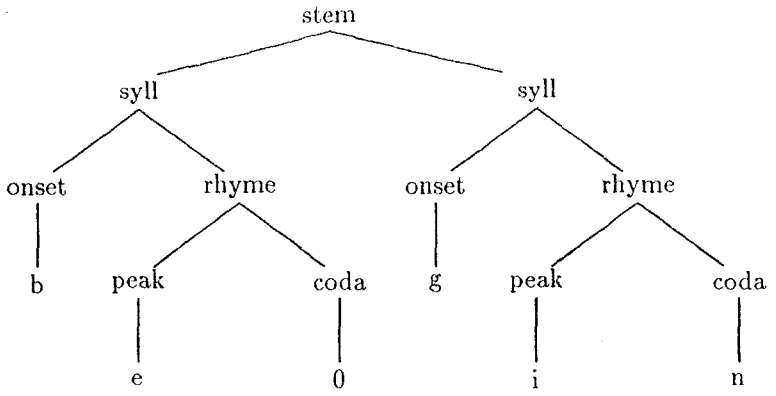

Figure 2: /begin/

functions on trees, so that a substitution involves the replacement of a single node in the tree together with all its daughters, with another node and all its daughters.

The LHS of the function is an expression consisting of a structurally unique mother category, a numerical argument indicating the syllable, and a numerical argument indicating the segment (the last two being optional with certain restrictions). This gives objects such as the following:

(stem, +1)	–	the first syllable.
(rhyme, –1)	–	the rhyme of the final syllable.
(coda, +1, –1)	–	the final segment of the coda of the first syllable.

The LHS of the function can also be qualified with additional phonological information, such as:

(coda,+1,–1)/d/
(coda,+1,–1)[+,voice]

which say respectively that the final segment of the coda of the final syllable must be /d/ or must have the feature [+, voice] for the function to have any effect. (In the event of any of the conditions failing, the identity functions applies, so that a function always applies although it may not have any effect.)

The RHS may be a description of a complete phonological object (i.e. a tree) or a feature set. Since all node names are assumed to be abbreviations for feature sets, a substitution of a feature set involves simply the substitution of some subset of features in the set.

The RHS may also consist of variables, indicated by upper case letters, which are bound to parts of the input. This is needed for reduplication functions, which require the affixation of some element (usually a syllable) which is the same as some part of the stem.

The tree-structured phonological representations are expressed with punctuation marks — a period separates the three terminal groups, onset, peak and coda, a semicolon separates syllables and commas separate the segments within a terminal group. Thus, the representation for /begin/ would be /b.e.;g.i.n/. All phonological objects, whether as input to the function or as a subtree to be substituted in within a function must have their structure specified in this way.

In addition to the qualifications which the LHS may have, any number of conditions may be placed on the input tree. A function which has conditions takes the form,

[LHS ⇒ RHS : C]

where the : behaves like the context slash / standardly used in phonology and C is any number of conditions. Each condition takes the same form as the LHS with qualifications, and may be combined as conjunctions (linked with commas) or disjunctions (linked with commas within curly braces {}). Thus, we may define a function which adds a suffix /.u./ only if the final syllable has the feature [+,light] (the plural suffixation to nouns in Old English),

[(stem, –0) ⇒ /.u./ : (stem, –1)[+,light]]

In many cases, more than one function needs to be applied, and the ways in which the functions are combined can be crucial. There are three possible situations:

- where one function must apply before another
- where two functions must apply simultaneously
- where two functions do not interact and so may apply in either order or simultaneously and yield the same result.

The first case is handled with composite application of functions. By composite application we mean

the application of functions combined with function composition, denoted by o, as standardly used. (See example of Latin reduplication below.)

For the second case we use conjoint application, indicated by &, which may apply to whole functions, or to rules within functions. Thus, we may have two alternations which both require the same conditions. For example, the Austronesian language Rotuman exhibits morphological metathesis, with alternations like /tiko/ – /tiok/, /fupa/ – /fuap/. This can be defined as a swapping of the onset and coda of the final syllable with the use of variables, but the variables must both be instantiated before either of the functions apply. We can define this using conjoint application of rules (where a rule is LHS ⇒ RHS) within a single function, e.g.

$$\text{meta} = [(\text{onset},-1) \Rightarrow /C/ \;\&\; (\text{coda},-1) \Rightarrow /O/:(\text{coda},-1)/C/,(\text{onset},-1)/O/]$$

where O and C are variables which are bound to parts of the input, in this case, the onset and coda of the final syllable respectively.

There are two types of situation which fit the third case,

- where there is an exclusive disjunction, only one of which can ever apply
- where the functions affect different parts of the structure, as in the case of Semitic verbs, where two peak change functions apply to the stem – one to the initial syllable and the other to the final syllable.

The latter is treated as function composition since there can be no question of interaction in these cases as they affect different parts of the stem. The former is treated as conjoint application of functions, since the use of function composition would seem a misuse in a situation where only one of the functions could ever apply.

4 Some Examples

There are a wide variety of different morphological functions which occur in language. This section will examine three in detail. The first of these is simple suffixation. Suffixation is the main function used in many European languages, and examples are numerous. We shall take the German example /ze:/–/ze:en/, "See" – "Seen", ('lake'–'lakes'). The basic function for describing suffixation is:

$$[(\text{stem}, -0) \Rightarrow /S/]$$

where S is the suffix. This says, intuitively, "the space after the stem is replaced with the suffix, S".

The function required to define the alternation /ze:/–/ze:en/, then is:

$$\text{suffix\_en} = [(\text{stem}, -0) \Rightarrow /.e.n/]$$

The second function we shall look at is the German umlaut. The German umlaut is a classic example of a phonological process which has become fossilized in the morphology. Once occurring as part of a vowel harmony process, it applied to stems, fronting a back vowel when a suffix with a front vowel was added, to preserve vowel harmony. In modern German, vowel harmony is no longer a productive phonological process, but the umlaut remains, largely without any suffix, as a morphological marker for plurality in some classes of nouns.

The description of the umlaut as the change from a vowel with the feature [+,back] to the corresponding vowel with the feature [–,back], is perhaps slightly misleading. Although with most vowels this is the case – /i/ – /y/, /o/ – /œ/, etc. – with the diphthong /au/, the "umlauted" equivalent is /oy/, in which the second vowel is fronted, but the first is raised. This raising can be seen as a phonological assimilation process, leaving us to define the umlaut as the fronting of a vowel if it is the only vowel, the fronting of both vowels if the peak consists of a doubled (long) vowel, and the fronting of the second vowel only in a peak which is a diphthong. This can be expressed by means of the following functions:

$$\text{umlaut1}' = [(\text{peak},?,—1)[+,\text{back}] \Rightarrow [-,\text{back}]\,]$$
$$\text{umlaut2}' = [(\text{peak},?,-2)[+,\text{back}] \Rightarrow [-,\text{back}] :(\text{peak},?,-1)/V/,(\text{peak},?,-2)/V/]$$

the second of which says that the first vowel in a peak with two vowels is fronted only if it has the same value as the second vowel.

The other problem with the umlaut is the syllable to which it applies. MOLUSC requires that a function be defined in terms of the syllable to which it applies, and this is one of the things which makes it a language for morphology rather than phonology. Phonological phenomena occur whatever the syllable, provided only that the context restrictions are satisfied. Morphological functions, we argue, although in other ways very similar to phonological ones, apply in a more restricted way to particular parts of stems, which we show by means of examples can be defined in terms of the representations described above.

Most German noun stems are monosyllabic, but those di-syllabic stems there are seem to divide fairly evenly between those to which the umlaut applies to the first syllable and those to which it applies to the second syllable, as Table 1 shows. However, if we look more closely at this table, we can see that all those nouns which undergo umlaut on the first syllable, have the unstressed, neutral-vowelled /ən/, /ə/ or /əl/ as their second syllables. This might lead us to propose an analysis which requires reference to

Nouns which take the umlaut on the first syllable		Nouns which take the umlaut on the second syllable	
Apfel	/'apfəl/	Ausflucht	/'ausfluxt/
Boden	/'boodən/	Auskunft	/'auskunft/
Bruder	/'bruudə/	Gebrauch	/gə'braux/
Garten	/'gaatən/	Gewand	/gə'vant/
Hammer	/'hamə/	Irrtum	/'iətum/
Laden	/'laadən/	Reichtum	/'raixtum/
Ofen	/'ofən/	Vormund	/'fɔmunt/
Sattel	/'zatəl/	Vorhang	/'fɔhaŋ/

Table 1: Di-syllabic German nouns

metrical feet, in order to specify that the stressed syllable takes the umlaut, but if we look again, we can see that in words such as /iətum/ ("Irrtum", 'error') and /raixtum/ ("Reichtum", 'wealth'), the unstressed, but non-neutral-vowelled second syllable is umlauted. Thus, it would seem that the neutralness of the vowel is the deciding factor, not the stress. The functions required for the umlaut in German are thus,

umlaut1 = [(peak,+1,-1)[+,back] ⇒ [-,back]
:(peak,-1,+1)/ə/]
umlaut2 = [(peak,+1,-2)[+,back] ⇒ [-,back]
:(peak,+1,-1)/V/,(peak,+1,-2)/V/,
(peak,-1,+1)/ə/]
umlaut3 = [(peak,-1,-1)[+,back] ⇒ [-,back]]
umlaut4 = [(peak,-1,-2)[+,back] ⇒ [-,back]
:(peak,-1,-1)/V/,(peak,-1,-2)/V/]

Note that there is no need to specify the context in which the second syllable undergoes the umlaut, as any stem with /ə/ in the second peak will be unchanged by the second two functions anyway, as /ə/ does not have the feature [+,back].

Since there is no interaction between these functions and only one is ever going to have any effect, we combine these with &, indicating conjoint application, thus,

umlaut = umlaut1 & umlaut2 & umlaut3
& umlaut4

Finally, let us look at a more complex function – the partial reduplication found in Latin. This involves alternations such as /fal/ – /fefeli/, /kur/ – /kukuri/. For this we need two functions. The first reduplicates the whole of the first syllable, and the second deletes the initial coda (that is, the coda of the reduplicative affix). This is necessary because we cannot reduplicate the onset and peak of the first syllable, as this is not a constituent in the structures we use. However, this analysis is actually quite attractive from a linguistic point of view, as it leaves us with two very natural functions – (reduplicative) prefixation and consonant cluster reduction, a common phonological process. The two functions are:

redup' = [(stem, +0) ⇒ /P/
:(stem, +1)/P/]
coda_0 = [(coda, +1) ⇒ /0/]

The first of these reads something like "/P/ is prefixed where /P/ is the same as the first syllable of the input stem", and uses variable binding. The whole reduplication can then be defined thus:

redup = [(coda, -1) ⇒ 0] o
[(stem, +0) ⇒ /P/
:(stem, +1)/P/]

5 Concluding Remarks

The approach to computational morphology advocated here allows one to enjoy all the benefits of powerful and economical inheritance mechanisms provided by lexical representation languages like those mentioned in 1 above, but still provide phonologically (semi-)realized forms as output. Using syllables as the basic unit of description for the realization language enables succinct and linguistically attractive definitions of a wide variety of morphological alternations from simple affixation to partial reduplication and morphological metathesis.

An interpreter for the language has been implemented in Prolog, and used to test a wide variety of morphological functions from a range of languages as diverse as English, Arabic and Nakanai. Cahill (1989) presents the language in full, with a formal syntax and semantics of the language, a description of the interpreter and analyses of substantial fragments of English and Arabic.

References

1. Cahill,L.J. – **Syllable-based Morphology for Natural Language Processing**, DPhil dissertation, University of Sussex, 1989.

2. Clements,G.N. and S.J.Keyser – **CV Phonology: A Generative Theory of the Syllable**, MIT Press, 1983.

3. Daelemans,W. – **An Object-Oriented Computer Model of Morphonological Aspects of Dutch**, Doctoral Dissertation, University of Leuven, 1987.

4. Durand,J.(ed.) – **Dependency and Nonlinear Phonology**, Croom Helm, 1986.

5. Evans,R. and G.Gazdar – "Inference in Datr", in *EACL-89*, Manchester, England, 1989, pp.66-71.

6. Evans,R. and G.Gazdar – "The Semantics of Datr", in *AISB-89*, Sussex, England, 1989, pp.79-88.

7. Koskennieni,K. – **A Two-level Morphological Processor**, PhD dissertation, University of Helsinki, 1983.

8. Koskenniemi,K. – "A General Computational Model for Word-form Recognition and Production", in *COLING '84*, 1984, pp.178-181.

9. Liberman,M. and A.Prince – "On Stress and Linguistic Rythm" in *Linguistic Inquiry 8*, 1977, pp.249-336.

10. Matthews,P. – **Morphology**, CUP, 1974.

11. de Smedt,K. – "Using Object-Oriented Knowledge-Representation Techniques in Morphology and Syntax Programming", in *ECAI-84*, 1984, pp.181-4.

12. Zwicky,A. – "How to Describe Inflection", in *Berkeley Linguistic Society 11*, 1985, pp.372-386.