Morphonology in the Lexicon

Lynne J Cahill*
School of Cognitive and Computing Sciences
University of Sussex, Brighton BN1 9QH, England
Email: lynneca@cogs.susx.ac.uk

Abstract

In this paper we present a means of defining morphonological phenomena in an inheritance based lexicon. We make use of the theory behind the formal language MOLUSC, in which morphological alternations were defined as mappings between sequences of tree-structured syllables. We discuss how the alternations can be defined in the inheritance-based lexical representation language DATR, and how the phonological aspects can be built upon to bring it closer to an integrated lexicon with representations which can be used by both the morphology and phonology of a language.

1 Introduction

The use of inheritance mechanisms in computational linguistics has become wide-ranging, with applications in semantics, syntax, morphology and phonology. In this paper, we shall examine the applicability of such mechanisms to phonological aspects of morphology.

The inheritance-based lexical representation language, DATR, has become widely used for various aspects of linguistic description, and previous treatments of both morphological and phonological phenomena in DATR have shown its applicability to this area, both for its handling of inheritance by default, and for its ability to define hierarchical structures. For example, [Gibbon, 1990] describes how Kikuyu tone displacement and Arabic non-concatenative morphology can be defined in DATR and [Reinhard, 1990] describes a hierarchical approach to German umlaut. In this paper we assume a knowledge of DATR and refer the reader to the introductions in [Cahill and Evans, 1990] and [Evans and Gazdar, 1990].

MOLUSC ([Cahill, 1990a],[Cahill, 1990b],[Cahill and Gazdar, 1990]), is a formal language for defining morphological alternations as mappings between sequences of tree-structured syllables. It is based on the theory that (many) morphological alternations are phonologically based, and can best be described as operating on hierarchical structures, such as the syllable. However, there are fundamentally linear aspects of morphological alternations, which require reference to concepts such as "initial", "final" and "penultimate".

An account of English verbal morphology was discussed in [Cahill, 1990b] which was expressed in a combined DATR/MOLUSC lexicon fragment. The morphological *distribution* was defined by the DATR while the morphological *realisation* was defined by a set of MOLUSC functions. In this paper, we discuss an account derived from this (see appendix), which expresses the distribution of alternations involved in the same underlying way, but which does not require a separate language to define them. In doing this, we can reduce the two-tiered DATR/MOLUSC approach originally used, to a single-tiered account. This has the obvious advantage of reducing the "mechanisms" needed. More importantly, however, we shall demonstrate, with discussion of how the morphonological information may be generalised to be more useful to the phonology proper, it also has the advantage of moving the account towards a fully-integrated lexicon, in which ultimately all levels of description – morphology, phonology, orthography, syntax, semantics – are combined.

*Thanks are due to Roger Evans and Dafydd Gibbon for comments on previous drafts of this paper.

In the following sections we shall consider the structures involved and how they may be defined in DATR, considering how to model both the precise structures used by MOLUSC and more generally useful phonological structures. We shall then consider how we might define the alternations. Finally, we shall discuss the advantages of this approach over previous descriptions of such phenomena as well as over the original MOLUSC language.

2 Phonological Structures

In many previous approaches to morphology, particularly in the English-dominated NLP community, it was assumed that morphology consisted fundamentally of "sticking together" morphemes, and making the necessary adjustments to allow for spelling peculiarities. [Cahill, 1990b] suggested that this was too narrow a view, even for the rather impoverished inflection displayed by English. That there are several subclasses of English verbs which inflect by means other than affixation (e.g. "bring"-"brought", "sit"-"sat") would seem to be a strong argument in itself, but looking at other languages such as the Semitic languages shows that there is an enormous body of interesting morphological phenomena which needs to be addressed. [Cahill, 1990b] showed that a view of morphological alternations as mappings between tree-structured syllables permitted a natural and succinct way of defining such alternations. It also showed that levels of structure *above* the level of the syllable, while clearly vital for phonological description, were not necessary for morphological description. Thus the approach used structures which consisted only of linear sequences of tree-structured syllables.

The question of structure above the level of the syllable is an interesting one. The use of metrical or tonal structure is clearly relevant to the phonology of a language, but it is debatable whether it has any place at all in the lexicon. While certain metrical notions such as stress *are* relevant to the lexicon (consider the noun–verb alternation "re'ject"–"'reject"), the actual metrical structure of even a polysyllabic word is dependent on the context in which it appears. Thus, it would seem reasonable to assume that the lexicon specifies the actual level of stress on each syllable of a word[1] but that the structure derived from this is extra-lexical.

In the two-tiered DATR/MOLUSC lexicon, the phonological structures were assumed to be defined fully at each lexical entry. This meant that we could not make use of the inheritance mechanisms in DATR, even though the structures lent themselves to such definition. In the present work we shall define the structures hierarchically in DATR, thus avoiding redundancy and enabling generalisations about the structures which were previously impossible.

The formal semantics of MOLUSC ([Cahill and Gazdar, 1990]) defined the structures as having sets of feature definitions at each node. Although this facility was not used in the examples provided nor in the implementation, it is an aspect of the theory which we shall build upon in the current work.

2.1 Trees

In the first instance, let us consider the simple situation, where there are onset, peak, coda and rhyme nodes which consist of (sequences of) phonemes. A simple mono-syllabic root may be defined in DATR by the following:

```
Word:   <root> == ("<onset>" "<rhyme>")
        <rhyme> == ("<peak>" "<coda>")
        <onset> == 0
        <peak> == 0
        <coda> == 0.
```

Here the root is said to consist of an onset and a rhyme (which are provided at the original query node), and a rhyme (if not explicitly defined at the query node) consists of a peak and a coda. The onset, peak and coda are by default `0`. Now we can define the root "spell" as follows:

```
Spell:  <> == Word
        <onset> == (s p)
        <peak> == e
        <coda> == 1.
```

The structure is inherited from the `Word` node, with just the values of the onset, peak and coda defined at the node `Spell`[2].

In our example theory defining a fragment of the English verb system, we only have mono- and disyllabic roots to contend with, but we need to consider how to handle a root consisting of an arbitrary number of syllables. We need to allow for a potentially infinite number of syllables in a root, but we also need each syllable in a root to maintain its own identity so as to permit both the definition of the values of the onset, peak and coda of the individual lexemes, and to allow for the definition of alternations in particular syllables. In MOLUSC this was achieved by means of a simple numbering convention where +N referred to the Nth syllable from the left and –N referred to the Nth syllable from the right.

In our DATR-only account, we achieve the linear structures by means of a path prefix "struct", and by defining the number of syllables in a root at its own lexical entry by means of a sequence of symbols – one for each syllable more than one. In the example below, we use the term "ext" (for "extension") to denote each syllable above one. Thus, a disyllabic root could be defined by the line:

[1] There is an issue of how many levels we may want to differentiate in the lexicon, but it is not one which we propose to address in the current work.

[2] The brackets around the `s p` are just for the benefit of DATR. They are removed in the final result.

```
<sylls> == ext
```

and a tri-syllabic root:

```
<sylls> == (ext ext)
```

and so on. At a higher node (in our case, "VERB") we can then define the structure of a root with the following:

```
<root> == <struct "<sylls>">
<struct ext> == (<struct> "<syll ext>")
<struct> == <syll>
<syll> == ("<onset>" "<rhyme>")
```

Let us run through how this works by looking at the example of a tri-syllabic root which does not have values for any <syll> paths defined at its node, so we can ignore the quotes on the <syll> paths. A tri-syllabic root will have the value (ext ext) for the path <sylls> at its own entry, so the first line defines the root in this case to be <struct ext ext>. The second line defines the value for the path <struct ext>, and this is the closest match for the path we want to evaluate. It is defined to be the list (<struct ext> "<syll ext ext>"), as the extra "ext"from the path we are evaluating gets added to any paths to be evaluated on the right hand side. Taking the second part of this first, <syll ext ext>, assuming it isn't explicitly defined at the word's entry, is defined as ("<onset ext ext>" "<rhyme ext ext>"). This again is because we carry over the extra elements from a left-hand side path to the right-hand side. The path <struct ext> is defined explicitly as (<struct> "<syll ext>"), and <struct> is defined as being the same as <syll>. The derivation can be viewed as follows, with the numbers of the lines from which values are derived in brackets:

```
<root> == <struct ext ext> (1)
<struct ext ext> == (<struct ext>
                    "<syll ext ext>") (2)
<struct ext> == (<struct> "<syll ext>") (2)
<struct> == <syll> (3)
<syll> == ("<onset>" "<rhyme>") (4)
-> <struct ext> == (("<onset>" "<rhyme>")
                    "<syll ext>")
<syll ext> == ("<onset ext>"
               "<rhyme ext>") (4)
-> <struct ext> == (("<onset>" "<rhyme>")
                    ("<onset ext>"
                     "<rhyme ext>"))
-> <struct ext ext> == (("<onset>"
                         "<rhyme>")
                        ("<onset ext>"
                         "<rhyme ext>")
                        "<syll ext ext>")
<syll ext ext> == ("<onset ext ext>"
                   "<rhyme ext ext>") (4)
-> <struct ext ext> == (("<onset>"
                         "<rhyme>")
                        ("<onset ext>"
                         "<rhyme ext>")
                       ("<onset ext>"
                        "<rhyme ext>"))
-> <root> == (("<onset>" "<rhyme>")
              ("<onset ext>" "<rhyme ext>")
              ("<onset ext ext>"
               "<rhyme ext ext>"))
```

The root which results is therefore,

```
("<onset>" "<rhyme>"
 "<onset ext>" "<rhyme ext>"
 "<onset ext ext>" "<rhyme ext ext>")
```

so that we can refer to the initial syllable and its constituents with paths without any "ext"s, the second syllable and its constituents with paths with one "ext" suffixed and so on.

The idea of having to define the number of syllables in a root at each lexical entry may seem a little undesirable, but we will need to define each syllable separately at the entry anyway, so the explicit information of how many syllables there are is a very small cost. In addition, since in our example fragment below most roots are monosyllabic anyway, this will not have to be defined for each entry. We can have a default value for <sylls> at the VERB node of "()". Although this is a language specific advantage, it is expected that it would not often be necessary to define polysyllabic *roots* for any language, since very long words will usually be made up of either compounded roots (as happens frequently in German) or a single root plus several affixes (as happens in agglutinative languages such as Turkish).

The structures as defined above allow us to refer to an individual syllable provided we know its position from the *left*. But if we want to refer to the *last* syllable in a root, for example, we need to know how many syllables are in each root, thus preventing us from making generalisations over classes of verbs which do not all have the same number of syllables. This is clearly undesirable, but it can be avoided. In the example of English verbs, it is a feature of the verb roots that while most roots are monosyllabic, those which are not only require reference to their *final* syllable, never their *initial* syllable. We therefore want to reverse the structure definition above, which can be very simply achieved by reversing the order of the paths <struct> and "<syll ext>" in the list on the right-hand side of the second line[3]. What this means is that we must define for each language or language fragment (it may be different for nouns and verbs, for example) whether any alternations take place at the right- or left-hand end of the root. It is possible to refer to a syllable any number in from either side, not just the initial or final

[3]In our example fragment, we have replaced the term "ext" with "pref" to reflect the fact that it is prefixing which extends the structure. The actual term used is, of course, irrelevant.

and any form or set of forms which show a deviation from the norm can be accommodated in DATR simply by means of overriding structure definitions at a lower node in the hierarchy. However, the definition of structure at the higher node(s) make for a generalisation about the set of forms covered by that node. This is in marked contrast to MOLUSC, which permitted equally easy reference to either end, and even permitted mixing within a single alternation definition. MOLUSC was much too powerful in this respect, and permitted the definition of alternations which do not occur in any language, so this is clearly a desirable restriction.

2.2 Segments within onset, peak and coda

As well as accessing syllables within a sequence, MOLUSC permitted the accessing of segments within the onset, peak and coda in a similar way. Although we do not want to go into detail here, as we do not propose to ultimately use discrete segments, we can do the same in the DATR framework outlined above, by means of a similar mechanism to that used for syllables. Again, we need to decide whether we want to extend leftwards or rightwards, and this again gives us a highly desirable restriction, which in this case we can use to restrict onsets to extend rightwards and codas to extend leftwards. Thus, we may refer to initial, second etc. segments within the onset and final, penultimate etc. segments within the coda but not *vice versa*. Of course, DATR itself does not force such restrictions, but the framework we have defined forces the lexicon writer to decide on how to apply the restrictions.

2.3 Phonological features

As mentioned above, we have used segments in the examples given so far for clarity of explanation. In MOLUSC and in the current work we intend the real unit of description to be the phonological *feature* or *event* rather than the segment. Much recent work, both computational and theoretical, has shown that the use of such units permits a more accurate, and above all, a declarative description of phonological phenomena such as ellision, epenthesis and assimilation (e.g. [Coleman, 1992], [Bird and Klein, 1990]). As mentioned above, [Cahill and Gazdar, 1990] defined a formal semantics for the MOLUSC language which permitted the definition of phonological features at any level in the structure, although the implementation and examples did not make use of feature definitions except at terminal nodes. Thus, the segment labels below the onset, peak and coda nodes were deemed to be abbreviations for a set of features, any one or more of which could be altered by a morphological alternation. It is possible to talk about inheritance of phonological features up or down the tree. For example, a `[+ voice]` feature at a rhyme node may be considered to be inherited by the peak and coda nodes below it, so that any segments within either of those two will contain the feature `[+ voice]`. Alternatively, the value of a coda node (null or non-null) may be inherited by the rhyme and syllable nodes above it in order to specify stress values which may be affected by whether a syllable is "open" or "closed".

In the account we are proposing here, we only require the latter type of inheritance, where the higher nodes inherit features from the lower nodes. This is because we are advocating an approach to phonology like that proposed by [Bird and Klein, 1990], [Coleman, 1992]. In both of these approaches, phonological features consist of a feature (or "event") name, a value for that feature and an argument which defines how it relates temporally to the other features in the word[4]. Thus, for example, in the word "bat" there may be features such as `[+ voice]`, `[+ labial]`, `[+ alveolar]`, `[+ consonant]` and `[+ vowel]` amongst others[5]. The voice feature would have a temporal argument which expressed the fact that it lasts for the entire word, the labial feature would be defined as lasting for some time from the beginning of the word until the onset of the vowel feature and the alveolar feature would be defined as commencing at the end of the vowel feature and ending at the end of the word. Of course, this is very approximate, but it is intended only to give the flavour of the treatment.

In an account of this nature, it is not necessary for the features at higher nodes to be "trickled" down to lower nodes, since the temporal arguments define how they relate. However, that is not to say that the structure is unimportant. Both of the theories from which we are borrowing here make use of the notion of a tree-structured syllable, and phonological restrictions are defined as holding within such structures. In particular, from the point of view of the current account, reference to parts of the structure is necessary for the definition of morphological alternations (see below). In our definition of the structure given above, the result is a simple list (of segments in the example we gave). If we use features instead of segments, the same result is achieved, but instead of a list of segments, considered to be temporally ordered, we end up with a list of features, not temporally ordered, but with explicit temporal arguments defining their relationship to each other.

2.3.1 The features and their arguments

Although we will not be using segments, we maintain the notion of segmental units in the temporal

[4] [Bird and Klein, 1990] does not use features in this form exactly, but the notion of temporal relations between the events is vital to their account

[5] It is important to note here that in this and all subsequent examples of actual phonological features, no claim is being made as to the accuracy of the actual features used. They are meant purely to demonstrate the applicability of the framework to (morpho)phonological description and not to demonstrate a full phonological theory.

arguments in our examples below. We argue that segments, although possibly unnecessary in strict phonological terms, do seem to have a role at some level. The very fact that our writing system makes use of segment-type units appears to be an argument in favour of maintaining their existence at some level, and in morphological terms it is clear that many alternations seem to require reference to such units. For example, the English alternation "bend"-"bent" can be defined as an alternation in the voicing feature of only the final segment of the coda. We are therefore assuming timing boundaries at what would have been segment boundaries. Thus the word "spell" is assumed to have four "timing sections", one for each of the conventional segments. The stem "spell" can thus be redefined as follows:

```
Spell:  <> == VERB
        <rhyme_feats> == ([ + voice 2-4 ])
        <onset> == ([ - voice 0-2 ]
                    [ + sibilant 0-1 ]
                    [ + alveolar 0-1 ]
                    [ + stop 1-2 ]
                    [ + labial 1-2 ])
        <peak> ==  ([ - round 2-3 ]
                    [ - high 2-3 ]
                    [ - low 2-3 ]
                    [ + front 2-3 ])
        <coda> ==  ([ + lateral 3-4 ]).
```

The third element in each list is a (very simple) temporal argument. The sibilant feature, for example, lasts from 0 to 1, i.e. the first "segmentsworth"; the approximant feature of the onset goes from 1 to 2, i.e. the second "segmentsworth"; the voice feature of the onset covers the whole two segmentsworth of the onset. These are of course extremely simplified, both in the definition of the temporal arguments, and in the descriptions of the features themselves. But the theories from which we are borrowing have plenty to say about these aspects of phonology which is not relevant to how it might be expressed in a DATR lexicon combining phonological and morphological description. Note that in the example above, since we have temporal arguments, it is possibly not necessary to differentiate the rhyme features (just the voicing feature in the above example) from syllable features. We can have the feature `[ + voice 2-4 ]` defined at either the rhyme or syllable node. Since all rhyme features are inherited by the syllable, it will only be relevant to make the distinction if an alternation requires reference to the rhyme features specifically. However, it is more accurate to maintain the distinction, and so we shall do so.

2.3.2 Inheriting feature arguments

The description above requires that every feature for which we want to define a value in a stem must be explicitly defined. In addition, every value for every feature must be explicitly defined. There is no room for a marked/unmarked distinction, for example. In doing this, we are not making use of DATR's default inheritance mechanisms to define default values for features. What we can do to improve on this situation is to define a set of features for which a value and a timing must be given (although the value may be "undef" or some such), and provide default values for each feature at a very high node.

The set of features we have chosen are not intended to be comprehensive or even necessarily consistent, but are simply those sufficient to describe the stems and alternations involved in our example fragment. The feature set is as follows:

```
alv    = alveolar
approx = approximant
fric   = fricative
high   = high
lab    = labial
lat    = lateral
low    = low
nasal  = nasal
round  = round
sib    = sibilant
stop   = stop
vel    = velar
voice  = voice
```

The default value for all features is "-" and the default timing is `rl`, for "root length" – i.e. the whole length of the root.

The definition of the structure of a stem (i.e. the number of syllables) is as before, but the definition of a syllable needs to take into account the fact that we are now dealing with lists of features and their values and timings, rather than linear sequences of segments. Since we are going to permit the permeation of features up the tree, we want the syllable node to contain all of the features for the onset and rhyme nodes, and the rhyme node to contain all of the features for the peak and coda nodes. One consequence of this is that we cannot simply allow the definition of features shared by say the peak and coda nodes at the rhyme node, since they will not then be inherited downwards, and any alternation which is dependent on the value of a feature at the coda node will need to look at the rhyme and syllable nodes' features, taking the timings into account as well. It would undoubtedly be possible to get around this problem but for our present purposes the extra cost of defining a shared feature at both nodes which share it is not a problem.

The feature sets can be defined as follows:

```
<syll> == ([ <feats syll> ]
           [ <feats onset> ]
           [ <rhyme> ])
<rhyme> == ([ <feats rhyme> ]
            [ <feats peak> ]
            [ <feats coda> ])
```

The paths are not quoted in this because we want the actual feature set to be defined at the top node, with

just the values and timings defined at the terminal nodes. Thus, the feature set can be defined:

```
<feats> == (
  [ alv    "<val alv>"    "<time alv>"
    approx "<val approx>" "<time approx>"
    fric   "<val fric>"   "<time fric>"
    high   "<val high>"   "<time high>"
    lab    "<val lab>"    "<time lab>"
    lat    "<val lat>"    "<time lat>"
    low    "<val low>"    "<time low>"
    nasal  "<val nasal>"  "<time nasal>"
    round  "<val round>"  "<time round>"
    sib    "<val sib>"    "<time sib>"
    stop   "<val stop>"   "<time stop>"
    vel    "<val vel>"    "<time vel>"
    voice  "<val voice>"  "<time voice>" ])
```

Then to find the set of features at the peak node, for example, the word **peak** is appended to all of the (quoted) paths in the feature list, thus evaluating the **val** and **time** for each feature at that node. The paths:

```
<val> == -
<time> == rl
```

then define default values for the **val** and **time** paths. With these definitions, we can define a stem by simply providing values for all those features which have the value "+"[6] and times for these. The example stem "spell' can therefore be defined as:

```
Spell:  <> == VERB_A
        <val sib onset> == +
        <val lab onset> == +
        <val stop onset> == +
        <val front peak> == +
        <val voice peak> == +
        <val lat coda> == +
        <val voice coda> == +
        <time sib onset> ==  0-1
        <time lab onset> ==  1-2
        <time stop onset> ==  1-2
        <time front peak> ==  2-3
        <time voice peak> == 2-3
        <time voice coda> == 3-4
        <time lat coda> ==  3-4.
```

Although the timings we are using here are extremely approximate, they can provide a starting point for phonological/phonetic systems, such as YorkTalk ([Coleman, 1992]). The YorkTalk system defines phenomena such as epenthesis as adjustments in the timings of such features, so as to blur the boundaries between "segment" sections. For example, the epenthesis which occurs in words such as "mince" (/mints/) is a result of the fact that the closure aspect of the nasal /n/ is carried over to the non-nasal /s/, resulting in a /t/ sound. This type of phonological phenomenon is not something we would expect to be represented in the lexical entry for the word "mince", but having approximate, relative timings of the features gives a system like YorkTalk something with which it can work more easily than simple segmental structures. They also eliminate the need to refer to individual segments within onset, peak and coda. The stem "bend" for example has a coda which consists of an "n" and a "d" (in conventional terms). This can be expressed in our account by the features voice, alveolar, nasal and stop having the value "+" in the coda, but with the following timings:

```
<time voice coda> == 2-4
<time nasal coda> == 2-3
<time alv coda> == 2-4
<time stop coda> == 3-4
```

The voice and alveolar features carry across the whole coda, but the nasal feature is only on the first section and the stop feature is only on the second.

There would appear to be a problem here, resulting from the decision to only allow inheritance of features up the tree, in that it is possible for a feature at a particular node to be given a value at that node but a timing which only covers part of the node. For example, the stem "swell" has an onset whose voice feature has the value "-" for the first section and "+" for the second section. However, as we noted in the example of "spell" above, it is possible for the syllable node to contain features whose temporal arguments do not cover the whole syllable. Thus, the onset of "swell" would have a feature "**[- voice]**" which has the timing "**0-1**" and the syllable node would have the feature "**[+ voice]**", with the temporal argument "**1-4**".

3 Morphological Alternations

Let us momentarily return to the use of segments for clarity, and consider how to define alternations between forms. In our example case of English verbs, most of the inflections take the form of suffixation, which can be defined trivially. For example, the present participle form might be defined:

```
<pres part> == ("<root pres>" ing)
```

(with the suffix itself having its structure defined – we are not concerned with that here). What is more interesting, however, is the definition of alternations such as that in the forms "bereave"–"bereft", "cleave"–"cleft". Such verbs, although only in small groups, do exhibit consistent, phonologically determined, but not strictly *phonological*, alternations. In MOLUSC, these could be defined by means of functions such as the following:

[(peak,-1)/ii/ ⇒ /e/]
[(coda,-1,-1)/v/ ⇒ /f/]
[(coda,-1,-1)[+ voice] ⇒ [- voice]]

[6] We are using simple boolean valued features here, but this is not a restriction. Multi-valued features, such as stress (see section 2 above), can be just as easily accommodated.

There are two aspects to these alternations. On the one hand, defining the alternation between, say, a peak of /ii/ with a peak of /e/ is extremely straightforward, simply requiring path extensions to the "<peak>"[7] definitions for past and present. Thus, the following would define the alternation:

```
<peak pres> == ii
<peak past> == e
```

However, in the account of English verbs in [Cahill, 1990b], such verbs were grouped together with a large number of other verbs which did not exhibit this precise alternation, with the peak alternation being dependent on the original peak. Thus, the past tense peak is /e/ *if* the present tense peak is /ii/ and the same as the present tense peak otherwise.

3.1 Defining context-dependent alternations

We can define this type of context-dependent alternation in our framework by evaluating the present tense value for the peak and using that as an argument in a path for defining the past tense peak. The code for this is:

```
<peak past> == <peak_change "<peak pres>">
<peak_change ii> == e
<peak_change> == "<peak pres>"
```

This says that the peak of the past tense root (<peak past>) is found by evaluating the path which has the word peak_change followed by whatever the value of the present tense peak is ("<peak pres>"). If this results in the path <peak_change ii> (i.e. if the present tense peak is "ii") then the past tense peak is "e". In any other case (the path with the present peak value unspecified) the past tense peak is the same as the present tense peak ("<peak pres>").

3.2 Defining feature value alternations

The coda change function is given in MOLUSC in two different forms – one with segments and the other with features. The version with segments can, unsurprisingly, be defined in exactly the same way as the peak change above. Let us consider the alternation defined as an alternation in the value of the voicing feature. The voicing feature of the set of verbs we are talking about is altered if the final segment of the coda is either a labial fricative ("v") or an alveolar stop ("d"). There are therefore four features in whose values we are interested: lab, fric, alv and stop. We can define the value of the voicing feature of the coda in the past tense form to be dependent on the values of all four of these features:

```
<val voice coda past> ==
        <coda_change "<val fric coda pres>"
                     "<val lab coda pres>"
                     "<val stop coda pres>"
                     "<val alv coda pres>">
```

and we can define the actual value simply by means of the following two DATR sentences:

```
<coda_change + +> == -
<coda_change - - + +> == -
```

The first says that, if the values of both the fric and lab features are "+" then the voice feature has the value "–", regardless of what the values of the stop and alv features are. The second says that if the fric and lab features both have the value "–" *and* the stop and alv features both have the value "+" then the value of the voice feature is "–". Note that the asymmetry is necessary but insignificant. It is not possible to define the alternation so that it is unimportant what the values of *either* the fric and lab features or the stop and alv features are, but it should be clear that in a consistent phonology, it would not be possible to have both the fric and stop features having the value "+" and even if it were possible to have the alv and lab features with the value "+", it is highly unlikely that it would affect such an alternation. That is to say, in the examples of alternations we have looked at, such conflicts have never arisen.

Two more alternations which can interestingly be handled very neatly in this framework are the sibilant/voice and alveolar/voice dependent "s" and "d" suffixes in English. The plural noun and present tense third person singular verb suffixes in English both have three realisations: /iz/ after sibilants, /s/ after unvoiced non-sibilants and /z/ after voiced non-sibilants. Traditionally this is defined with rules such as,

S → /iz/ / [+ sib] __
S → /s/ / [– voice] __
S → /z/ / [+ voice] __

where the first rule must apply before the other two. Alternatively, the feature [– sib] must be specified in the second and third rules in order to eliminate the need for ordering. In our account, we can define this alternation declaratively and succinctly. As with the coda voicing alternation described above, we need to evaluate a path which contains values of features – in this case the sib and voice features. The present tense third person singular form is defined as:

```
<pres third sing> ==
 ("<root pres>" <ssuff "<val sib coda>"
                       "<val voice coda>">)
```

and the value of the suffix ("ssuff") is defined very simply with the following lines[8]:

```
<ssuff +> == iz
<ssuff - +> == z
<ssuff ->> == s
```

[7] This is the peak of the *final* syllable in all cases. We have already discussed above how to define roots as extending from either the right or the left, and we assume here that the roots all extend from the right.

[8] We have left the suffix forms as segments rather than expanding them out to features for simplicity.

This says that if the value of the **sib** feature is "+" then the ssuff is "iz", regardless of what the value of the **voice** feature is, and if the **sib** feature has the value "-" then the ssuff is "z" if the **voice** feature has the value "+" and "s" otherwise. We can do a similar thing for the past tense /id/-/d/-/t/ suffix with the **alv** and **voice** features. This analysis permits us to define the alternation declaratively, and hence without any need for rule ordering, but we can specify one feature value less than is necessary to avoid ordering in the traditional description.

4 Conclusions

We have presented an approach to describing morphological alternations in the lexicon which combines linear and hierarchical notions, making use of the theory behind MOLUSC. Let us now consider the advantages of this approach, both over the MOLUSC language and over previous DATR approaches to such phenomena.

MOLUSC defined all morphological alternations as mappings between linear sequences of tree-structured syllables, including affixation. This required extending the numerical labelling to include +0 and -0 to represent the prefix and suffix slots. While this was a reasonable extension to permit the definition of all morphological alternations within the same framework, it ignored the obvious difference between affixation and phonologically related alternations. It also implied (although it did not require) that all affixes were monosyllabic. While this is very often the case, it is by no means always so (e.g. English "ation", Latin "amus" etc.) and MOLUSC did not have anything to say about these. Equally, it did not permit compounding, since every morphological process had to involve a stem and an affix.

In the account we have proposed here, we can have the best of both worlds. We can use the type of definitions of alternations that MOLUSC used to handle the phonologically related phenomena, but we can leave the affixation and compounding to be treated as simple concatenation in DATR lists.

The account proposed here also has the advantage, mentioned above, that certain types of alternation and structural definition are much harder to define than in MOLUSC. MOLUSC was noticably overpowered, permitting the definition of alternations which affected both the first syllable and penultimate coda, dependent on the value of the third onset, for example, a combination unlikely in the extreme. [Cahill, 1990b] discussed some possible ways to restrict the language to have context dependencies adjacent to the alternation being defined and to only permit reference to the initial, second, final and penultimate syllables, for example. Such restrictions are not *enforced* by the account discussed above, but the kind of alternations which we would want to avoid are notably more difficult to define, which is in contrast to MOLUSC.

The present account has much in common with that in [Gibbon, 1990], which provided accounts of Kikuyu tone displacement and Arabic *binyan* morphology. The account Gibbon gave of Arabic can be directly contrasted with the general approach proposed here. Gibbon, like most others, makes use of a C V template level, with the C and V slots being filled by inheritance through a DATR lexicon. In our account, we can deal with the Arabic "template" morphology without the need for this extra layer, by using the syllabic structure. The vowels are defined simply to be the peaks of the first, second etc. syllables and the consonants are defined as the onsets and codas. An analysis along these lines using MOLUSC was given in [Cahill, 1990b], and it could be translated into the framework described above in the same way as the English fragment has been. This would amount to a description very similar to that in [Gibbon, 1990], but the resultant form, instead of being simply a sequence of segments, would be a fully specified phonological structure of the type described above. Thus, the node for each triliteral root would define the three basic consonant feature sets, with the form for each *binyan* being defined as a syllable sequence, for which the onset and coda for each syllable would inherit from the root definition. The vowel alternations would be defined exactly as the peak alternations in the English example above.

A small example DATR theory by Dafydd Gibbon in [Evans and Gazdar, 1990] (pp. 99-100) also gives a small example of phonological underspecification could be expressed in DATR. An interesting extension of the current work would be to attempt to integrate it with the definition of underspecified phonology given by Gibbon.

The framework outlined here, then, permits the same intuitive description of morphonological alternations as did MOLUSC, but with the following advantages:

- it forces the lexicon writer to restrict, or at least guide, the types of alternations occurring in any language fragment;
- it permits a more simple and intuitive treatment of concatenation;
- it moves the theory closer to an integrated lexicon – the output of the morphology is phonological representations which could be used by existing phonological theories and implementations.

Appendix: The DATR code

```
VERB:
    <> == ()
    <root> == <struct "<sylls>">
    <sylls> == ()
    <struct pref> == ("<syll pref>"
                          <struct>)
    <struct> == <syll>
```

```
    <syll> == ([ <feats syll> ]
               [ <feats onset> ]
               [ <rhyme> ])
    <rhyme> == ([ <feats rhyme> ]
                [ <feats peak> ]
                [ <feats coda> ])
    <feats> ==
    ([alv    "<val alv>"    "<time alv>"
      approx "<val approx>" "<time approx>"
      fric   "<val fric>"   "<time fric>"
      high   "<val high>"   "<time high>"
      lab    "<val lab>"    "<time lab>"
      lat    "<val lat>"    "<time lat>"
      low    "<val low>"    "<time low>"
      nasal  "<val nasal>"  "<time nasal>"
      round  "<val round>"  "<time round>"
      sib    "<val sib>"    "<time sib>"
      stop   "<val stop>"   "<time stop>"
      vel    "<val vel>"    "<time vel>"
      voice  "<val voice>"  "<time voice>"])
    <val> == -
    <time> == rl
    <pres> == "<root pres>"
    <pres part> == ("<root pres>" ing)
    <pres third sing> == ("<root pres>"
                 <ssuff "<val sib coda>"
                        "<val voice coda>">)
    <ssuff +> == iz
    <ssuff - +> == z
    <ssuff - -> == s
    <past> == ("<root past>"
               "<dsuff "<val alv coda>"
               "<val voice coda>">")
    <dsuff +> == id
    <dsuff - +> == d
    <dsuff - -> == t.

VERB_A:
    <> == VERB
    <feats peak past> == <peak_change
                         "<feats peak pres>">
    <peak_change ii> == e
    <peak_change> == "<feats peak pres>"
    <val voice coda past> ==
         <coda_change "<val fric coda pres>"
                      "<val lab coda pres>"
                      "<val stop coda pres>"
                      "<val alv coda pres>">
    <coda_change + +> == -
    <coda_change - - + +> == -
    <coda_change> == "<val voice coda pres>"
    <dsuff> == t.

Spell:  <> == VERB_A
        <val sib onset> == +
        <val lab onset> == +
        <val stop onset> == +
        <val front peak> == +
        <val voice peak> == +
        <val lat coda> == +
        <val voice coda> == +
        <time sib onset> ==  0-1
        <time lab onset> ==  1-2
        <time stop onset> ==  1-2
        <time front peak> ==  2-3
        <time voice peak> == 2-3
        <time voice coda> == 3-4
        <time lat coda> ==  3-4.

Live:   <> == VERB
        <val lat onset> == +
        <val voice onset> == +
        <val high peak> == +
        <val front peak> == +
        <val voice peak> == +
        <val voice coda> == +
        <val fric coda> == +
        <val lab coda> == +
        <time lat onset> == 0-1
        <time voice onset> == 0-1
        <time high peak> == 1-2
        <time front peak> == 1-2
        <time voice peak> == 1-2
        <time voice coda> == 2-3
        <time fric coda> == 2-3
        <time lab coda> == 2-3.

Bereave:<> == VERB_A
        <sylls> == pref
        <val voice onset pref> == +
        <val lab onset pref> == +
        <val stop onset pref> == +
        <val front peak pref> == +
        <val voice peak pref> == +
        <feats coda pref> == 0
        <val voice onset> == +
        <val approx onset> == +
        <val high peak> == +
        <val front peak> == +
        <val voice peak> == +
        <val voice coda> == +
        <val fric coda> == +
        <val lab coda> == +
        <time voice onset pref> == 0-1
        <time lab onset pref> == 0-1
        <time stop onset pref> == 0-1
        <time front peak pref> == 1-2
        <time voice peak pref> == 1-2
        <time voice onset> == 2-3
        <time approx onset> == 2-3
        <time high peak> == 3-5
        <time front peak> == 3-5
        <time voice peak> == 3-5
        <time voice coda> == 5-6
        <time fric coda> == 5-6
        <time lab coda> == 5-6.

Bend:   <> == VERB_A
```

```
<val voice onset> == +
<val lab onset> == +
<val stop onset> == +
<val front peak> == +
<val voice peak> == +
<val voice coda> == +
<val nasal coda> == +
<val alv coda> == +
<val stop coda> == +
<time voice onset> == 0-1
<time lab onset> == 0-1
<time stop onset> == 0-1
<time front peak> == 1-2
<time voice peak> == 1-2
<time voice coda> == 2-4
<time nasal coda> == 2-3
<time alv coda> == 2-4
<time stop coda> == 3-4.
```

References

[Bird and Klein, 1990] S. Bird and E. Klein. Phonological events. *Journal of Linguistics*, 26, 1990.

[Cahill and Evans, 1990] Lynne J. Cahill and Roger Evans. An application of DATR: The TIC lexicon. In *Proc. ECAI-90*, pages 120–125, 1990.

[Cahill and Gazdar, 1990] L. J. Cahill and G. J. M. Gazdar. The semantics of MOLUSC. In *ECAI-90*, pages 126–131, Stockholm, 1990.

[Cahill, 1990a] L. J. Cahill. Syllable-based morphology. In *COLING 90*, volume 3, pages 48–53, Helsinki, 1990.

[Cahill, 1990b] L. J. Cahill. Syllable-based morphology for natural language processing (DPhil Dissertation). Technical Report Cognitive Science Research Report 181, Cognitive and Computing Sciences, University of Sussex, 1990.

[Coleman, 1992] J. S. Coleman. Synthesis by rule without segments or rewrite rules. In C.Benoit and G.Bailly, editors, *Talking Machines*. Elsevier, 1992.

[Evans and Gazdar, 1990] R. Evans and G. Gazdar. The DATR papers. Cognitive science research report 139, Cognitive and Computing Sciences, University of Sussex, 1990.

[Gibbon, 1990] Dafydd Gibbon. Prosodic association by template inheritance. In Walter Daelemans and Gerald Gazdar, editors, *Proceedings of the Workshop on Inheritance in Natural Language Processing*, pages 65–81. Institute for Language Technology, Tilburg, 1990.

[Reinhard, 1990] S. Reinhard. Verarbeitungsprobleme nichtlinearer Morphologien: Umlaut-beschreibung in einem hierarchischen Lexicon. In B. Rieger and B. Schaeder, editors, *Lexicon und Lexicographie*. Olms Verlag, Hildesheim, 1990.