Lynne J. Cahill and Gerald Gazdar
University of Sussex
Email: {Lynne.Cahill,Gerald.Gazdar}@cogs.susx.ac.uk

ALLOMORPHY IN PolyLex[1]

Abstract

The PolyLex project aims to produce a hierarchical multilingual lexicon for Dutch, English and German, in which information common to more than one language is inherited from a shared component. The PolyLex work done to date has concentrated on the morphology and morphophonology of the three languages. In this paper we present the morphological framework used in PolyLex with examples of the ways in which allomorphic variation is handled.

1. Introduction

Our general approach to inflectional morphology[2] falls within the tradition that treats paradigms (inflectional classes, declensions, conjugations, etc.) as analytically central[3] rather than epiphenomenal or of secondary status[4]. The central notion is the lexeme, not the word or the morpheme. Words exist, but only as *realizations* of (morphosyntactic specifications of) lexemes – hence Stump's use of the term *realizational* to characterize this tradition. Morphemes also exist, but only as second class citizens. The appearance of a morpheme is just one among several ways that morphosyntactic information gets expressed in the realization of a lexeme as a word (cf. Wurzel 1990, 208-209). And we share Zwicky's view that "all realization rules are treated as expressing **defaults**, which are automatically overridden by more specific rules (and these in turn by still more specific rules, and so on)" (1985, 372).

As regards current work, our approach is closely related to Corbett & Fraser's Network Morphology[5] and the most recent version of Stump's Paradigm Function Morphology (forthcoming). In our approach, unlike those of Stump and Corbett *et al.*, abstract inflectional rules are typically stated in terms of phonological units, most commonly the syllable and the segment (as in Cahill 1990a, 1990b, 1993). Gibbon and his collaborators in the ILEX (Integrated Lexicon with EXceptions)

[1]This work was supported by ESRC research grant *Multilingual lexical knowledge representation,* number **R000235724**, to Gazdar & Cahill.

[2]Described in more detail in `http://www.cogs.susx.ac.uk/lab/nlp/polylex/polylex.html`, Cahill and Gazdar (1997, forthcoming).

[3]As in the work of Matthews (1972), van Marle (1985), Zwicky (1985, 1990), Carstairs (1987), and Stump (e.g., 1992; 1993a; 1993b; 1993c; 1995).

[4]Thus, for example, inflectional class is a secondary notion for Wurzel (1990, 204): for him it is the citation form that determines the inflectional class, not the converse.

[5]See Brown *et al.* (1996), Brown & Hippisley (1994), and Fraser & Corbett (1995; in press) for work in this framework.

project at Bielefeld[6] have pioneered the use of default inheritance hierarchies for the representation of lexical phonology and morphophonology. Our work is thus also indebted to theirs.

2. The DATR language

The PolyLex lexicons are implemented in the lexical knowledge representation language DATR (Evans & Gazdar 1996)[7]. DATR is a rather spartan nonmonotonic language for defining inheritance networks with path-value equations. The development of DATR was guided by a number of concerns which we summarise here. The objective was to design a language which (i) has an explicit theory of inference, (ii) has an explicit declarative semantics, (iii) can be readily and efficiently implemented, (iv) has the necessary expressive power to encode the lexical information presupposed by work in the unification grammar tradition, and (v) can express all the evident generalizations and subgeneralizations about such entries. In keeping with its intendedly minimalist character, it lacks many of the constructs embodied either in general purpose AI knowledge representation languages or in contemporary grammar formalisms. The language is nonetheless sufficiently expressive to represent concisely the structure of lexical information at a variety of domains of language description.

It should be stressed that DATR itself is no more than a very general **language** for lexical description and therefore does not commit or restrict the linguist using it to any particular linguistic framework, theory or formalism, nor is it restricted in the class of natural languages that it can be used to describe. Clearly, it is well suited to lexical frameworks that embrace or are consistent with inheritance and non-monotonicity through networks of nodes, but these are not requirements. DATR can be (and has been) used to implement differing theoretical approaches (including ILEX, HPSG, Word Grammar, LTAG, Finite State Morphology, Network Morphology, Paradigm Function Morphology), and is perhaps best thought of as a programming language which can be used to implement and test linguistic theories. Indeed, it would not be entirely misleading to think of DATR as a kind of assembly language for constructing (or reconstructing) higher level theories of lexical representation. Unlike most other formal languages proposed for lexical knowledge representation, DATR is also not restricted in the domains of linguistic description to which it can sensibly be applied. It is designed to be equally applicable at phonological, orthographic, morphological, syntactic and semantic domains of description. But it is not intended to replace existing approaches to those domains. DATR cannot be (sensibly) used without a prior decision as to the theoretical frameworks in which the description is to be conducted; there is thus no 'default' framework for describing, say, morphological facts in DATR.

In DATR, information is organised as a network of **nodes**, where a node is essentially just a collection of related information. In the context of lexical description, a node might correspond to a phoneme, a syllable, a morpheme, a word, a lexeme, etc., or a class of such items. For example, for German, we might have

[6] See Bleiching (1992; 1994), Bleiching et al. (1996), Gibbon (1990; 1992), Gibbon & Bleiching (1991), Reinhard (1990) and Reinhard & Gibbon (1991) for examples of this work.

[7] See also `http://www.cogs.sussex.ac.uk/lab/nlp/datr/datr.html`

a node describing an abstract `Word`, a node for the class of nouns, a node for the subclass of nouns that mark plurals with *-s*, a node for the particular noun lexeme `Klub` ('club') and still more for the individual words that are instances of this lexeme *Klub*, *Klub-s*. Each node has associated with it a set of equations that define partial functions from **paths** to **values** where paths and values are both sequences of **atoms** (which are primitive objects). Atoms in paths are sometimes referred to as **attributes**. The syntax and terminology of DATR, like its name and its minimalist philosophy, owes more than a little to that of the unification grammar language PATR (Shieber 1986).

3. Phonology

Our interest in phonology in the PolyLex project is restricted to those aspects of phonological structure that are relevant to the description of inflection in the languages considered. Those aspects include syllable structure but do not include any structure above the level of the syllable, such as metrical structure.

We also restrict ourselves to a segmental representation of the phonology. Our phonological segment inventory is taken from CELEX (Baayen et al., 1995) and uses the SAMPA machine-readable phonetic alphabet (Wells, 1987). As one of us has shown in earlier work (Cahill 1993), the step from representing structures with segments to representing the same structures with full feature sets at each point in the tree is relatively simple. We have not taken that step here because it would not add anything to most of the present analysis but it would make our DATR code much harder to read. However, a featural encoding would permit a more elegant treatment of phonological alternations such as final consonant devoicing and morphophonological alternations such as vowel lengthening and umlaut.

As in Cahill (1990b) and Bleiching (1992), we define syllabic structures by means of simple context-free phrase structure rules:

syllable → *onset rhyme*
rhyme → *peak coda*
coda → *body tail*
disyllable → *syllable syllable*
trisyllable → *syllable syllable syllable*

A syllable consists of an onset and a rhyme; a rhyme consists of a peak and a coda; and a coda consists of a body and a tail[8]. A disyllable consists of two syllables, and a trisyllable of three. We can express these in DATR as follows[9]:

```
Syllable:
    <phn $yll form>  == "<phn $yll onset>" "<phn $yll rhyme>"
```

[8]The tail of a coda is its final segment and the body consists of any remaining consonants in the coda. This simplifies reference to final consonants of roots.

[9]We have simplified and/or modified the DATR code from the actual PolyLex lexicons whenever this has looked likely to enhance the readability of the present paper and assist us in making the points at issue. We have also spared our readers the many pedantic footnotes that would be required to document every case of such code editing.

```
    <phn $yll rhyme> == "<phn $yll peak>"  "<phn $yll coda>"
    <phn $yll coda>  == "<phn $yll body>"  "<phn $yll tail>"
    <phn root>  == <phn syl1>
    <> == Null.

Disyllable:
    <> == Syllable
    <phn root>  == <phn syl2> <phn syl1>.

Trisyllable:
    <> == Syllable
    <phn root>  == <phn syl3> <phn syl2> <phn syl1>.
```

This rule schema makes crucial use of a variable **$yll** that ranges over attributes (**syl1, syl2, ..**) that denote syllable positions. Note also that the maximally unspecified path (<>) at the **Syllable** node is defined by reference to **Null** which always returns the empty sequence as its value. An **<onset>**, **<peak>** or **<coda>** which is left undefined at lower levels of the hierarchy will, as a consequence, end up as null.

The definitions of di- and trisyllables number the syllables from the right. This is a language-specific aspect of our analysis and reflects the fact that Dutch, English and German morphology all primarily involve suffixation. Reference to final syllables is thus more frequent than reference to the initial syllables and it is technically convenient to have a constant identifier (**syl1** here) for final syllables.

Given this set of axioms for syllabic structure, we can now use them to help define particular concrete (poly)syllables. Here, for example, is a possible definition for the monosyllabic *-es* suffix, realized phonologically as **/@s/**.

```
Suffix_es:
    <> == Syllable
    <phn syl1 peak> == @
    <phn syl1 coda> == s.
```

Likewise, a disyllabic word root such as the German *Tutor* can be specified in terms of the individual components of its two syllables[10]:

```
Tutor:
    <> == Noun_L
    <phn root form>  == Disyllable
    <phn syl2 onset> == t
    <phn syl2 peak>  == u:
    <phn syl1 onset> == t
    <phn syl1 peak>  == O
    <phn syl1 tail>  == r.
```

[10]Default information for a lexeme node like this comes from the declensional class node, in this case, **Noun_L**.

From these node definitions, taken together with the axioms for syllable structure given above, we can now infer that:

```
Suffix_es:
    <phn root form> = @ s.

Tutor:
    <phn root form> = t u: t O r.
```

4. The representation of allomorphy

Within this framework, there are two principal methods for representing allomorphy: (i) the use of path extensions on the left hand side of equations and (ii) the use of conditional statements on the right hand side of equations. These two methods can also be combined. In discussing the applicability of these two approaches, we make a distinction between the variant and inherent properties of a class of lexemes: nouns, for example, have gender as an inherent morphosyntactic property whilst case and number are variant morphosyntactic properties. The variance or inherence of a property is relative to the class of lexemes involved, thus adjectives, for example, have gender as a variant property, not an inherent one[11].

4.1. Path extensions

When querying the form of a word, a query path is invoked that is partly composed of attributes representing the particular values of the variant morphosyntactic properties of the lexeme involved. So, to find the form of the genitive singular of a noun, for instance, the query path would be `<mor word sing gen>`. The morphological word is defined, by default, as a root followed by a (possibly null) suffix. The `Word` node, from which all word class nodes and ultimately all words inherit by default, thus appears as follows:

```
Word:
   <> == Syllable
   <mor word> == "<phn root form>" "<mor suffix>".
```

Given this definition, the query path `<mor word sing gen>` leads to the phonological form query (`<phn root form>`) having the variant morphosyntactic attributes appended, so the query path for the root is `<phn root form sing gen>`. This allows us to define realizations which are contingent on variant morphosyntactic properties by specifying the relevant attributes in appropriate path equations as follows:

```
Noun_L:
    ...
    <phn syll peak plur> == Lengthen:<"<phn syll peak>">.
```

[11]Note that we are making the distinction with respect to classes of lexemes, not individual lexemes. There is a sense in which the noun lexeme *trousers* is inherently plural, but that sense is not to the point here.

which says that if the feature **plur** is present in the query path then the peak is realized by application of the **Lengthen** function.

Several examples of this kind of allomorphy can be found in the three PolyLex languages. In one class of Dutch nouns the stem vowel in the plural form is always /**e:**/, regardless of what vowel the singular form has, e.g. *stad/steden, lid/leden.* This is captured in PolyLex in the following manner:

```
Noun_e:
    ...
    <phn syl1 peak plur> == e:.
```

English nouns which have a final voicing alternation, such as *wife/wives, house/houses* can be accounted for in a similar way, the realization of their final coda being dependent on whether the form is singular:

```
Noun_D:
    ...
    <phn syl1 coda sing> == Devoice:<"<phn syl1 coda>">.
```

This is just a restricted application of final consonant devoicing, something which applies more generally in German and Dutch.

German umlaut is the classic example of this type of alternation, and is interesting in the present context because of the fact that the relevant morphosyntactic property differs in nouns and verbs. In German nouns which belong to one of the declensional classes which undergoes umlaut, the umlaut function applies only in the plural forms:

```
Noun_U:
    ...
    <phn syll peak plur> == Umlaut:<"<phn syll peak>">.
```

However, in one class of verbs, the (relevant) vowel undergoes umlaut in past tense forms:

```
Verb_U:
    ...
    <phn syll peak past> == Umlaut:<"<phn syll peak>">.
```

All three languages exhibit this type of allomorphic variation in their numeral forms, with variation between, for example, *twee/twin-, two/twen-, zwei/zwan-*. In our account of the numerals expressions (Cahill & Gazdar, 1996) we capture this alternation by the use of morphosyntactic features to indicate the "teen" and "ty" forms of the numerals. Thus, the form of a numeral combined with either "teen" or "ty" is marked with an attribute **bound** that encodes a variant morphological property of morphemes (free/bound). Given this attribute, the variation in forms can be stated as follows:

```
Phon002:
    <> == Syllable
    <M phn body bound>  == n

    <D phn onset>       == t w
    <D phn peak>        == 'e:'
    <D phn peak bound>  == I

    <E phn onset>       == t
    <E phn onset bound> == t w
    <E phn peak>        == 'u:'
    <E phn peak bound>  == E

    <G phn onset>       == t s v
    <G phn peak>        == a i
    <G phn peak bound>  == a.
```

4.2 Conditional statements

The use of path extensions is the natural way to deal with allomorphic variation that is conditioned by variant properties of the unit involved. But it cannot be used for inherent properties of the unit since such properties will not be represented in the attributes that specify the inflected form. In such cases a different approach is required.

The approach adopted in PolyLex employs one of the most common idioms of modern programming languages, the **if ... then ... else ...** construct. In DATR, this construct takes the following form:

`IF:<`*condition* `THEN` *value1* `ELSE` *value2*`>`

where the condition is stated as some boolean combination of atomic truth-valued statements and *value1* and *value2* are phonological units (segments, for example). The atomic statements may involve predicates, such as `SCHWA`, `SIBILANT`, `VOICED`, `FEMININE`, and `ANIMATE` applied to arguments denoting phonological, morphological, or lexical units.

The *condition* can thus refer to any lexical information available, not just phonological. So, for instance, the realization of a phonological constituent may be determined by phonological aspects of the root or suffix, syntactic gender of the root or even semantic properties of the root (e.g., animacy in Russian noun inflection).

One of the major noun classes in Dutch suffixes an *-s* in the plural. The phonological realization of this is dependent on whether the root ends in a sibilant or not, taking the form `/@s/` if it ends in a sibilant and `/s/` otherwise. This suffix node is defined in PolyLex as follows:

```
Suffix_S:
    <> == Affix
```

```
<phn syl1 tail>  == s
<phn syl1 peak>  == IF:<SIBILANT:<"Root:<phn syl1 tail>">
                    THEN @
                    ELSE Null>.
```

German has an identical suffix, with identical variants. English also has an *-s* suffix, but because of the absence of final consonant devoicing in English, it also has a voicing contrast. This therefore requires two conditional statements, one for the peak which is identical to that for Dutch and one for the tail, stating that if the root final consonant is either voiced or a sibilant then the tail of the suffix is /z/ and otherwise it is /s/:

```
Suffix_Z:
    <> == Affix
    <phn syl1 peak> == IF:<SIBILANT:<"Root:<phn syl1 tail>">
                       THEN @
                       ELSE Null>
    <phn syl1 tail> == IF:<OR:<VOICED:<"Root:<phn syl1 tail>">
                            SIBILANT:<"Root:<phn syl1 tail>">>
                       THEN z
                       ELSE s>.
```

German has two noun classes which suffix *-e*, one with umlaut and the other without. We include in these classes nouns which do not inflect in their plural (or which only umlaut the peak) where this is phonologically determined. The phonological requirement in these cases is that the final syllable must have a schwa peak. So the noun *Adler* has the singular and plural form /a:dl@r/. We capture this in PolyLex by defining a suffix node **Suffix_e2**, distinct from **Suffix_e1**, which incorporates the alternation between /@/ and null:

```
Suffix_e2:
    <> == Affix
    <phn syl1 rhyme> == IF:<SCHWA:<"Root:<phn syl1 peak>">
                        THEN Null
                        ELSE "Suffix_e1:<phn syl1 rhyme>">.
```

4.3. Combined path extensions and conditionals

Some allomorphic alternations involve both variant and inherent properties. In such cases it is appropriate to combine the two approaches to allomorphy outlined above. In the following hypothetical (but linguistically plausible) example, the plural form of the peak is /@/ if the final segment of the root is a sibilant and /I/ otherwise, and the singular is always /a/:

```
Noun_a:
    <phn syl1 peak plur> == IF:<VOICED:<"Root:<phn syl1 tail>">
                            THEN e:
```

```
                                ELSE e>
    <phn syl1 peak>         == a.
```

There are not many examples of this kind in the PolyLex lexicons but one notable one is the German singular genitive suffix which only appears on masculine and neuter nouns. We can encode this fact as follows:

```
Suffix_s:
    <> == Affix
    <phn syl1 rhyme sing gen> ==
                        IF:<FEMININE:<"Root:<syn gender>">
                        THEN Null
                        ELSE "Suffix_S">.
```

The rhyme of this suffix is null if the syntactic gender of the noun is feminine and otherwise is inherited from the `Suffix_S` node, which as we have seen above, incorporates an additional phonological condition.

5. Conclusions

We have described the principal ways of representing allomorphic variation within the PolyLex lexicons. Alternations which are dependent solely on variant properties of the unit in question are captured with path extensions. Alternations which are dependent solely on inherent properties of the unit in question are captured with conditionals. These conditional statements may refer to any lexically available inherent information. In the case of the PolyLex languages, this includes morphosyntactic information (noun gender) but mostly involves phonological information about the root. When an alternation involves both variant and inherent properties of the units involved, then it is necessary to combine the use of path extensions with the use of conditionals. We have illustrated these methods with examples drawn from the PolyLex lexicon(s) for Dutch, English and German.

References

[1] Harald Baayen, Richard Piepenbrock & H. van Rijn (1995) The CELEX Lexical Database, Release 2 (CD-ROM). Linguistic Data Consortium, University of Pennsylvania, Philadelphia, PA.

[2] Doris Bleiching (1992) Prosodisches Wissen in Lexicon. In G. Görz, ed., *Proceedings of KONVENS-92,* Berlin: Springer-Verlag, 59-68.

[3] Doris Bleiching (1994) Integration von Morphophonologie und Prosodie in ein hierarchisches Lexicon. In Harald Trost, ed., *Proceedings of KONVENS-94,* Vienna: Österreichische Gesellschaft für Artificial Intelligence, 32-41.

[4] Doris Bleiching, Guido Drexel & Dafydd Gibbon (1996) Ein Synkretismusmodell für die deutsche Morphologie. In Dafydd Gibbon, ed., *Natural Language Processing and Speech Technology: Proceedings of KONVENS-96, Bielefeld.* Berlin: Mouton de Gruyter, 237-248.

[5] Dunstan Brown, Greville Corbett, Norman Fraser, Andrew Hippisley & Alan Timberlake (1996) Russian noun stress and network morphology. *Linguistics*, 34.1, 53-107.

[6] Dunstan Brown & Andrew Hippisley (1994) Conflict in Russian genitive plural assignment: A solution represented in DATR. *Journal of Slavic Linguistics*, 2.1, 48-76.

[7] Lynne Cahill (1990a) Syllable-based morphology. *Proceedings of the 13th International Conference on Computational Linguistics, COLING-90*, Vol. 3, 48-53.

[8] Lynne Cahill (1990b) Syllable-based morphology for natural language processing. DPhil Dissertation, University of Sussex, available as Cognitive Studies Research Paper 181.

[9] Lynne Cahill (1993) Morphonology in the lexicon. *Sixth Conference of the European Chapter of the Association for Computational Linguistics*, 87-96.

[10] Lynne Cahill & Gerald Gazdar (1996) A lexical analysis of numeral expressions in three related languages. *Proceedings of the AISB Workshop on multilinguality in the lexicon*, Brighton, UK, 69-75.

[11] Lynne Cahill & Gerald Gazdar (1997) The inflectional phonology of German adjectives, determiners and pronouns. *Linguistics* 35.2, 211-245.

[12] Lynne Cahill & Gerald Gazdar (forthcoming) German noun inflection. To appear in *Journal of Linguistics.*

[13] Andrew Carstairs[-McCarthy] (1987) *Allomorphy in Inflection.* London: Croom Helm.

[14] Roger Evans & Gerald Gazdar (1996) DATR: A language for lexical knowledge representation. *Computational Linguistics*, 22.2, 167-216.

[15] Norman Fraser & Greville Corbett (1995) Gender, animacy, and declensional class assignment: a unified account for Russian. In Geert Booij & Jaap van Marle, eds. *Yearbook of Morphology 1994.* Dordrecht: Kluwer, 123-150.

[16] Norman Fraser & Greville Corbett (in press) Defaults in Arapesh. To appear in *Lingua.*

[17] Dafydd Gibbon (1990) Prosodic association by template inheritance. In Walter Daelemans & Gerald Gazdar, eds. *Proceedings of the Workshop on Inheritance in Natural Language Processing.* Tilburg: Institute for Language Technology, 65-81.

[18] Dafydd Gibbon (1992) ILEX : a linguistic approach to computational lexica. In Ursula Klenk, ed. *Computatio Linguae: Aufsatze zur algorithmischen und quantitativen Analyse der Sprache (Zeitschrift fur Dialektologie und Linguistik,* Beiheft 73), Stuttgart: Franz Steiner Verlag, 32-53.

[19] Dafydd Gibbon & Doris Bleiching (1991) An ILEX model for German compound stress in DATR. *Proceedings of the FORWISS-ASL Workshop on Prosody in Man-Machine Communication*, 1-6.

[20] Jaap van Marle (1985) *On the Paradigmatic Dimension of Morphological Creativity.* Dordrecht: Foris.

[21] Peter Matthews (1972) *Inflectional Morphology: A Theoretical Study Based on Aspects of Latin Verb Conjugation.* Cambridge: Cambridge University Press.

[22] Sabine Reinhard (1990) Verarbeitungsprobleme nichtlinearer Morphologien: Umlautbeschreibung in einem hierarchischen Lexikon. In Burghard Rieger & Burkhard Schäder *Lexikon und Lexikographie.* Hildesheim: Olms Verlag, 45-61.

[23] Sabine Reinhard & Dafydd Gibbon (1991) Prosodic inheritance and morphological generalisations. *Fifth Conference of the European Chapter of the Association for Computational Linguistics*, 131-136.

[24] Stuart M. Shieber (1986) *An Introduction to Unification Approaches to Grammar.* Stanford: CSLI/Chicago University Press.

[25] Gregory T. Stump (1992) On the theoretical status of position class restrictions on inflectional affixes. In Geert Booij & Jaap van Marle, eds. *Yearbook of Morphology 1991.* Dordrecht: Kluwer, 211-241.

[26] Gregory T. Stump (1993a) On rules of referral. *Language* 69, 449-479.

[27] Gregory T. Stump (1993b) Reconstituting morphology: The case of Bantu preprefixation. *Linguistic Analysis* 23, 169-204.

[28] Gregory T. Stump (1993c) Position classes and morphological theory. In Geert Booij & Jaap van Marle, eds. *Yearbook of Morphology 1992.* Dordrecht: Kluwer, 129-180.

[29] Gregory T. Stump (1995) The uniformity of head marking in inflectional morphology. In Geert Booij & Jaap van Marle, eds. *Yearbook of Morphology 1994.* Dordrecht: Kluwer, 000-000.

[30] John C. Wells (1987) Computer coded phonetic transcription, *Journal of the International Phonetic Association*, 17:2, 94-114.

[31] Wolfgang Wurzel (1990) The mechanism of inflection: lexicon representations, rules, and irregularities. In Wolfgang U. Dressler, Hans C. Luschützky, Oskar E. Pfeiffer & John R. Rennison, eds. *Contemporary Morphology.* Berlin: Mouton de Gruyter, 203-216.

[32] Arnold Zwicky (1985) How to describe inflection, *BLS* 11, 371-386.