

- Literatur. Geistes- und sozialwissenschaftliche Klasse, Mainz. 1983:6. Wiesbaden: Franz Steiner Verlag.
- Van Valin, Robert D. 2001. *An introduction to syntax*. Cambridge: Cambridge University Press
- Van Valin Robert D. & Randy J. LaPolla. 1997. *Syntax: structure, meaning and function*. Cambridge: Cambridge University Press.
- Winter, Werner. 1980. 'Morphological signalling of selection properties: Transitivity in Tocharian B and A verbs'. In Jacek Fisiak (ed.), *Historical morphology*, 421-42. The Hague: Mouton de Gruyter.
- Winter, Werner. 1990. 'The importance of fine points in spelling: deletion of accented vowels in Tocharian B'. In Jacek Fisiak (ed.), *Historical linguistics and philology*, 371-91. Berlin: Mouton de Gruyter.

Gerd Carling <Gerd.Carling@ling.lu.se>

Analogical morphology is undecidable

Mats Eeg-Olofsson

Analogical morphology

Analogical morphology (Eeg-Olofsson 1989) is a generative morphological theory based on the intuitive concept of analogy. In analogical morphology this concept is made explicit by being formalized in special derivational rules. Such formal analogies can be used for deriving word forms from other word forms derived previously. The basis of all derivations is a lexicon consisting of fully specified word forms. A word form $W = (S, G)$ is defined formally as a pair consisting of a string, S , and a list of features, G ("grammar"). Strings are made up of characters representing segments like phonemes or graphemes. Features may specify number, gender, declension etc.

Definition of analogy

An analogy can be described as a quintuple $(P1, G1, P2, G2, C)$, where $P1$ and $P2$ are string patterns and $G1, G2$, and C are feature specifications. It is to be interpreted as the statement that a word form that has the features $G2$ and matches the string pattern $P2$ can be derived from any word form matching $P1$ with features $G1$. In addition, the word forms must both have the feature values specified in C , which is a list of common feature values.

For example, an analogy like

$([X], [\text{number:sing}], [X, "s"], [\text{number:plur}], [\text{cat:n}, \text{gender:G}])$

might be employed to describe plural formation in some language by suffixation of the string s . X is a string variable, in this case matching the entire word form string in the singular. The word forms must both have n as the value of the feature cat . In addition, the variable G creates a linkage for the gender feature. Both word forms must have the same value for this feature, e.g. both masculine or both feminine.

More formally, derivation by application of an analogy can be described as follows:

a) The string part of word form 1 (the source word form) is matched with the string pattern P1. If the match fails, the application is blocked. Otherwise, string values are assigned to the variables in the string pattern (e.g. X in the above example), as a result of the match.

b) The feature part of word form 1 is checked for compatibility with the unified feature specifications G1 and C. On failure, the derivation is blocked. Otherwise, if C contains linkage variables over feature values (e.g. G for gender in the above example), specific values (e.g. *masculine*) from word form 1 may be assigned to these variables.

c) The string part of the resulting word form 2 is obtained from the pattern P2 by substituting for the variables in P2 the string values assigned to them as a result of the matching in step a).

d) The feature part of word form 2 is the unification of G2 and C, where feature values from word form 1 have been assigned to the linkage variables in step b).

The compatibility check and linkage variable assignment in step b) can be implemented by destructive unification of the feature list of word form 1 with G1 and C.

It is evident from the above definition that the variables in P2 must all occur in P1 as well. If both the patterns P1 and P2 in an analogy contain the same variables, it makes sense to define *backward derivation* as backward analogical application. Backward application of an analogy can be described as substituting the index 1 for 2, and vice versa, in the above definition. For any derivation of a word form from the lexicon there is a corresponding backward derivation that takes the word form as point of departure and goes back to the lexicon (Eeg-Olofsson 1994).

Variables in string patterns

Variables in the patterns are of two kinds. Essential variables ('arbs') are denoted by single capital letters (e.g. X), matching any string (including the empty string). Restricted 'don't cares' match any single character in a specified set and are denoted by constructions like any("sxz", C), where "sxz" is the character set in question and C is the variable. Special abbreviations, e.g. vowel(C) for any("aoueyi", C), may be introduced for frequently occurring constructions. The abbreviation any(C) can be used for unrestricted don't cares, matching any single character in the entire character set.

It goes without saying that multiple occurrences of a variable in a string pattern must all denote the same string value. This can be exploited to

introduce some context sensitivity into the pattern matching. In this way analogical morphology can readily account for structural reshaping like *reduplication*. An example is the alternation *πεπαιδευκα - παιδευω* in classical Greek of forms of the verb meaning 'raise, bring up', representing one of the ways in which the perfect tense is constructed. This particular kind of reduplication can be represented by the pattern pair (P1:P2)

$$[\text{cons}(C), "ε", \text{cons}(C), \text{vowel}(V), X] : [\text{cons}(C), \text{vowel}(V), X]$$

The designation 'essential variables' for arbs suggests that other variables, i.e. the variables associated with don't cares, are in some way inessential. They can indeed be dispensed with without affecting the generative power of analogical morphology. Since these variables can only take on a finite number of values, they can be eliminated from the analogies by being systematically replaced with all possible values. For example, an analogy containing a pattern pair

$$[X, \text{any}("sxz", C)] : [X, \text{any}("sxz", C), "es"]$$

can be replaced with a set of three analogies containing the pattern pairs

$$\begin{aligned} [X, "s"] &: [X, "s", "es"] \\ [X, "x"] &: [X, "x", "es"] \\ [X, "z"] &: [X, "z", "es"] \end{aligned}$$

respectively. Such variables still serve a useful purpose, expressing valid linguistic (e.g. phonological) generalizations and providing for a more compact notation.

Analogical morphology without features

Different kinds of feature values

In syntactic theories like Modern Phrase Structure Grammar (see e.g. Borsley 1996) and Lexical Functional Grammar (Bresnan 1982) essential use is made of feature structures containing features that have other feature structures as values, but in the above examples of analogical morphology, all features only have values taken from a finite number of atoms. For instance, the *number* feature may have the (atomic) values *singular* and *plural*.

Purpose of features

Features serve a useful purpose in analogical morphology, even when used in this restricted way. Generally speaking, the wide-spread use of features in all

branches of modern linguistics is motivated by the fact that features are well suited to expressing generalizations, systematic similarities and differences. Their main function in analogical morphology is to liberate the representation of linguistic content from the representation of expression. A case in point is *apophony*, whereby alternating stem vowels are used as exponents of various verb tenses, e.g. Swedish *sitta-satt-suttit, fara-for-farit*. The crucial point here is that there is not any permanent association between tense and stem vowel. For instance, in the above example *a* occurs as stem vowel in the infinitive as well as in the past and perfect tense forms.

Atomic features do not affect decidability of analogical morphology

The main result of this paper is the undecidability of analogical morphology. Thus, analogical morphology can be considered too powerful for effective computation. It is then a legitimate question whether this excessive expressivity is caused by the feature mechanism of the system. The question can be answered in the negative for systems where all features have finite numbers of atomic values. Such systems are in a certain sense equivalent to systems having no features. In the next section it will be shown that even such "text-only" systems are undecidable. Consequently, it is the pattern matching mechanism rather than the feature mechanism that is "responsible" for the undecidability of analogical morphology.

Translation of features into text

Analogical morphology systems with features having finite numbers of atomic values can be 'translated' into text-only systems. The main idea is to express the feature values as text in the string that represents a word form.

The device of coding feature values as text is used heavily in two-level morphology (Koskenniemi 1983), where features are represented as 'diacritics' without any surface realization. Feature values in lexical entries are coded as single digits, punctuation signs etc, so that they can be sensed by the two-level correspondence rules. Two-level morphology has been criticized for this 'segmentalization' of features, which obscures the distinction between linguistic content and expression.

Here, the feature values will be kept in a special 'compartment' of the word form string. Pattern matching and substitution can be used to manipulate this compartment as well as the compartment containing the word form proper (the word body). A special character not used elsewhere, say, #, is employed to separate the two string compartments. Thus, a string like

`number.singular/person.3/#goes`

might be used to represent the English word form *goes*. The word body is kept in the right hand side compartment, whereas the features are kept in the left compartment. Specifications of feature values are separated by slashes (or any other character that is not used for any other purpose). Feature names (like *number*) are separated from their values (e.g. *singular*) by another special-purpose character, e.g. the dot. All features should have specified values in all word form representations. If a feature is considered irrelevant in some word forms, e.g. tense in nouns, its value can be represented by the string *irrev* (or any other such special-purpose string). Finally, in order to ensure a unique representation of feature specifications, feature names (with their values) should be ordered systematically, e.g. alphabetically, within their compartment.

It is evident that there is a simple correspondence between ordinary lists of feature values and the feature string compartments constructed in the above way. Given a feature string compartment, a feature value list can be constructed uniquely, essentially by deletion of irrelevant features (if any). Conversely, an ordinary feature value list can be expanded into a feature string compartment by addition of variables to represent values of relevant features not specified in the original list. In addition, irrelevant features must be represented by the conventional value (e.g. *irrev*).

An example may help to clarify the correspondence. The fictitious analogy

`([X], [number:sing], [X, "s"], [number:plur], [gender:G])`

might be translated into the following text-only analogy:

`(["gender.", G, "/number.sing/person.irrev/#", X],
["gender.", G, "/number.plur/person.irrev/#", X, "s"])`

where the variable *G* is employed to represent the common value of the *gender* feature, whereas the *person* feature is considered irrelevant.

Undecidability of analogical morphology without features

Given an arbitrary string *W*, a set of analogies *A*, and a lexicon *L* one might ask whether *W* is a legitimate word form with respect to *A* and *L* (neither of which includes features). This is tantamount to asking whether *W* can be derived from some string in *L* in a finite number of steps through application of analogies in *A*.

For some specific instances of this general problem, i.e. certain choices of W , A and L , it is possible to find the (affirmative or negative) answer to the above question. But there is no *algorithm* that can answer the question, given an arbitrary instance of the problem as input. In other words, it is not possible to construct a formal procedure that is guaranteed to halt with the correct decision for any combination of W , A and L .

Idea of proof: Reduction of Post's Correspondence Problem (PCP)

This undecidability of analogical morphology without features can be proved by reduction of a well-known undecidable problem, Post's Correspondence Problem (PCP). It can be shown that if analogical morphology were decidable, PCP would also be decidable. Since PCP is known to be undecidable (see, e.g., Hopcroft & Ullman 1969), analogical morphology must also be undecidable.

Description of PCP

PCP can be stated as follows: W and X are two lists of (non-empty) strings over the same alphabet, with the same number of strings, say, k , in each list. In other words, an integer index ranging between 1 and k can be assigned to each string in W and X , denoting its position in the list. Consequently W and X can be described as, say,

$$W = w_1, w_2, \dots, w_k \text{ and } X = x_1, x_2, \dots, x_k$$

This instance of PCP is said to have a solution if there is a (non-empty) sequence of integers i_1, i_2, \dots, i_m such that

$$w_{i_1}, w_{i_2}, \dots, w_{i_m} = S = x_{i_1}, x_{i_2}, \dots, x_{i_m}$$

In other words, there is a string S that can be constructed by concatenation of strings in W in such a way that concatenation in the same order of the corresponding strings in X yields the same result, S .

A trivial example is $W = \text{bee}, n$ and $X = \text{be}, \text{en}$; evidently the string *been* can be constructed by concatenating the first and second string in each list. It is less obvious that the instance of PCP with lists $W = \text{da}, \text{add}, \text{dad}$ and $X = \text{dad}, \text{dd}, \text{add}$ does not in fact have any solution.

Reduction of PCP to analogical morphology

Any instance of PCP can be reduced to a corresponding problem in analogical morphology by the following method. Suppose that the instance of PCP is described by the lists W and X of strings over some alphabet V . The alphabet

of the corresponding analogical morphological system includes V , in addition to two special characters, $\&$ and $\#$, that are not members of V . $\#$ is used as a delimiter in analogies. For any pair of corresponding strings in W and X , say, w_i and x_i , there is an analogy

$$P, \#, Q : w_i, P, \#, x_i, Q$$

(where P and Q are variables denoting arbitrary strings).

The lexicon of the analogical morphological system consists of the single string $\#$.

The special character $\&$ is used in the analogy

$$P, \text{any}(Q), \#, P, \text{any}(Q) : \&$$

but nowhere else in the analogical morphological system. (The pattern $\text{any}(Q)$ is included to block direct derivation of $\&$ from the lexical axiom $\#$.)

Evidently, this instance of PCP has a solution if and only if the 'word' $\&$ can be derived in the corresponding analogical morphological system constructed in this way. If there were an algorithm to decide whether an arbitrary string can be derived in an arbitrary analogical morphological system, PCP would also be decidable.

References

- Borsley, Robert D. 1996. *Modern phrase structure grammar*. Oxford: Blackwell Publishers.
- Bresnan, Joan. 1982. *The mental representation of grammatical relations*. Cambridge, Massachusetts: MIT Press.
- Eeg-Olofsson, Mats. 1989. 'A morphological Prolog system for Swedish based on analogies'. In Östen Dahl & Kari Fraurud (eds.), *Papers from the first Nordic conference on text comprehension in man and machine*, 49-62. Institute of Linguistics, University of Stockholm.
- Eeg-Olofsson, Mats. 1994. 'Reversibility in analogical morphology'. *Working papers* 42, 5-13. Department of Linguistics, Lund University.
- Hopcroft, John E. & Jeffrey D. Ullman. 1969. *Formal languages and their relation to automata*. Reading, Massachusetts: Addison-Wesley.
- Koskeniemi, Kimmo. 1983. *Two-level morphology: a general computational model for word-form recognition and production*. Department of General Linguistics, University of Helsinki.

Mats Eeg-Olofsson <Mats.Eeg-Olofsson@ling.lu.se>