Lund University, Dept. of Linguistics
Working Papers 42 (1994), 5-13

Reversibility in Analogical Morphology

Mats Eeg-Olofsson

1. Introduction

The fundamental problem of computational morphology is to account in a computationally tractable way for the relation between surface word forms and underlying lexical units like stems or morphemes. Practical applications of computational morphology work in opposite directions. Analysis goes from surface word forms towards the lexicon with its content specification, while synthesis (generation) takes the lexicon as its point of departure in order to express a given content.

2. Reversibility

The word *reversibility* (or, bidirectionality), when characterizing some kind of linguistic knowledge representation for computational purposes, means that the representation can be used for both analysis and synthesis. It is not clear from the point of view of psychological plausibility whether reversibility is a desirable property. From a practical point of view, however, reversibility is a distinct advantage, contributing to a suitable division of labour between the linguist and the programmer. Ideally, a reversible formalism should enable the linguist to work out a description of a language once and for all, without regard to its potential uses, leaving the programmer the task of devising suitable programs that can interpret the formalism for both analysis and synthesis.

2.1 Reversibility in two-level morphology

The two-level model of Koskenniemi (see Koskenniemi 1983) is the system that has had the greatest single influence on present-day computational morphology. It is frequently viewed as the background against which new proposals should be evaluated. Factors that have contributed to the success of the two-level model are its applicability to a great number of different languages and its alleged bidirectionality.

The statement that two-level morphology is bidirectional must be taken with a grain of salt. Given a sequence of morphs representing a word form at the lexical level, two-level morphology can describe the surface level realization. But it cannot directly describe the various realizations of morphemes, taken to be units of form rather than of substance. The reason for this is that two-level morphology does not contain any systematic means for describing the relations between morphs and morphemes.

3. Analogical morphology

Analogical morphology (Eeg-Olofsson 1989) has been introduced as a reversible system for morphological knowledge representation with a suitable interface to syntax. The meaning (in a broad sense) of a word form is represented as a set of feature values. In the formalism of analogical morphology such bundles of feature values are written as lists of category-value pairs. For instance, the representation of the Swedish word form *hästarnas* 'the horses'' might be

(root:häst, number:plural, definiteness:definite, case:genitive)

where each colon separates the name of a feature category from its value. The value of a feature is either undefined (in which case the feature category is left out from the feature list) or taken from a finite set of atomic values.

The basic units of analogical morphology are concrete word forms, defined as pairs consisting of a string part and a feature structure part. Word forms considered more central than others belong to the lexicon and may be called lexemes. Thus the lexicon is a finite list of lexemes.

Word forms (including grammatical descriptions) that are not found in the lexicon can be derived from lexemes by the application of analogies.

3.1 Analogies

An analogy relates a word form with a certain shape and kind of feature structure to a similar word form with a corresponding shape and feature structure. The shapes of the word forms are described by string pattern expressions including variables. The descriptions of the feature structures may include specifications of the values of particular features as well as *linkages.* A linkage for some feature between two word forms prescribes that the values associated with the feature in the two word forms must be identical.

For instance, the following analogy describes a relation between the comparative and the superlative forms of Swedish adjectives:

(X,"re") / (X,"st") = (deg:comp) / (def:indef,deg:sup)
(cat:a,comp:yes,infl:Alpha,neuinfl:Beta)

This analogy relates comparative (deg:comp) forms ending in *-re* to superlative indefinite (def:indef,deg:sup) forms ending in *-st.* Common feature values are specified by the second line of the analogy. Thus, the word forms must both be synthetically comparable (comp:yes) adjectives (cat:a). The linkages proper are described by the part (infl:Alpha,neuinfl:Beta), which states that the values for the features infl and neuinfl, though unspecified in the analogy, should be identical in the two word forms (technically, by being unified with the variables Alpha and Beta, respectively).

Thus, an analogy has a *left side* and a *right side,* each containing a pattern expression (in this case (X,"re") and (X,"st"), respectively) and a feature specification (here (deg:comp) and (def:indef,deg:sup), respectively). In addition, there is a description of common feature values, specified either as constant values (cat:a,comp:yes) or linkages (infl:Alpha,neuinfl:Beta).

3.2 Abstract generation

An analogical description of the morphology of some language consists of a lexicon and a system of analogies. The language of word forms generated by an analogical description is defined as the union of the lexicon and all word forms whose existence is implied by the lexicon by (possibly repeated) application of analogies. An analogy can be used for generation by being applied to a word form to produce a new related word form. A series of such applications may be called a *generation path,* if it starts from the lexicon and the output of each application (except the last one) provides the input for the next application.

Application. Application consists of compatibility checking followed by construction. The compatibility test compares a given word form with one of the sides of the analogy to be applied. If the test is successful, a new word form is constructed according to the description of the other side of the analogy. Thus, analogies can be applied in two directions. Forward application means that the left side of a rule is used as input specification and the right side as output specification. Backward application is the converse of forward application.

Compatibility checking consists of string pattern matching and feature compatibility checking.

If the input string pattern matches the input word form string, the variables in the pattern (X in the above example) can be instantiated to the matching substrings, and the output string can be constructed from the output string pattern, which contains these variables (and, possibly, constants). For instance, the word *högre* 'higher' matches the pattern (X,"re"), instantiating the variable X to the string "hög", which can then be substituted for X in the pattern (X,"st") to produce "högst" 'highest'.

Compatibility between two feature structures means that no feature category in the structures has different values in the structures. Consequently, two feature values are compatible if at least one of them is unspecified, or if they both have the same constant value.

If the compatibility checks are successful, the output feature structure can be constructed. The values in this structure can be determined either by the output feature specification or by linkage. Otherwise, they are left unspecified.

Compatibility checking and output construction can both be implemented by feature unification.

3.3 Analysis and synthesis

Generation in the above abstract sense must be distinguished from the problem of synthesis. Given a feature specification F, the problem of synthesis is to find all S, such that S is the string part of some word form that has feature specification F, and can be generated (in the abstract sense) from the lexicon. Correspondingly, analysis is defined as finding all feature structures F, such that F is the feature specification of some word form that can be generated from the lexicon with a given string part S.

3.4 The analogical relation interpreted as equivalence

The relation described by an analogy might be interpreted as a kind of equivalence. Equivalence means that there is a word form compatible with the description of any one of the sides of an analogy if and only if there is a corresponding form described by the other side.

3.4.1 Advantage of equivalence interpretation: Reversibility

The attraction of the equivalence interpretation is that it guarantees reversibility. Analysis as well as synthesis can be performed by successive backward application of analogies.

Analysis. Analysis starts with a given string and an empty feature structure. In the course of the analysis process, features of the original structure may become instantiated through linkage and other compatibility conditions. The process stops when such a backward generation path reaches the lexicon.

Synthesis. Synthesis starts with an unknown string and a given feature specification (which, typically, includes some root or meaning representation). The output pattern specification of an analogy describes the output string as a concatenation of constant and variable pattern elements, where the variables obtain their values by matching parts of the input string. The input string, in turn, typically is the output string of some other analogy, etc. When the process reaches the lexicon, the last pattern expression can be solved for its variables and these values can be substituted for the variables into the next to last pattern expression to yield a string value, which can then be used for pattern matching determining other variable values etc. In this way the original unknown string can be determined by successive pattern matching and substitution.

It is presupposed that the same variables appear in both the input pattern specification and the output pattern specification of any analogy. Also, the patterns must be such that they match any string in at most one way.

3.4.2 Disadvantage of equivalence interpretation: No rules with exceptions

The most convenient way to describe many morphological relations is to view them as general rules with exceptions. For instance, it is natural to describe the formation of the plural forms of Swedish adjectives as a general analogy

(X) / (X,"a") = (def:indef,gen:utr,num:sing) / (num:plu)
(cat:a,comp:Alpha,deg:pos,infl:yes,neuinfl:Beta)

relating forms like *blek* 'pale' and *bleka* 'pale-Plural', and supplemented by exception analogies like

(X,"ad") / (X,"ade") = (def:indef,gen:utr,num:sing) / (num:plu)
(cat:a,comp:Alpha,deg:pos,infl:yes,neuinfl:Beta)

relating forms like *konstlad* 'artificial' and *konstlade* 'artificial-Plural'. It would be more complicated and much less natural to describe the relation by analogies with mutually exclusive patterns, e.g. one for forms ending in *-ad* and another for forms whose last letter is not *d* or whose next to last letter is not *a*.

Definition of generation modified for default logic. If default logic in the form of rules with exceptions is to be admitted into the system of analogical morphology, the definition of generation must be changed slightly. In the process of generation, whenever there is a choice between analogies, the one with the most specific pattern and feature conditions must take precedence over analogies with more general conditions. Consequently, an analogy must never be used when a more specific analogy is applicable, and the only real choice is between analogies whose degree of specificity cannot be directly compared.

Incompatibility of equivalence interpretation and default logic. Unfortunately, the relation expressed by analogies involved in systems of rules with exceptions cannot be interpreted as equivalence. For instance, in the case of the general plural-forming analogy above, it is not true that a certain adjective form ending in *-a* exists if and only if there is a certain other adjective form lacking the *-a*.

3.5 Analogical relation re-interpreted as default implication

A possible replacement for equivalence as the interpretation of the analogical relation is what might be called *default implication:* The existence of a form described by the left side of an analogy implies the existence of a form corresponding to the right side of the analogy, *if no other analogy with a more specific left side is applicable.* Thus, in addition to default logic, the new interpretation introduces the restriction that generation must consist of forward application steps only.

4. Modified reversibility under default implication

The main result of this paper is the discovery that reversibility can be maintained in a modified form under the default implication interpretation of the analogical relation. It is formulated in the following Theorem:

Theorem (Modified reversibility): For every generation path from a lexeme L to a word form W there is a backward generation path from W to L.

It should be noted that the converse is not true. For instance, by use of the default plural-forming analogy above, the form **konstlada* can be analysed as an inflexion of the lexeme *konstlad.* But of course **konstlada* cannot be generated from *konstlad,* since the exception analogy takes precedence over the default, producing instead the correct form *konstlade.*

The modified reversibility theorem means that analysis and synthesis in analogical morphology can both be performed in two passes, as a kind of double generation. The first pass is backward generation, deriving a lexeme from the given string (analysis) or feature structure (synthesis), without consideration of exceptions. The second pass is forward re-generation, checking whether the backward generation steps can be reversed, with due consideration of exceptions and defaults. Spurious backward generation paths are revealed by this second pass.

Proof of the Theorem: The truth of the Theorem follows from the reversibility of each single generation step.

Obviously, the conditions on string patterns stated above (under Synthesis) guarantee that the two strings involved in an analogy application determine each other uniquely.

The reversibility of feature structure specifications is expressed by the following Lemma:

Lemma: Suppose that the feature structure f1 is compatible with the input feature specification F1 of a certain analogy and that f2 is the feature structure resulting from forward application of that analogy to f1. Then for any feature structure f2' compatible with f2 there exists a feature structure f1' such that f1' results from backward application of the analogy to f2' and f1' is compatible with f1.

In the general case, f1 is the result of applying some analogy to some other feature structure f0. Repeated application of the Lemma proves the existence of another feature structure f0' resulting from the application of this other analogy to f1' etc. In this way, a backward generation path can be constructed as a kind of mirror image of a given forward generation path.

Proof of the Lemma: There are two things to show, i.e. that the analogy in question is applicable to f2', and that f1' is compatible with f1.

If f2' is incompatible with the output specification F2 of the analogy, then there must be some feature F whose value is specified as some constant value in F2 and as another constant value in f2'. But since f2 is the result of analogy application with F2 as output specification, f2 must have the same value for F as F2 has, which contradicts the assumption that f2' is compatible with f2. Consequently, f2 must be compatible with F2.

It remains to show that f1', which is the result of backward application of the analogy to f2', is compatible with f1. This can be done by proving that the values in f1 and f1' for any arbitrary feature F must be compatible.

If the value for F in f1' is unspecified, it is, by definition, compatible with that of f1.

If F obtains its value in f1' by being specified as a constant in F1, then the corresponding feature value in f1 cannot be any other constant, since f1 is compatible with F1. Thus, the F value in f1 is either unspecified or the same constant as in f1', which means that f1' and f1 are compatible with respect to the feature F.

Alternatively, F may get its value in f1' by linkage between F2 and F1. In this case, too, the F value in f1 cannot be incompatible with that of f1'. If the F value in f1 is specified as a constant, this value is propagated to f2 through the linkage. Similarly, the F value in f1' must be identical with that in f2'. The conclusion follows from the assumption that f2 and f2' are compatible.

5. Summary and conclusion

This paper has proved that a modified kind of reversibility can be maintained in analogical morphology with default logic. Reversibility guarantees the existence of simple algorithms for analysis and synthesis, while default logic provides for notational adequacy. These properties of analogical morphology make it a serious candidate for a general framework for describing the morphology of natural languages.

I consider the most important problem remaining to be solved to be the question as to what types of patterns (and pattern elements) are necessary to express generalizations about recurrent formal alternations (like reduplication, vowel harmony etc) in the languages of the world. Another important task is to find criteria for ranking competing analogical descriptions of a given language.

Other problems concern the structure of feature systems. Sometimes it would be more appropriate to say that a certain feature is irrelevant than that it is unspecified. How should this be represented in analogical morphology? A related problem is how redundancy in systems of feature values could be exploited.

Practical use of systems of analogies leads to the problem how they could be optimized by compilation and other methods.

Finally, it is a tempting project to explore the potential of analogical morphology as a learning model for morphology.

Acknowledgement

I am indebted to Eva Ejerhed for valuable comments on this study.

References

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.

Koskenniemi, Kimmo. 1983. *Two-level morphology: A general computational model for word-form recognition and production.* Publication No. 11, Department of General Linguistics, University of Helsinki.