MORPHIX
A Fast Realization of a Classification–Based Approach to Morphology

Wolfgang Finkler and Günter Neumann
University of Saarbrücken
D–6600 Saarbrücken 11
Germany
fine@fb10vax.informatik.uni-saarland.dbp.de

1988

Abstract

This paper presents an alternative approach to the use of the Finite State Automata for morphology in inflectional languages. The essential feature is the use of the morphological regularities of these languages to define a fine–grained word–class–specific subclassification. Morphological analysis and generation can be performed at the level of this classification by means of simple operations on n–ary trees. This approach has been implemented in the package MORPHIX which handles all inflectional phenomena of the German language. In spite of the complexity of the German inflection, the average time required to analyse a word is between 0.01 and 0.02 cpu–seconds.

1 Introduction

In order to develop a package for the morphological analysis and generation, it is necessary to consider requirements of linguistics and computer science.

Important properties for the integration of a package into natural language systems are: portability, the possibility of extension, and efficiency As a precondition for a useful integration as a package a wide range of morphological phenomena has to be dealt with. During analysis and generation, linguistical methods should be used in the same manner to structure and to process the morphological characteristics.

Based on the generative phonology there have been several approaches to the treatment of morphological phenomena in inflectional languages by means of Finite State Automata (FSA). FSA describe the morphological regularities of transformations from stems to surface forms and vice versa. The transitions of the automata are directed by testing characters [Kos83] or character strings[Kay82] of the surface form and the underlying lexical representation. Koskenniemi demonstrated that this approach is capable of both morphological analysis and production of word forms for a wide class of inflectional languages [Kos84]. Kay showed how a two–level account might even be given for nonconcatenative morphological phenomena [Kay87].

The starting point of our classification–based approach[1] is to consider morphological regularities as the basis for the definition of a fine–grained word–class–specific classification. Besides morphosyntactic features, phonological phenomena e.g. gemination of a consonant are also used in refining the classification. This approach is applicable to all languages where such regularities can be identified. Classification–based systems for German morphology already exist but with the restriction that they either handle only morphological analysis e.g. [Ber83], [Son80] or generation e.g. [Bus83], or that they consider only single categories [Sch72]. An approach which is able to handle both morphological analysis and generation is described in [TD85] In contrast to the classification used there our hierarchical classification serves to structure two central knowledge sources of our model: the stem lexicon and the inflectional allomorph lexicon (IAL). The stem lexicon contains information about the classification of each stem. The IAL relates each inflectional morph to all its possible combinations of morphosyntactic information (the IAL contains 131 entries). In order to be able to disambiguate the morphosyntactic information of an inflectional morph, the various possibilities

[1]The work presented here is being supported by the German Science Foundation (DFG) in its Special Collaborative Program on AI and Knowledge–Based Systems (SFB 314), project N1 (XTRA).

are differentiated with the help of the classification. Each entry in the IAL is an n–ary tree, the nodes of which describe the classes and the leaves of which contain the appropriate inflectional information (examples of such trees: see figure 1). During the morphological analysis of word forms, the classes which are found in the lexicon direct the traversal of the n–ary tree associated to the candidate inflectional morph which was segmented out of the word form. The fundamental data structures used in our approach to verify or to reject previous segmentations of a word form are the n–ary trees in the IAL. In computer science, there is an abundance of simple and efficient algorithms to operate on trees, thus allowing our model to be implemented efficiently.

The basis of both the Finite State Automata approach and the classification–based approach are the morphological regularities of inflectional languages.

In the Finite State Automata approach, rules are set up and compiled into automata. During morphological analysis the transitions of the Finite State Automata are directed by the actual word form. The determination of which morphological phenomena are present is made in the course of the transitions.

In our classification–based approach, the regularities are used to define classes. Words with the same morphological phenomena are grouped together in classes. Since the classes are contained in the lexicon entries of stems, the morphological analysis entails only simple operations:
– lexicon access to extract classification information
– traversal of an n–ary tree which is found under the segmented inflectional allomorph.

The storage of classes in the lexicon entries reduces the processing–time of the morphological analysis compared to the Finite State Automata approach, where the membership to the class has to be tested directly on each word. Therefore it is appropriate to use a classification–based approach in a package for morphology.

2 The Package MORPHIX

Our classification–based approach of morphological computation has been realized in MORPHIX, a morphological package for natural language systems (NLS's) with German as the input language. A wide range of word classes has been covered so that MORPHIX can be integrated into NLS's.

In designing our package, we opted for a word–level analysis, meaning that MORPHIX has to compute all readings of a given word form. The disambiguation of the results can be achieved by components of the NLS which have syntactic and semantic knowledge at their disposal.

2.1 Inflecting vs. Non–inflecting Word Classes

In inflectional languages, a division of word classes is possible into those which are not capable of inflection – e.g. prepositions, conjunctions, adverbs – and those which are – e.g. nouns, verbs and adjectives.

These linguistic facts are modelled in MORPHIX. Word forms belonging to non–inflecting word classes are stored in a word form lexicon. For such forms only the lexical access is necessary without trying to decompose the word form into candidate stem and inflectional morph pairs. For example, without regarding these facts, one might try to split off the possible inflectional suffix *er* from the German adverb *immer* ('always'), although it is a non–inflecting word. As the non–inflecting word classes are closed and domain independent, the word form lexicon belongs to the fixed lexical knowledge of the package regardless of the specific NLS.

2.2 Segmentation of Inflecting Word Classes

The first step in analyzing forms which are not found in the word form lexicon or which are marked as homographs is to decompose them into candidate stems and corresponding prefixes and suffixes. This process is highly language–specific since at least the inflectional morpheme inventory varies from one language to another. With the information stored in the IAL, MORPHIX first separates the longest possible inflectional suffix from the input word. Because all possible stems for a word form have to be found, it is necessary to split a complex suffix into its parts and to concatenate the stem with these parts step by step. The information about the parts of complex suffixes is contained in the IAL beside the n–ary trees for each inflectional allomorph.

In analogy, a corresponding process is done to split off all possible prefixes to handle prefix/suffix combinations.

For example, for the input word *rasten*, the process of segmentation yields as result no possible prefix and the following possibilities for stems and suf-

fixes:
ra – sten, ras – ten, rast – en, raste – n, rasten – $\emptyset$
Further analysis will show that the three stems *ras* ('to race'), *rast* ('to rest', 'rest') and *raste* ('notch') will lead to correct results.

2.3 Morphophonological Alternation in the Stem

In languages where irregular or even suppletive stems occur in addition to inflectional morphs to form surface word forms, it is possible that the segmentation process may not yield the canonical stems. In such cases either the corresponding canonical stems can be obtained algorithmically or the irregular stems can be stored in the stem lexicon and related to the canonical ones.

In the German language, for instance, there are nouns which form their plural with an inflectional suffix plus umlaut in the stem. Nouns having this feature can be grouped together [Sch72]. An example of such a noun is the German word *haus* ('house'). As previously described, the segmentation process of *häusern* (dative plural of 'house') leads to the following combinations of stems and suffixes:
häus – ern, häuser – n, häusern – $\emptyset$
The canonical stem *haus* is not contained in any of these candidate stems, so the morphological analysis would fail. Because umlaut often occurs in German and since its reduction is efficient to implement, umlaut will be reduced for all candidate stems in the segmentation process. Therefore, it is sufficient to store only the canonical stems of such nouns in the stem lexicon. In the example, the augmented segmentation process thus obtains the following additional combinations:
haus – ern, hauser – n, hausern – $\emptyset$

Changes in the stems of irregular verbs are more difficult to handle than the umlaut phenomenon in nouns. Particular verbs can form their irregular stems by umlaut or ablaut alternations. For example, the German verb *fahren* ('to drive') takes four different stems: the canonical stem *fahr* and the irregular stems *fähr*, *fuhr* and *führ*. In MORPHIX, the irregular stems are stored in the stem lexicon with a pointer to their canonical stem. This method is more efficient than the reduction of irregular stems to their canonical stems algorithmically during segmentation, especially since the number of irregular verbs is limited in German (about 200).

2.4 The Use of the Classification

In the further analysis of the word form thus segmented, MORPHIX determines which of the proposed combinations of stem and affixes describe correct segmentations of the word form. A segmentation is correct if the stem can be found in the lexicon and if the inflectional morpheme in conjunction with the stem represents a grammatical well–formed word form. The algorithm that accepts or rejects each previously derived segmentation is as follows:

- Try to access the stem in the stem lexicon.
- If an entry exists, extract the classification information and construct search–paths containing the classes, otherwise reject this segmentation.
- Extract the n–ary tree from the IAL and match the search–paths within the n–ary tree.
- For each search–path test: If both, the matching succeeds and the next subtree in the n–ary tree is a leaf, then the segmentation was correct. If no leaf could be reached, the combination of stem and inflectional morph was not correct.
- For successful search–paths, collect the stem of this segmentation and the morphosyntactic information that is found in the leaf as a part–result of the algorithm.

Before giving a detailed example, we explain how the German verbs are classified in MORPHIX. First, we consider **morphosyntactical** features. In section 2.3, we mentioned that several irregular stems may be needed to produce all forms of a verb. The irregular stems are grouped into four different classes according to [Bus83]. These classes express the use of a stem within its paradigm, for example that a particular non–canonical stem is used to form the past tense. Disjoint verb types are defined by the combination of these classes. MORPHIX makes use of 11 verb types which express different degrees of irregularity. To illustrate this, we explain the classification of the German verb *fahren* ('to drive'). *fahren* uses four different stems to build all its inflectional forms:

verb–stem	used to build forms of	coded in the lexicon as
führ	subjunctive	VGFD
fuhr	past indicative	VGFC
fähr	2nd/3rd singular present indicative	VGFB
fahr	else	VGFA

Verbs which use four stems in this manner for their inflectional forms belong to verb type 6 in our classification. Most verbs inflect using only the canonical stem, the so–called weak verbs classified as verb type 1.

So far the classification has only considered morphosyntactic features. However, it is possible to use **morphophonological** features to refine the existing classification. In inflectional languages, phonological requirements often lead to morphophonological alternation in the stem coda or inflectional suffix. In German, these alternations consist in the insertion or elision of segments at the juncture between stem and suffix. It can be determined from the nature of the stem coda whether and which alternations occur. In MORPHIX therefore, the irregular stem classes are partitioned into disjoint subclasses.

The subclass A2 of canonical stems (coded as VGFA) may serve here as example:

- It contains all elements that use the canonical stem to form the present 2nd sg and whose stem coda is either "d" or "t". To build the present 2nd sg forms of verbs of this subclass an "e" is inserted between the stem and the regular suffix *st* – e.g. *du arbeit–e–st* ('you work').

A consequence of this refined classificaton is the requirement of the IAL that in the n–ary tree for the suffix *st* no path containing the class A2 as a node is allowed to reach a leaf containing the morphosyntactic information *present 2 sg*. If such a path existed then wrong forms such as *arbeitst* would be analysed.

Example

- input word: *rasten*

- successful lexicon access for the stems (cf. 2.2):
  ra : ((WORTART nomen) (GENUS mas) (SG 0))
  ras : ((WORTART verb) (VTYP 1) (VGFA A4))
  rast : ((WORTART verb) (VTYP 1) (VGFA A2))
  rast : ((WORTART nomen) (GENUS fem) (SG 0) (PL 4))
  raste : ((WORTART nomen) (GENUS fem) (SG 0) (PL 3))

- build the search–paths:

$$
\left.\begin{array}{l} verb--vtype--1--VGFA--A4 \end{array}\right\} \quad for \quad ras
$$

$$
\left.\begin{array}{l} verb--vtype--1--VGFA--A2 \\ nomen--sg--0 \\ nomen--pl--4 \end{array}\right\} \quad for \quad rast
$$

$$
\left.\begin{array}{l} nomen--sg--0 \\ nomen--pl--3 \end{array}\right\} \quad for \quad raste
$$

- extract the n–ary trees for the suffixes *ten*, *en* and *n* from the IAL:
  e.g.

Figure 1: the n–ary tree for the suffix *en*

- result for the morphological analysis of *rasten*:
  (*ras* (WORTART verb) (FLEXION ((imperfekt ((pl (1 3)))) (konjunktiv-2 ((pl (1 3))))
  (*rast* (WORTART nomen) (FLEXION ((fem ((pl (nom gen dat akk)))))))
  (*rast* (WORTART verb)
  (FLEXION ((infinitiv) (praesens ((pl (1 3)))) (konjunktiv-1 ((pl (1 3)))))))
  (*raste* (WORTART nomen) (FLEXION ((fem ((pl (nom gen dat akk)))))))

2.5 Generation of Word Forms

It is possible to use our approach in the opposite direction, i.e. to produce word forms (see [FN88] for the use of MORPHIX in the how-to-say component of a natural language generation system). This requires that the canonical stems in the stem lexicon have additional pointers to their irregular stems. The input structures for the generation process are a canonical stem and the morphosyntactic information for the desired surface form. The algorithm for this task is as follows:

- Try to access the canonical stem in the stem lexicon.
- If an entry exists, extract the classification information, otherwise the stem lexicon is insufficient and a clarification dialog can be started to augment the lexicon.
- Compute the correct stem for the desired inflection with the help of the lexicon entry and construct a search–path for the IAL. The morphosyntactic description from the input constitutes the last element of the search–path.
- This search–path is matched against all n–ary trees of the IAL which are relevant for the desired category.
- The root of the unambiguous entry of the IAL which matches the search–path is the correct inflectional morph and appended to the computed stem.

2.6 The Clarification Dialog

The user is able to augment the MORPHIX lexicon interactively by means of a clarification dialog. Because it is easy to compile a domain–specific lexicon, the integration of the package into natural language systems with different domains is thus facilitated.

The clarification dialog guarantees that only syntactically well–formed lexicon entries are constructed, thus requiring no user knowledge about the internal structure of the lexicon. Only linguistic knowledge is required of the user in the clarification dialog – for example, whether an adjective gets an umlaut in its comparative form – and no knowledge about the subclassification of word classes is necessary – for example which classes handle adjectives with this feature. In addition, existing lexicon entries are used and computations are made to relate stems to their subclassification, for example to check the stem coda. Hence, the user need not carry out the entire classification of an entry by hand.

3 Implementation

The prototype of MORPHIX was developed in Franz–Lisp version 38.79 on a VAX–11/780 under UNIX 4.2. BSD. This prototype has been ported to Interlisp–D running on a XEROX 1108, and has also been ported to Zetalisp and Common Lisp. It runs on SYMBOLICS and TI EXPLORER lisp–machines and on several workstations such as SUN 3, MicroVax II and HP 9000/350.

Run time evaluation was carried out for these environments with a stem lexicon of about 6000 entries. The average cpu–time required for the analysis of one input word is listed below:

Computer	Lisp–Dialect	CPU–Seconds/Word
XEROX 1108	Interlisp–D	0.040
SUN 3	Common Lisp	0.024
TI EXPLORER I	Zetalisp	0.019
VAX – 8700	Franz Lisp	0.016
SYMBOLICS 3640	Zetalisp	0.014
HP 9000/350	Common Lisp	0.009
VAX – 8700	Kyoto Common Lisp	0.005

These run–time evaluations were performed for the prototype of MORPHIX [FN86] that didn't handle all aspects of the classification–based approach. Now we have implemented the approach described in this paper in a new version of MORPHIX. Although, this version has increased the linguistic power of the prototype its average run–time per word is 0.02 cpu–seconds on a SYMBOLICS 3640.

4 Example of an Application

MORPHIX has been integrated into the syntactic component SB–PATR which is part of XTRA (eXpert TRAnslator), a natural language access system for expert systems [XTR87]. SB–PATR is a reimplementation of the D–PATR system, a development environment for unification–based grammars [Kar86]. With the PATR formalism, there has been developed XTRAGRAM a unification grammar for German sentences [Har86]. To integrate MORPHIX into SB–PATR, only the word classes and their inflectional information from the MORPHIX output had to be mapped onto the categories of XTRAGRAM (see figure 2 for the context–free part of a parsed sentence).

The integration of the package MORPHIX into SB–PATR resulted in the following overall improvements:

- During the parse of an input sentence, it is possible to insert entries for unknown words into the stem lexicon with the help of the clarification dialog. The parsing algorithm is just interrupted and then resumed after the clarification dialog is finished. Without MORPHIX, SB–PATR was not able to obtain a result for input sentences containing unknown words.

- The installation of MORPHIX with its stem lexicon of about 6000 entries enables XTRAGRAM to use approximately 60000 surface forms instead of storing all these forms in a word form lexicon, which would have been necessary without the integration of MORPHIX.

These facts and the excellent run time behavior of MORPHIX illustrate the power that is made available to a natural language system by such a package.

5 Conclusion

Due to the very abstract level of description for morphological phenomena used in our approach, the course of morphological analysis is easy to understand. The use of a hierarchical classification can be represented directly by means of n–ary trees. For this reason the testing of performed segmentations can be done with simple algorithms in order to traverse these trees. The realization of MORPHIX shows that our approach can be implemented very efficiently as well. Although morphology has often been studied and handled in computer programs, the demand for MORPHIX - more than 10 external institutes - demonstrates the necessity of such a portable package.

References

[Ber83] H. Bergmann. Lemmatisierung in ham-ans. Memo ANS-10, Fachbereich Informatik, Universität Hamburg, 1983.

[Bus83] S. Busemann. Oberflächentransformationen bei der automatischen generierung geschriebener deutscher sprache - entwurf und implementierung des modularen und anpassbaren systems sutra. Master's thesis, Fachbereich Informatik, Universität Hamburg, 1983.

[FN86] W. Finkler and G. Neumann. Morphix - ein hochportabler lemmatisierungsmodul für das deutsche. Memo 8, Fachbereich Informatik, Universität des Saarlandes, 7 1986.

[FN88] W. Finkler and G. Neumann. Popel-how: Eine komponente zur parallelen, inkrementellen generierung natürlichsprachlicher sätze aus konzeptuellen einheiten. Technical report, SFB 314, Fachbereich Informatik, Universität des Saarlandes, 1988.

[Har86] K. Harbusch. A first snapshot of xtragram, a unification grammar based on patr. Memo 14, SFB 314, Fachbereich Informatik, Universität des Saarlandes, 12 1986.

[Kar86] L. Karttunen. D-patr. a development environment for unification-based grammars. Technical report, SRI International and Center for the Study of Language and Information, Stanford, 1986.

[Kay82] M. Kay. When meta-rules are not meta-rules. In Sparck-Jones & Wilks, editor, *Automatic natural language processing.* University of Essex, Cognitive Studies Centre, 1982. CSM-10.

[Kay87] M. Kay. Nonconcatenative finite-state morphology. In *3rd Conference of the European Chapter of the ACL*, pages 2–10, University of Copenhagen, Copenhagen, Denmark, 1987.

[Kos83] K. Koskenniemi. Two-level model for morphological analysis. In *IJCAI 1983*, pages 683–685, Karlsruhe, 1983.

[Kos84] K. Koskenniemi. A general computational model for word-form recognition and production. In *COLING 1984*, pages 178–181, Stanford University, California, 1984.

[Sch72] G. Schott. Automatic analysis of inflectional morphems in german nouns. *Acta Informatica*, 1:360–374, 1972.

[Son80] Sonderforschungsbereich 100, editor. *SALEM: Ein Verfahren zur automatischen Lemmatisierung deutscher Texte*. Niemeyer, Tübingen, 1980.

[TD85] H. Trost and G. Dorffner. A system for morphological analysis and synthesis of german texts. In D. Hainline, editor, *Foreign Language CAI*. Croom Helm, London, 1985.

[XTR87] Künstliche intelligenz - wissensbasierte systeme, Frühjahr 1987. SFB - 314 Arbeits- und Ergebnisbericht für die Jahre 1985 - 1986 - 3/1987.

Figure 2: context–free part of the sentence: *Die Kinder rasten in der Schule.*