TWO-LEVEL MODEL FOR MORPHOLOGICAL ANALYSIS

Kimmo Koskenniemi
University of Helsinki
Department of General Linguistics
Hallituskatu 11-13, SF-00100 Helsinki 10
Finland

ABSTRACT

This paper presents a new linguistic, computationally implemented model for morphological analysis and synthesis. It is general in the sense that the same language independent algorithm and the same computer program can operate on a wide range of languages, including highly inflected ones such as Finnish, Russian or Sanskrit. The new model is unrestricted in scope and it is capable of handling the whole language system as well as ordinary running text. A full description for Finnish has been completed and tested, and the entries in the Dictionary of Modem Standard Finnish have been converted into a format compatible with it.

The model is based on a lexicon that defines the word roots, inflectional morphemes and certain nonphonological alternation patterns, and on a set of parallel rules that define phonologically oriented phenomena. The rules are implemented as parallel finite state automata, and the same description can be run both in the producing and in the analyzing direction.

I INTRODUCTION

There have been few, if any, morphological parsers that would be truly language independent or even applicable to a wide class of languages with nontrivial inflection. The formalism of generative phonology is powerful enough to describe almost any language. Nevertheless, it has been very difficult to implement computationally. Martin Kay and Ron Kaplan (1981) have recently worked on a model where rules of generative phonology are compiled into finite automata, but until now their system has worked only in the producing mode for testing the descriptions. The ultimate size of their total analyzing automaton is still unknown.

The two-level model has been developed in the course of a project on the computer analysis of Finnish, sponsored by the Academy of Finland and directed by professor Fred Karlsson. The new model is an alternative to the formalism of generative phonology, it has been inspired both by computational aspects and by those trends in linguistics that strive for more concrete and psychologically real phonological models. Even in the study of syntax there is a wide interest in simpler parsing mechanisms that would be more feasible as models of human language processing, e.g. Gazdar's context free grammars without transformations as well as some attempts going even further to finite state techniques (K. Church and E. Ejerhed 1982).

The two-level model differs from generative phonology in that it proposes parallel rules instead of successive ones. In this way it avoids the existence of intermediate stages in the derivation of single word forms. The name "two-level model" reflects the setup, where only the lexical and the surface levels ever "exist", there are no intermediate levels even logically. The very problematic rule ordering is also avoided in the two-level model. The two-level model is attractive as a process model, because it is based on finite state automata, which are the simplest machinery possible. They can be realized with many kinds of networks and devices.

II TOE LEXICON

The lexicon contains just one entry for each word even though the stem is subject to various alternations in the inflection. This is accomplished by two mechanisms. First, morphophonemes may be used in the lexical representations with corresponding rules that govern their realization on the surface. In Finnish, there are several suppletive stem alternation patterns, which have their historical origins, but which are synchronically active only as whole patterns, rather than as a result of any active individual independent rules. As the second mechanism, the lexicon contains one alternation pattern for each such type, and this is referred to in the entries of the corresponding inflectional type. An example of

such an entry for a root would be:

(1) hevo nen/S "Horse";

Here the first item is the phonological representation of the stem, and the last item is the information stored for the lexeme, in this case the English translation. The second item indicates what must come after this entry. In this case, it is the name of an alternation pattern:

(2) nen/S nen SO
sE S123

Here too, the first items nen and sE are the phonological representations, (the capital E is a morphophoneme, which is realized as null before plural i). The second items (SO, S123) refer to subsets of inflectional endings. The root entry together with this pattern defines the various stems hevonen, hevosen, hevosia, etc. Such mini-lexicons have previously been used by Lauri Karttunen (1981) in his TEXFIN-system for analyzing Finnish word forms.

III THE RULES

The essential contribution of the two-level model is the concept of parallel two-level rules that relate the phonological representation defined by the dictionary and the surface form to each other. The rules do not rewrite or process forms, instead, each rule is like an equation that a given surface form and a given lexical representation either satisfy or do not satisfy. Rules are easiest to conceptualize if we assume both levels to be present. Let us take as an example Finnish plural i, which is realized as j if it occurs between vowels. The rule is formulated as:

(3) $i \overset{j}{<=>} V + \_\_ V$

Here the plus sign is a boundary signal between the stem and the inflectional endings. It is used e.g. for indicating plural i:s and similar phenomena. The rule states that i on the lexical level may only correspond to a j on surface if it is preceded by a vowel (on both levels) and the boundary, and followed by a vowel. The rule also says that this is the only possible realization of i in this environment, and furthermore that this is the only environment for this correspondence. In analysis (resp. in production) all rules together act like simultaneous equations. We know the surface (resp. the lexical) representation and find the other as a solution of the equations. Inflectional morphology is quite complicated in Finnish, and the description contains about 50 rules.

IV RULES AS FINITE STATE AUTOMATA

Two-level rules correspond to and are implemented as finite state automata, where the input units are symbol pairs, one symbol from the lexical level and the other (or zero) from the surface level. The automaton corresponding to rule (3) is:

	V	+	i	i	=	=
	V	0	i	j	0	=
1:	2	1	2	0	1	1
2:	2	3	2	0	2	1
3:	2	1	4	5	3	1
4.	0	4	0	0	4	1
5.	2	5	2	0	5	0

The numbered rows stand for the states 1 colon and nonfinal with a period. The column labels consist of character pairs. Zero symbolizes the null (i.e. absence of a character), V stands for any vowel and the equal sign for any character. Sets refer to pairs that are not more explicitly mentioned in another column in the same automaton. Thus the first column does not cover i:s corresponding to i:s or j:s, and the last column does not cover vowels corresponding to vowels. State 1 is always the initial state, and numbers in the table denote state transitions. A zero transition indicates a forbidden configuration. Below is a demonstration of the procession of the automaton in a configuration:

(5) Lexical:	t	a	1	"o	+	i	A	
Surface:	t	a	1	o	0	j	a	
State:	1	1	2	1	2	3	5	2

The other alternative, taloia, would have failed, because the transition on column 1 in state 4 is zero.

The rules (the automata) work together in parallell, a configuration is accepted if all rules (automata) pass. One contradicting rule is enough to ruin the correspondence. The columns with set symbols get their exact meaning only after all rules are given. The model presents a method for synchronizing the rules by collecting all explicit correspondences and aligning the automata to operate coherently. The set of rules (automata) act as a filter in the analysis, when matching entries are sought from the lexicon. In this way nomographic word forms also get all grammatically correct interpretations.

The correspondence between the rule-like formalism and the automata is so close that a compiler is planned for translating rules into automata. However,

the automata are fairly easy to write and understand as such, and the rules in the full description of Finnish inflection were written directly as automata. Some twenty automata were needed and their manual compilation and testing took only a few weeks.

Below are a few examples of two-level analyses of Finnish word forms. The first line of each example is the word form to be analyzed, the second is the sequence of lexical entries that have been matched according to the rules, and the third line gives the information in the entries.

(6)
```
katolla
katTo$HA
Roof Subst ADE SG
(='on the roof')
```

In (6) the T in the lexical form is represented as null on the surface, because of a morpholexical trigger $ in the ending. Capital A in the ending realizes either as a or as a according to vowel harmony.

(7)
```
hakatuimmassa
hakkast*SZTUS+imPAS+issA
Hit Verb PCP2 PSS SUP INE PL
(='those that have
been most beaten')
```

Example (7) is quite complicated as it contains three occurrences of gradation and vowel harmony, and the match consists of five lexical entries: one root, one alternation pattern and three endings.

V COMPUTER IMPLEMENTATION

The two-level program was written in standard Pascal programming language, initially on a Burroughs B7800, but it runs now on DEC-20 as well. It could probably also be run on microcomputers for test purposes with a small lexicon. The program can alternate between producing word forms and analyzing them. Production then starts from the morphophonological representations of the lexical entries and endings, and a valid surface form (according to all rules) is generated.

The 70 000 entries in the Dictionary of Modern Standard Finnish have been transformed into the format of the two-level description. By adding derivational rules and excluding redundant and obsolete entries the whole active lexicon (about 15000 entries) could be simultaneously used by the program. We have tested so far with sections of the lexicon at a time, e.g. entries beginning with k or r. The analysis procedes with a restricted number of steps between each input character. With a large lexicon it takes about 0.1 CPU seconds to analyze a reasonably complicated word form.

It is worth noting that the two-level algorithm provides a language independent framework for dealing with word inflection in several applications. E.g. in information retrieval it would provide means for improving the accuracy of the queries and for reducing the size of inverted files, if the inflected word forms would be replaced by their base forms. As a byproduct the algorithm also contributes to the general solution of spelling correction, by locating invalid word forms.

VI FUTURE

We have plans for demonstrating the validity of the program and the model by creating descriptions for languages of other types, e.g. for some Slavic languages and perhaps some Oriental languages. As the next step in our project we shall investigate more general syntactic models that could be applied to loose word order languages. In highly inflected languages like Finnish, complete morphological analysis is much more important and it covers a significant portion of what is treated as syntax in less inflected languages like English.

ACKNOWLEDGEMENTS

This work has been sponsored by the Academy of Finland and the Cultural Foundation of Finland. My sincere thanks are due to Lauri Karttunen and Martin Kay and to my instructor Fred Karlsson.

REFERENCES

[I] Eva Ejerhed and Kenneth Church, "Finite state parsing." In Papers from the 7th Scandinavian Conference of Linguistics. Helsinki: University of Helsinki, Department of General Linguistics, 1983.

[2] Ronald Kaplan and Martin Kay, "Phonological rules and Finite-State Transducers." Paper at the Annual Meeting of the ACL on 28 December 1981 in New York City.

[3] Lauri Karttunen, "Morphological Analysis of Finnish". (Ibid.)