Morphologically based automatic phonetic transcription

by K. Wothke

A system is described that automatically generates phonetic transcriptions for German orthographic words. The entire generative process consists of two main steps. In the first step, the system segments the words into their morphs, or prefixes, stems, and suffixes. This segmentation is very important for the transcription of German words, because the pronunciation of the letters depends also on their morphological environment. In the second step, the system transcribes the morphologically segmented words. Several transcriptions can be generated per word, thus permitting the system to take pronunciation variants into account. This feature results from the application area of the system, which is the provision of phonetic reference units for an automatic large-vocabulary speech recognition system. Statistical evaluations show that the transcription system has an excellent linguistic performance: more than 99 percent of the segmented words obtain a correct segmentation in the first step, and more than 98 percent of the words receive a correct phonetic transcription in the second step.

A large-vocabulary speech recognition system for German is being developed by the IBM Heidelberg Scientific Center.[1] The methodological and the technical basis of the German system is the English speech recognition system TANGORA, developed at the IBM Thomas J. Watson Research Center in Yorktown Heights, New York.[2] The task of the project is the development and improvement of a voice-activated typewriter that creates the written equivalent of the utterances dictated by the user of the system.

To perform the recognition task, the speech recognition system needs advance knowledge of several types of representations for each word of its reference vocabulary. One important representation is the phonetic transcription of the words. Our speech recognition system uses reference vocabularies of up to 20 000 inflected words per application domain. Therefore, it is a very arduous task to manually provide the phonetic transcriptions. Fortunately, it is possible to develop systems that automatically generate phonetic transcriptions for orthographic words. In the past, such systems were mainly developed as components of text-to-speech synthesis systems, where they supplied intermediate transcriptions of the words to be acoustically synthesized.[3] At the beginning of our project we generated the transcriptions for the speech recognition system with a semiautomatic procedure, which required a high amount of manual and intellectual effort. This paper describes a new system for the automatic generation of phonetic transcriptions, developed at the Heidelberg Scientific Center.[4] Compared with the previous system, the new one has two

crucial innovations that result in a substantial reduction of the formerly high amount of manual work.

First, in the new system, when an orthographic word is phonetically transcribed, its *morphological structure* is taken into account. In German, the pronunciation of a letter is not only dependent on the letter itself and its letter context, but also on its morphological context. By the morphological structure of a word, we refer to the way in which the word is composed of prefixes, stems, and suffixes. Several automatic transcription systems for German also carry out a morphological analysis (see, for example, the systems described by Kommenda,[5] Schnabel and Roth,[6] and Wolf et al.[7]). Additional systems that carry out a morphological analysis for applications other than phonetic transcription are described by Koch et al.[8] and by Thurmair.[9] None of these systems, however, utilizes such an extensive morphological knowledge base as the system presented here. Furthermore, no morphological analysis system is known to us that has such a broad coverage of German morphology and carries out a morphological analysis of German words with such a high degree of correctness as the system presented here.

The second innovation in the new system is that *several phonetic transcriptions are generated for a word* if it has several significantly differing common pronunciations. The transcription components in text-to-speech systems always provide one transcription per word, because a text-to-speech system must synthesize only one pronunciation of a word. A speech recognition system, on the other hand, must adequately react to different common pronunciations of a word. Therefore, a transcription component providing transcriptions for a speech recognition system must generate multiple transcriptions per word, representing at least its most significantly differing common pronunciations. A system that generates multiple transcriptions for an Italian speech recognition system was developed by Scarci and Taraglio.[10] To our knowledge, the system presented here is the first automatic transcription system that generates multiple transcriptions for German words.

In this paper we first define the problem in detail, then we provide a short survey of our former semiautomatic approach to the generation of phonetic transcriptions. The drawbacks of that approach lead us to the new approach with the two innovations mentioned above. The main part of this paper describes the fundamental principles of the new approach, the system architecture, and the linguistic knowledge and its representation. Finally, we present statistical data illustrating the linguistic performance of the system. An appendix is provided that contains examples of morphological analyses and phonetic transcriptions automatically generated by the system.

Throughout the paper phonetic transcriptions are given using the International Phonetic Alphabet, which is defined in detail in Reference 11. A shorter definition of a more actual and revised version of the alphabet is given by Ladefoged in Reference 12.

We conclude this introduction with two fundamental remarks concerning the task of automatically generating phonetic transcriptions.

First, almost every system simulating language-specific human capabilities has limitations in its linguistic correctness and completeness. The same is true for transcription systems: currently each automatic transcription system for German produces at least a small number of incorrect transcriptions.

Second, the feasibility and the complexity of an automatic transcription system vary from language to language. They strongly depend on the reflection of the pronunciation in the orthography of the language. Italian and Spanish are known as languages for which it is relatively easy to develop transcription systems, whereas it is more difficult for English, French, and German.

We ask the reader to consider these final remarks while reading this paper.

The problem

The speech recognition system now being developed in the IBM Heidelberg Scientific Center requires that the phonetic transcriptions are available for each word of its reference vocabulary. The transcriptions are used as reference units during the recognition process.

The concept *word* includes six subclasses:

Table 1 Examples for multiple phonetic transcriptions per word (ordered according to the different subclasses of our concept of a word)

	Orthographic Representation		Phonetic Transcription	English Translation
1.	< Chance >	⇒	['ʃãsə] ['ʃaŋsə]	
	< erblichen >	⇒	['ʔɛrplɪçn̩] [ʔɐ'blɪçn̩]	hereditary paled
2.	< u.a. >	⇒	[ʔʊnt 'ʔandərəs] [ʔʊnt 'ʔandərə] ['ʔʊntɐ 'ʔandərəm] ['ʔʊntɐ 'ʔandɐm]	and other things and others inter alia
	< kWh >	⇒	[ka:ve:ha:] [kɪlo'vatʃʊndə]	kilowatt hour
3.	< 3. >	⇒	['drɪtəns], ['drɪ̂ns] ['drɪtɐ], ['drɪtə] ['drɪtəs], ['drɪtəm] ['drɪ̂m], ['drɪtən] ['drɪ̂n]	different inflections and pronunciation variants of thirdly, third
4.	< - >	⇒	['bɪndəʃtrɪç] ['mi:nʊs]	hyphen minus
5.	< , >	⇒	['kɔma] ['baɐ̯ʃtrɪç]	
6.	< KO-Großschreibung >	⇒	[ko:'gro:sʃra̯ebʊŋ] [ko:'gro:ʃra̯ebʊŋ]	

1. Full orthographic words such as <Chance> (English: chance), <erblichen> (English: hereditary, paled)
2. Abbreviations such as <u.a.> (English: and other things, and others, inter alia) and <kWh> (English: kilowatt hour)
3. Ordinal and cardinal numbers in digit representation (<3.>, <34>, <2,32>, <1.231>, etc.)
4. Special characters (<$>, <%>, <&>, <->, etc.)
5. Punctuation marks (<.>, <,>, <?>, <!>, etc.)
6. Formatting commands such as <KO-Großschreibung>, which triggers the speech recognition system to write the dictated words that follow exclusively in capital letters.

For words coming out of each subclass, the speech recognizer requires transcriptions. Some subclasses require different methods of supplying transcriptions. For a large quantity of words, it will not suffice to provide one transcription. Many words have several significantly differing common pronunciations, as Table 1 illustrates.

Our current prototype of the German speech recognition system, which includes a reference vocabulary of about 12 000 different words, has on average 1.23 transcriptions per word.

We cannot limit the pronunciation variants of a word to those listed in the pronunciation dictionaries for German.[13–15] The description of the pronunciation variants in these dictionaries is too restrictive to be used in a speech recognition system. The decision to be made concerning the pronunciation variants used as reference units should depend on the real pronunciation of German speakers, as long as this pronunciation is not

Table 2 Correspondence rules between the German letter string ⟨ch⟩ and its pronunciations (verbal representation)

	Letter String <ch>	Pronounce	Examples		
			Orthographic Representation	Phonetic Transcription	English Translation
1.	At the beginning of a word and before <l> or <r>	[k]	<Chlor> <Chrom>	[klo:ɐ] [kro:m]	chlorine chromium
2.	At the beginning of a word and *not* before <l> or <r>	[ʃ]	<Chauffeur>	[ʃɔ'fø:ɐ]	chauffeur
3.	Together with preceding <s>	[ʃ]	<Tasche>	['taʃə]	pocket
4.	Before <sl> or (<s> plus one of the derivational suffixes <isch>, <ig>, or <ung>)	[k]	<Drechsler> <sächsisch> <flechsig> <Verwachsung>	['drɛkslɐ] ['zɛksɪʃ] ['flɛksɪç] [fɐ'vaksʊŋ]	turner Saxon, ADJ sinewy deformation
5.	After one of the letters <a>, <o>, or <u>	[x]	<Bach> <Loch> <Tuch>	[bax] [lɔx] [tu:x]	brook hole cloth
6.	Else	[ç]	<Technik> <mich> <Bäche> <Löcher> <Tücher> <Milch> <durch>	['tɛçnɪk] [mɪç] ['bɛçə] ['lœçɐ] ['ty:çɐ] [mɪlç] [dʊrç]	technique me brooks holes cloths milk through

influenced by strong dialectal variations or by speech disorders.

Approaches to the problem solution

The former approach and its inadequacies. In the past we generated the transcriptions with a semi-automatic procedure.

The first *automatic step* originated from the text-to-speech system TETOS.[16] With a rule-based approach, it created one transcription for each full orthographic word. Abbreviations, special characters, and digit strings were first automatically mapped onto their full orthographic equivalents, before being transcribed by means of the rules.

In order to demonstrate how the letter-to-phone rules functioned, we give a short description of the rule formalism and an example of letter-to-phone rules expressed in this formalism. A complete definition of the formalism and a listing of all rules used by the TETOS system can be found in Reference 17.

The general format of the letter-to-phone rules is:

left[*string*]*right* ⇒ *phonetics*

One would read this as: Map *string* on the transcription *phonetics*, if *string* occurs in the orthographic word, which is to be transcribed, within the context *left* and *right*.

Table 3 Correspondence rules between the German letter string ⟨ch⟩ and its pronunciations (representation with rule formalism)

	Letter String		Phonetic Transcription
1.	#[ch]l	⇒	k
	#[ch]r	⇒	k
2.	#[ch]	⇒	ʃ
3.	[sch]	⇒	ʃ
4.	[ch]sl	⇒	k
	[ch]sS	⇒	k
5.	B[ch]	⇒	x
6.	[ch]	⇒	ç

Note: # represents the word boundary

Some important conventions associated with this general format are:

- *String* is a string of letters.
- The contextual conditions of a rule, i.e., *left* and *right*, could contain letter strings as well as names for sets of letter strings. The sets are defined by the rule writer and are designated with single capital letters, whereas all letter strings are lowercase.
- The character <#> is used to represent the word boundary.
- The rules are ordered. This means that, for a given letter string, the rules are applied in their order within the rule set.

A well-known example concerning the correspondence between letters and phones in German is the pronunciation of the letter string <ch>. If we overlook some exceptions, we can formulate the correspondence rules as in Table 2.

If we define the two sets

S = {isch ig ung}

B = {a o u}

we can express these correspondence rules in the rule formalism as shown in Table 3.

The automatic part of our former semiautomatic transcription procedure had two major drawbacks entailing a high amount of revision of the transcriptions in the second *manual step*.

First, only one transcription was created per orthographic word. Thus transcriptions of additional pronunciations had to be added manually.

Second, the former approach neglected one important parameter when mapping letters on phonetic transcriptions. This caused many transcription errors, which had to be identified and corrected manually. The procedure mapped letters on their transcriptions depending upon the letters themselves and their letter context. The emphasized (bold appearing) letters and the transcriptions corresponding to them in the pairs of sample words found in Table 4, show that these two parameters are not the only ones determining the phonetic equivalent of a letter in German. The different transcriptions of the same letter string in these word pairs is a consequence of the fact that the words have different *morphological structures*, as can be seen in the second column of each example. We want to explain this in more detail with the pronunciations of the letter strings <ch> and <sch> in the word pairs <Wach-stube> vs <Wachs-tube> and <Häus-chen> vs <täusch-en> respectively.

Applying the rules of Table 3 to the string <ch> in the two words <Wach-stube> and <Wachs-tube> generates the pronunciation [x] by Rule 5 both times, which is incorrect for <Wachs-tube>. The only way to differentiate between the two correct pronunciations [x] (for <Wach-stube>) and [k] (for <Wachs-tube>) in these two identically written words is to refer to the different morphological structures:

- <ch> after <a>, <o>, or <u> and at the end of a morph—pronounce [x]
- <ch> before <s> followed by a morph boundary—pronounce [k]

When we apply the rules to <sch> in the two words <Häus-chen> and <täusch-en>, Rule 3 generates the pronunciation [ʃ] each time, which is incorrect for <Häus-chen>. There are two ways to differentiate here between the two correct pronunciations [sç] (for <Häus-chen>) and [ʃ] (for <täusch-en>).

Table 4 Examples illustrating the dependency of phonetic transcriptions on the morphological structure of words

	Orthographic Representation	Morphological Structure	Phonetic Transcription	English Translation
1.	< Wa***ch***stube >	< Wa***ch*** stem ***s***tube > stem	['vaxʃtu:bə]	guardroom
	< Wa***chs***tube >	< Wa***chs*** stem tube > stem	['vakstu:bə]	wax tube
2.	< fuß***en***de >	< fuß stem ***en*** suffix d suffix e > suffix	['fu:səndə]	being based on
	< Fuß***e***nde >	< Fuß stem ***e***nde > stem	['fu:s?ɛndə]	foot of a bed
3.	< b***u***cht >	< b***u***ch stem t > suffix	[bu:xt]	he books
	< B***u***cht >	< B***u***cht > stem	[bʊxt]	bay
4.	< ***ver***anlagen >	< ***ver*** prefix ***an*** prefix lag stem en > suffix	[fɐ'?anlɑ:gən]	to assess
	< ***Vera***nda >	< ***Vera***nda > stem	[ve'randa]	veranda
5.	< Häu***sch***en >	< Häu***s*** stem ***ch***en > suffix	['hɔ̈ʏsçən]	small house
	< täu***sch***en >	< täu***sch*** stem en > suffix	['tɔ̈ʏʃn]	to deceive
6.	< Volk***s***park >	< Volk stem ***s*** suffix park > stem	['fɔlkspark]	public park
	< Kalk***s***pat >	< Kalk stem ***s***pat > stem	['kalkʃpa:t]	lime spar

Note: The emphasized (bold and italic) letters in each word pair, athough spelled the same and in the same letter context, have different pronunciations.

One way is to formulate rules with relatively long contextual conditions and to differentiate between the two correct pronunciations by making reference to the sole discriminating feature of the two orthographic words, i.e., their initial letter:

häus[ch] ⇒ ç

[sch] ⇒ ʃ

This method of solving the problem is quite awkward, because the set of letter-to-phone rules becomes gradually confused by many exceptional rules and by long contextual conditions. Furthermore, the method does not consider the real rea-

son for the different pronunciations, which, after all, is the different morph structure of the words.

The second way is to take into account the various morphological structures of the words and formulate rules such as the following:

- Pronounce <sch> as [sç] if <ch> is the beginning of the suffix <chen>
- Else, pronounce <sch> as [ʃ]

The new approach. Disregarding the morphological structure of the words when phonetically transcribing them caused many transcription errors in our former approach. This deficiency, as well as the drawback that only one transcription was generated per word, led us to a new approach. The remarkable innovations of the new approach compared with the former one are:

- It takes into account the morphological structure of words.
- It can generate several transcriptions per word if there exist several significantly differing common pronunciations of the word.

The new approach consists of the two main steps, *morphological segmentation* and *phonetic transcription*.

The first step, morphological segmentation, has to segment the orthographic words into the morphs they consist of, i.e., into prefix, stem, and suffix. The boundary before each morph has to be marked with a special character, which indicates the coarse class of the following morph. In order to mark the boundaries, we use the following characters:

+	before a prefix
=	before a stem
%	before suffixes of German origin
_	before suffixes of Latin or Greek origin
~	before suffixes of French or English origin

Furthermore, the word boundaries are marked with the word boundary symbol <#>, in order to facilitate, in the second step, the application of such letter-to-phone rules that deal with special pronunciations found only at the beginning or at the end of a word.

Some words have several morphological segmentations with different pronunciations. An example is <Wachstube> previously discussed. Therefore, the first step of the new approach is to segment these special words in several ways.

Two major, and five minor, linguistic knowledge sources are necessary for morphological segmentation and for morph boundary marking.

The two major knowledge sources are:

1. A *morph dictionary* containing the morphs and a *fine* classification of the morphs according to syntactically relevant morph classes
2. A *word syntax* describing those sequences of (fine) morph classes that underlie syntactically well-formed words. The word syntax will prevent syntactically ill-formed segmentations, such as the segmentation of <Walzer> (English: waltz) in

   *#=Wal+zer#

   i.e., into the stem <Wal> (English: whale) and the prefix <zer>. This incorrect segmentation can be rejected by means of the syntax, since a prefix may not occur at the end of a word. (An asterisk in front of a word or a segmentation is used above and in the following text to indicate that the word or the segmentation is ungrammatical.)

The five minor knowledge sources are:

1. A morph boundary table. This table contains for each (fine) morph class the symbol that the segmentation procedure has to use when marking a boundary before a morph belonging to this class. For example, the table can fix that +, =, %, _, and ~ have to be used as specified above.

2. A table of forbidden morph classes. To explain the concept of forbidden morph classes, we must make a short digression: German orthography has a special feature that must be taken into account by a morphological segmentation procedure. If two stems are concatenated, where

   - The first stem ends in a vocalic letter and two identical consonantal letters, and
   - The second stem starts with the same consonantal letter and a vocalic letter

then the result of the concatenation does not contain the consonantal letter three times, but only twice. For example, the concatenation of <Krepp> (English: crepe) and <Papier> (English: paper) is not *<Krepppapier>, but <Kreppapier>. To make the segmentation of such words into their original components possible, the segmentation procedure has to check each word to determine whether it contains a double consonant enclosed by vocalic letters. If so, the segmentation procedure not only tries to segment the original word (with two consonants), but also a modified copy of it (with three consonants). Now, each of the three consonants arising from the consonant trebling must belong to a stem; they must not belong to a prefix, or suffix. To avoid incorrect segmentations like that of <Hersteller> (English: producer) into

*#+Her=stell%ler#

where one of the trebled <l> erroneously occurs in the suffix <ler>, we introduce a table of forbidden classes, which contains the names of the prefix and suffix classes, and which must not attract any of the three consonantal letters arising from consonant trebling.

3. A dictionary of exceptional words. By an exceptional word, we refer to a word with a segmentation that requires morph classes and word syntax rules applying to only one or two different words. The integration of such rules into the word syntax is too costly compared to a simple insertion of the one or two exceptional words with their segmentations in this dictionary.

4. A dictionary of abbreviations, special characters, and punctuation marks. The morphological segmentation, by means of the knowledge sources described above, works for full orthographic words. Abbreviations, special characters, and punctuation marks cannot be segmented into morphs by means of a morph dictionary. They must first be mapped onto their full orthographic equivalents, which then can be segmented. The dictionary of abbreviations, etc., contains abbreviations, special characters, and punctuation marks with their full orthographic equivalents.

5. A procedure for the mapping of digit strings on their segmented full orthographic equivalents.

Input to the second step of the new approach, phonetic transcription, is the output of the first step, i.e., orthographic words annotated with morph and word boundary symbols. The second step has to generate the transcriptions for these words. The linguistic knowledge necessary to carry out this process is a set of letter-to-phone rules that must fulfill two conditions in order to take into account the two innovations of the new approach:

1. The rules must be able to consider the morphological structure of the orthographic words, indicated by the morph boundary symbols inserted by the first step.
2. It must be possible to map the same letter string on several alternative transcriptions in order to facilitate the generation of multiple transcriptions per word, which represent pronunciation variants.

This two-step process, consisting of morphological segmentation and subsequent phonetic transcription, is illustrated in Figure 1 with the example word <Wachstuben>. The first step must generate the two segmentations #=Wach=stube%n# (English: guardrooms) and #=Wachs=tube%n# (English: wax tubes). In the second step #=Wach=stube%n# has to be mapped on the transcriptions ['vaxʃtu:bən] and ['vaxʃtu:ɓm], and #=Wachs=tube%n# has to be mapped on ['vakstu:bən] and ['vakstu:ɓm].

Realization of the new approach

Fundamental principles. The new phonetization system requires several knowledge sources for morphological segmentation and for phonetic transcription. As these knowledge sources are extensive and complex, it is not appropriate to integrate them into a program. For the sake of flexibility, we decided to separate linguistic knowledge and software as much as possible, and to encode the knowledge in representation formalisms that are interpreted by the software. This permits development of the linguistic knowledge independently of the software. Updates in the knowledge do not entail any modifications of the software, as long as the formalisms for knowledge representation are left unchanged.

Figure 1 Example illustrating the two-step process of the new approach for automatic phonetic transcription

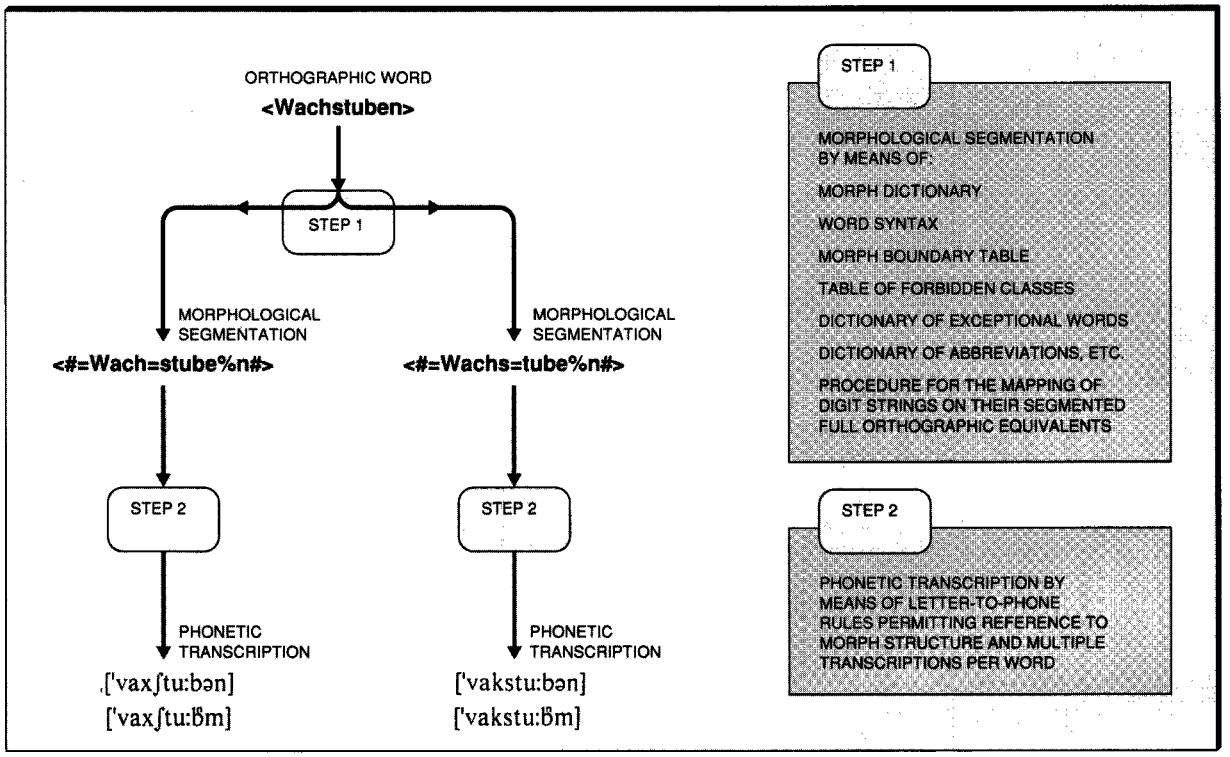
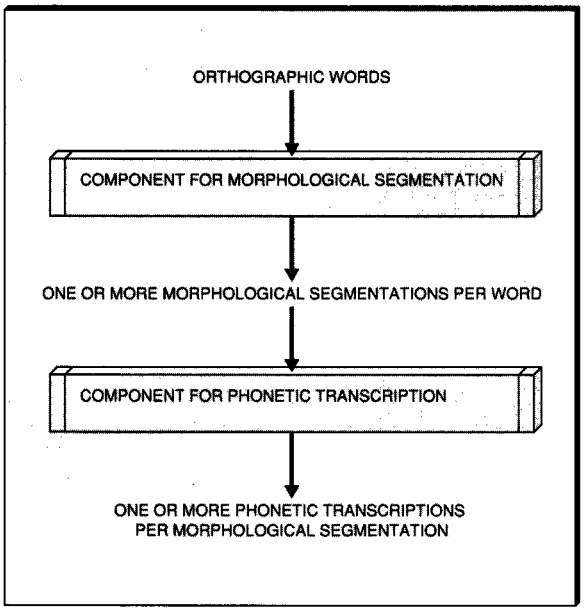


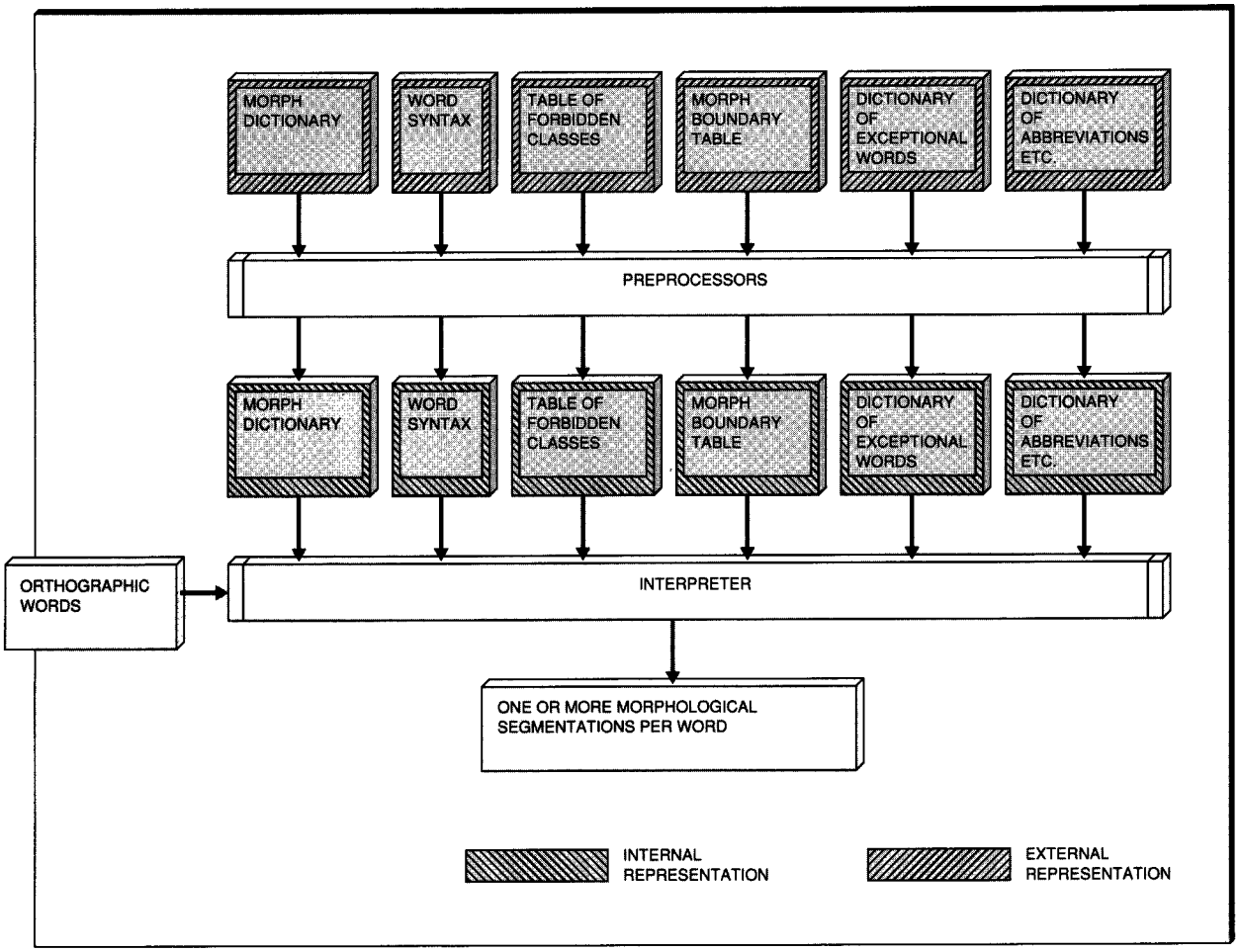
Figure 2 General survey of the system for phonetic transcription



Furthermore, separation of linguistic knowledge and software permits the user to develop a system for another natural language without changing the program code. The only condition that must be met is that the knowledge for morphological segmentation and for phonetic transcription in that language can be expressed in the representation formalisms.

Unfortunately, we were not able to achieve a complete separation of language-specific knowledge and software. The mapping of digit strings on their segmented full orthographic equivalents had to be integrated into the software. The adaptation of this knowledge to a different language requires the replacement of the current German-specific software module for digit processing with an appropriate digit processing module for the new language. The German-specific trebling of double consonants between vowels, which is also integrated into the software, can be suppressed

Figure 3 Architecture of the component for morphological segmentation



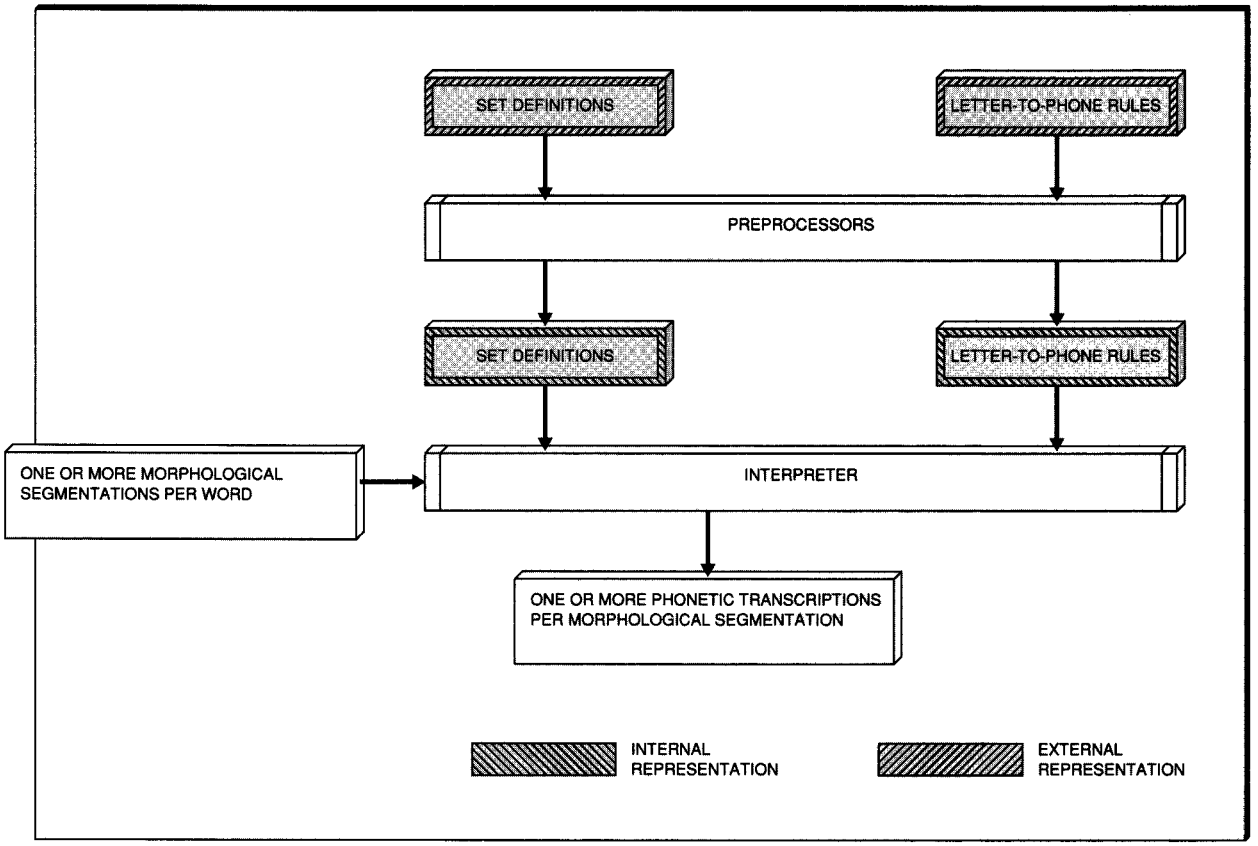
for a different language, without any software modifications, simply by filling the table of forbidden classes with the names of *all* morph classes.

System architecture. In its basic form, the system can be described as consisting of two black boxes (see Figure 2). Input data for the whole system are orthographic words. After the morphological segmentation, one or more morphological segmentations per word occur as an intermediate result. The segmentations are the input to the phonetization component, which generates one or more phonetic transcriptions per morphological segmentation.

The morphological component. Figure 3 illustrates the architecture of the system component for morphological segmentation. As we described in the previous section "The new approach" this component uses linguistic knowledge for the segmentation of orthographic words, as well as a language-specific procedure for the treatment of digits. The linguistic knowledge is distributed over six knowledge sources that exist in two representations:

- An external representation, which is readable by a person and can easily be modified by means of an editor
- An internal representation, which is automatically generated by preprocessors from the external representation and is more suitable for the processing by the interpreter function of the software

Figure 4 Architecture of the component for phonetic transcription



The interpreter uses the internal representation of the linguistic knowledge and segments the input words into morphs. It determines, for each word, *all* segmentations that are possible according to the knowledge.

The algorithmic aspects of the morphological component have been documented in References 18 and 19.

The phonetic component. The architecture of the system component for phonetic transcription is shown in Figure 4. It is very similar to that of the component for morphological segmentation.

The linguistic knowledge concerning the phonetic transcription is stored in two files:

- The first file contains definitions of sets of letter strings and of phone strings. We explain the purpose of these set definitions in a following subsection, "Knowledge for phonetic transcription."
- The second file contains the letter-to-phone rules.

As in the morphological component, the knowledge sources exist in an external representation and in an internal representation. The preprocessors translate the external representation into the corresponding internal representation.

The interpreter reads the internal representation of the linguistic knowledge and transcribes the segmentations provided by the morphological component according to the set definitions and the letter-to-phone rules. For each morphological segmentation it generates at least one phonetic transcription.

The algorithmic aspects of the component for phonetic transcription have been documented in References 20 and 21.

The linguistic knowledge and its representation. At present, the system works with linguistic knowledge for the segmentation and transcription of German words. The knowledge is encoded in formalisms that we partially introduce in the two following discussions. The system can also be fed with linguistic knowledge for different languages. Its suitability then depends on the adequacy of the formalisms for the representation of the knowledge, which is necessary to describe the processes of segmentation and transcription in these languages.

Knowledge for morphological segmentation. In this section we give a survey of the linguistic knowledge currently incorporated into the morphological component and we describe the formalisms for the external representation of the knowledge sources. The contents of some knowledge sources are quite trivial. Therefore, we confine our survey to the two major knowledge sources, the morph dictionary and the word syntax.

The *morph dictionary* contains a large number of morphs and their morph classes. Each dictionary record contains a morph with up to six morph classes. The set of morph classes, used in the dictionary, must be identical with the set of morph classes in the word syntax.

The German morph dictionary currently contains 10 784 morphs. We created this inventory in the following way:

- We started with a list of about 2000 morphs selected from Ortmann.[22]
- This initial list was augmented with about 4000 morphs, which we extracted from a machine-readable morph list, which the Institut für deutsche Sprache in Mannheim (Germany) made available to us.
- Finally, an additional 4000 morphs were inserted, which we obtained by an analysis of the vocabulary in Wahrig.[23]

The average number of classes per morph is 1.87. The entire classification scheme consists of 198 different classes. The classes subclassify the morphs

- Into prefixes, stems, and suffixes
- Into verbal, adjectival, nominal morphs
- According to number, case, tense, mode, degree of comparison, and ablaut

To understand the contents of the *word syntax* file, we first define the formalisms of right linear regular grammars and finite state transition networks.

A great number of formalisms exist for the representation of the syntaxes of natural languages. These formalisms differ from each other in the sets of languages they can describe, and in the complexity, time, and storage requirements of the algorithms, which parse strings according to grammars written in these formalisms.

Many formalisms and the corresponding parsing algorithms that could be used for morphological segmentation have too much overhead, if used for this task. These are, for example, the context-free and context-sensitive grammars of the Chomsky hierarchy.[24] A syntax formalism of the Chomsky hierarchy, which may only be sufficient for morphological segmentation, seems to be the formalism of right linear regular grammars (see Figure 5).

Concerning the suitability of regular grammars for morphological analysis, Hellwig writes in Reference 25:

> Natürliche Sprachen als ganze gehören nicht zu den regulären, die mithilfe eines endlichen Automaten erkannt werden können, möglicherweise jedoch Ausschnitte davon, wie der Silbenbau, die Morphologie und die Wortbildung.
>
> [Parts of natural languages, such as the structure of syllables, the morphology, and the word formation, may be recognized by a finite automaton, or regular grammar.]

It is well-known from the theory of formal languages, that right linear regular grammars describe the same set of languages as finite automata and finite state transition networks. A *finite state transition network* consists of a set of labeled states, connected by directed arcs, each representing a transition between two states and being labeled with a pattern. Some of the states are initial and others are terminal. The parsing of a given string is performed by finding a path through the

Figure 5 Definition of the formalism of right linear regular grammars

A right linear regular grammar G is a quadruple

$G = (N, T, P, S)$

where

- N is the alphabet of the nonterminal symbols.
- T is the alphabet of the terminal symbols.
- $N \cap T = \emptyset$.
- P is a finite set of productions of the form

  $\alpha \Rightarrow \beta\ \gamma$

  where $\alpha \in N$, $\beta \in T$, $\gamma \in N$

  or

  $\alpha \Rightarrow \beta$

  where $\alpha \in N$, $\beta \in T$
- S is the set of start symbols. $S \subset N$.

network from an initial state to a terminal state along a sequence of arcs whose label sequence totally matches the complete given string.

In the following we illustrate with an example how the morphological structure of words can be described with the finite state transition network formalism. We then show how this representation can be translated into a functionally equivalent right linear regular grammar.

Figure 6 shows an exemplary network representing a very simplified description of a small part of German morphology. It describes a word as consisting of an arbitrary number of prefixes followed by at least one stem. The stem(s) may be followed by an arbitrary number of derivational suffixes. The word may end with a inflexional suffix or it may form a compound word by being concatenated to a word of the structure just described, optionally interspersed with a linking suffix.

The rules of the right linear regular grammar describing the same language as a given finite state transition network are derived from the network in the following way: For each state′-arc-state″ transition write down

- The rule

  state′_label ⇒ arc_label state″_label

  if state″ is not a terminal state and not an optional terminal state

- The two rules

  state′_label ⇒ arc_label state″_label

  state′_label ⇒ arc_label

  if state″ is an optional terminal state

- The rule

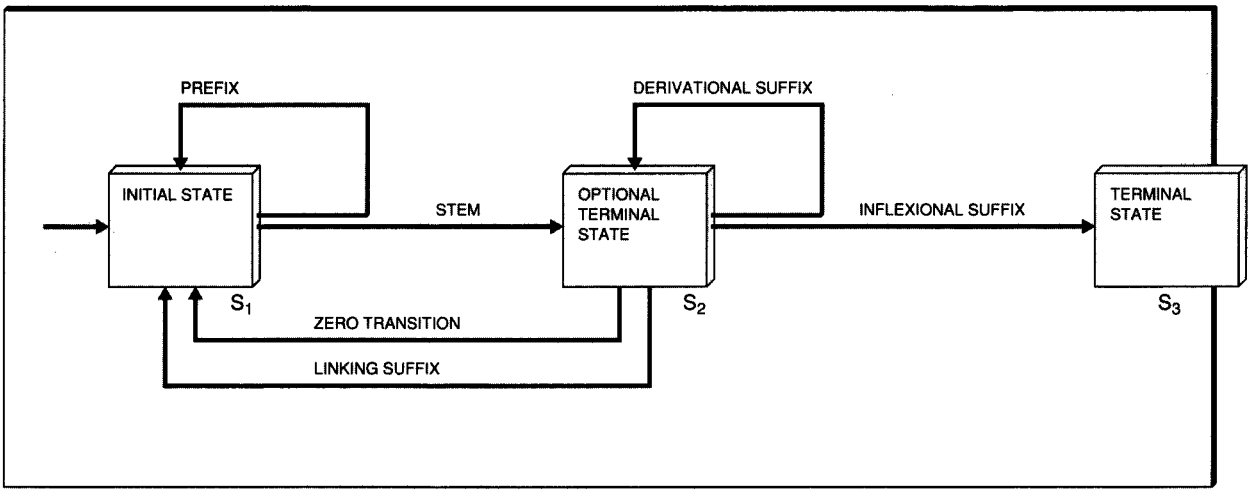
  state′_label ⇒ arc_label

  if state″ is a terminal state

The alphabet N of nonterminal symbols of the grammar consists of the state labels of the network.

The alphabet T of terminal symbols of the grammar consists of the arc labels of the network.

Figure 6 Example of a finite state transition network for a small part of German morphology



The start symbols S of the grammar are the labels of the initial states of the network.

Table 5 shows the right linear regular grammar describing the same language as the sample transition network given in Figure 6.

The word syntax file expected by the preprocessors of the morphological component must contain the rules that describe the sequences of morph classes underlying syntactically well-formed words. The rules must be written in a right linear regular grammar, where the morph classes are the terminal symbols, i.e., the arc labels of the corresponding finite state network. It turned out that for the manual development of the word syntax, the formalism of finite state networks is easier to handle than a right linear regular grammar. So we first represented the syntax with a finite state network, which we finally translated into a functionally equivalent right linear regular grammar, having a format more suitable for machine input. The right linear regular grammar was fed into the automatic segmentation system, where a preprocessor automatically mapped it onto a functionally equivalent nondeterministic finite automaton, which permits a very efficient automatic processing of the word syntax. The interpreter of the morphological component uses the word syntax in the following way. It segments an orthographic input word into morphs found in the morph dictionary. To each morph of the word the interpreter assigns the set of morph classes indicated for it in the dictionary, thus creating a sequence of sets of morph classes associated with the whole word. The syntactic well-formedness of the segmentation is checked by trying to find a path through the whole sequence of sets matching a path from an initial state to a final state through the finite automaton. To avoid an explosion of syntactically ill-formed segmentations (before the syntax is consulted) the syntax checking is nested into the segmentation procedure.

Table 5 Example of a right linear regular grammar for a small part of German morphology

$S_1 \Rightarrow$ prefix S_1
$S_1 \Rightarrow$ stem S_2
$S_1 \Rightarrow$ stem
$S_2 \Rightarrow S_1$
$S_2 \Rightarrow$ linking suffix S_1
$S_2 \Rightarrow$ derivational suffix S_2
$S_2 \Rightarrow$ derivational suffix
$S_2 \Rightarrow$ inflexional suffix

Our German word syntax takes into account the following basic features of German morphology:

- *Composition.* Many words are compound words, i.e., they are concatenations of stems, optionally interspersed with prefixes and suf-

fixes. Example: <Lebensgefahr> (English: danger to life) is the concatenation of <Lebens> (genitive of <Leben>, English: life) and <Gefahr> (English: danger).

- *Derivation.* In addition, numerous words are derivations, i.e., they are concatenations of stems with prefixes or derivational suffixes. Example: <untragbar> (English: unbearable) is the concatenation of prefix <un>, stem <trag>, and derivational suffix <bar> (English: un+bear+able).
- *Inflexion.* The different morphological features of nouns, adjectives, verbs, and pronouns (e.g., person, number, gender, case, tense, mode, degree of comparison) are mainly expressed by inflexional suffixes. Our computations based on a large machine-readable dictionary resulted in an average of 5.2 different inflected forms per lemma.

The current word syntax is a finite automaton with 289 states and 1368 transitions.

A more detailed description of the German specific linguistic knowledge used for morphological segmentation is given in Reference 26.

Knowledge for phonetic transcription. The phonetic component accesses two knowledge sources: set definitions and letter-to-phone rules (as previously mentioned in the section "The phonetic component"). The representation formalisms for these knowledge sources have been influenced to some degree by a formalism introduced by Chomsky and Halle.[27]

The purpose of *set definitions* is to reduce the number of letter-to-phone rules and to make them more clear. We have introduced sets of letter strings and sets of phone strings as the two types of set definitions.

Different letter strings, occurring in the contextual conditions of a sequence of rules, may be united into a *set of letter strings*, if the rules do not differ in any other way. The set is designated with a set identifier. The sequence of rules can then be replaced with one functionally equivalent rule that contains the set identifier, instead of the different letter strings. We want to show this with an example: the letter <b> is usually pronounced [p], if it occurs within a morph left of the letters <t>, <s>, or <k>. This requires three rules:

[b]t ⇒ p e.g.: <Abt> (English: abbot)

[b]s ⇒ p e.g.: <Herbst> (English: autumn), <grabschen> (English: to grab)

[b]k ⇒ p e.g.: <Wiebke> (female first name)

If we define a set /VLCONS/ with the letters usually corresponding to voiceless consonants, we can replace the three rules with

[b]/VLCONS/ ⇒ p

As we see below, the rule formalism offers the possibility of specifying several alternative transcriptions on the right side of a rule. This is very useful if a letter string has different pronunciations. Look, for example, at the letter <z> in <Walze> (English: roller), <Salz> (English: salt), <tanzen> (English: to dance), and <Schwanz> (English: tail). In addition to the standard pronunciation [ts], also [s] is possible. Thus we may have two rules with the same alternative pronunciations on the right side:

l[z] ⇒ ts s

n[z] ⇒ ts s

The concept of *sets of phone strings* permits us to simplify these rules to a small degree. If we define a set

|Z_PHON| = { ts s }

we can write instead

l[z] ⇒ |Z_PHON|

n[z] ⇒ |Z_PHON|

The general external format of the set definitions is as follows:

SET /*id*/ = $\{x_1 \ldots x_i \ldots x_n\}$ (for sets of letter strings)

SET |*id*| = $\{x_1 \ldots x_i \ldots x_m\}$ (for sets of phone strings)

where

- *id* is the set identifier.
- Each x_i is a set element. In a letter set, each x_i represents a letter string. In a phone set, it represents a string of phonetic symbols.

Currently, 31 sets are defined, including such sets as:

- The vocalic letters
- The vocalic letters usually corresponding to back vowels
- The vocalic letters usually corresponding to front vowels
- The vocalic letter strings usually corresponding to diphthongs
- The consonantal letters
- The consonantal letter strings usually corresponding to voiceless consonants
- The consonantal letter strings usually corresponding to plosives
- All morph boundary symbols

The *letter-to-phone rules* have the general external format

left[*string*]*right* ⇒ $phonetics_1 \ldots phonetics_n$

One would read the above expression as: Map *string* on the transcriptions $phonetics_1, \ldots, phonetics_n$, if *string* occurs within the context *left* and *right*.

There are several conventions to be observed:

- *String* is a string of letters and/or morph boundary symbols. It must not be empty.
- *Left* and *right* each consists of up to five units. One of these units may be a string of letters or morph boundary symbols, or both. The other units must be identifiers of letter sets. At the beginning of *left* or at the end of *right*, or both, may occur the word boundary symbol <#>. *Left* and *right* may be empty.
- An identifier of a letter set may be followed by the restricted Kleene operator. The format of this operator is

  *n

  where $1 \leq n \leq 5$. The Kleene operator implies that elements of the preceding set may occur up to n-times, but may also be absent. For instance, if you want to describe the pronunciation of the string <ern> in stems like <Stern> (English: star), <fern> (English: far), or <Ernte> (English: harvest), you could use instead of the three rules

  =/CONS//CONS/[ern] ⇒ ɛʁn

  =/CONS/[ern] ⇒ ɛʁn

  =[ern] ⇒ ɛʁn

  the rule

  =/CONS/*2[ern] ⇒ ɛʁn

  where /CONS/ designates the set of consonantal letters.

- Each $phonetics_i$ is a phonetic transcription of *string*. If *string* has (also) no phonetic equivalent, this is to be indicated with a comma <,> in place of one $phonetics_i$. Instead of n alternative phonetic transcriptions, an identifier of a phone set containing these alternative transcriptions may occur on the right side of a rule.

That the left side of a rule (i.e., *left*, *string*, and *right*) may contain morph boundary symbols facilitates that the letter-to-phone rules can consider the morphological structure of the segmented words. This is a considerable advantage compared with our former phonetization system. Furthermore, since more than one phonetic string or a phonetic set may occur on the right side of a rule, it is possible to create multiple transcriptions per word. This is a further substantial advantage.

In a rule set, specific rules have to be placed ahead of the general rules, in order to ensure that the specific rules can be applied. The interpreter of the phonetic component applies a set of letter-to-phone rules in the following way: It starts the transcription of a morphological segmentation with the first character after the initial <#>. This is always a morph boundary symbol. With this character, and the left and right context of the character in the segmentation, "in mind," the interpreter searches for the first matching rule, in the sequence of rules, whose *string* begins with the character. The right side of the first matching rule is appended to the output buffer, where the phonetic transcriptions are created. Now the interpreter advances in the segmentation as many

Table 6 Examples of set definitions and of letter-to-phone rules (in the formalism used in the new phonetization system)

```
Set definitions:
   SET /SUFFIX/    =  {isch ig ung}
   SET /BVOWEL/    =  {a o u}

Letter-to-phone rules:
   = [ch]l          ⇒  k
   = [ch]r          ⇒  k
   = [ch]           ⇒  ʃ
   [sch]            ⇒  ʃ
   [ch]s%ler        ⇒  k
   [ch]s%/SUFFIX/   ⇒  k
   /BVOWEL/[ch]     ⇒  x
   [ch]             ⇒  ç

   Where morph boundary is represented by:

      = before a stem
      % before suffixes of German origin
```

characters as the *string* of the applied rule contains, and thus it continues with the next character in the segmentation, which has not yet been transcribed. With this character and its context, it restarts the process just described. The interpreter continues this entire process, until the final word boundary symbol <#> has been met.

It is still necessary to describe the actions of the interpreter in the following two situations.

1. If there is more than one phonetic transcription available on the right side of a matching rule, the interpreter produces as many copies of the output string already available as there are different phonetic transcriptions on the right side, and it appends to each copy one of the transcriptions.
2. If a character in a segmentation does not match any rule, the interpreter distinguishes two cases:
   - If the character is a morph boundary symbol, it is skipped and the next character in the segmentation is processed.
   - Else, a copy of the character is appended to the output buffer and then the next character in the segmentation is processed.

To show how letter-to-phone rules can be expressed with the formalisms just introduced, we reproduce in Table 6 the rules of Table 2 with these formalisms. In this representation we take the opportunity to indicate morph boundaries in the rules.

Currently, 1460 rules are implemented. Approximately 180 of these rules deal with words that the morphological component cannot segment.

A complete listing of all set definitions and letter-to-phone rules has been published by Heinecke and Wothke in Reference 28.

Statistical results

We determined the linguistic performance of the components for morphological segmentation and for phonetic transcription by applying them to the words on rank 1–2000 (control set I) and on rank

Table 7 Statistical evaluation of automatically generated morphological segmentations

Segmentation Data	Types of Control Set I (n = 2,000)		Types of Control Set II (n = 1,000)	
Number of words segmented	1,915		851	
Percentage related to total number of types (= n)		95.75%		85.10%
Number of segmentations	2,095		1,011	
Ratio of segmentations obtained per segmented word on average		1.09		1.19
Number of correct segmentations	2,045		916	
Percentage related to total number of segmentations		97.61%		90.60%
Number of incorrect segmentations	50		95	
Percentage related to total number of segmentations		2.39%		9.40%
Number of words with at least one correct segmentation	1,913		844	
Percentage related to number of segmented words		99.90%		99.18%

200 001–201 000 (control set II) of a descending frequency list created from a machine-readable newspaper corpus with a total size of 13 208 225 words (tokens). The 2000 most frequent words (the types in control set I) correspond to 9 532 135 tokens, i.e., they cover 72.2 percent of all tokens in the corpus.

Results of morphological segmentation. Table 7 shows statistical data concerning the performance of the segmentation procedure. Of the types in the control sets, 92.2 percent ((2 × 95.75% + 85.10%)/3) were segmented. More than 99 percent of the segmented words received a correct segmentation.

Table 8 gives an overview of the words that were not segmented. Many of them are words with spelling errors, foreign words, and proper names, which one cannot expect to be morphologically segmented.

Results of phonetic transcription. Table 9 shows the statistical results of an evaluation of automatically generated phonetic transcriptions. A similar table, containing slightly less affirmative results, was also published in Reference 29. Meanwhile, we have improved the system, and the positive results can be seen in Table 9. The table shows better results for the tokens of control set I than for its types. This conforms very well with the requirements of our speech recognition system, which uses the transcriptions as reference units, because it will serve for the dictation of texts (consisting of tokens) and not for the dictation of dictionaries (consisting of types). The reason for the better results in the token statistics is that the very frequent words are transcribed correctly altogether.

Table 8 Statistical results for words rejected by the segmentation procedure

Segmentation Data	Types of Control Set I and II (n = 3,000)	
Number of words rejected out of total number of types (= n)	234	
Words with spelling errors	6	(2.56%)
Foreign words that are not used in German	10	(4.27%)
Proper names	73	(31.20%)
Words that should be segmented	145	(61.97%)

The percentual average for "only incorrect transcriptions" for Types of Control Set I and Types of Control Set II is 1.47 percent, i.e., (2 × 0.6% + 3.2%)/3. In other words, more than 98 percent of the types received a correct transcription.

The inspection of the words with exclusively incorrect transcriptions revealed that a large number of them are proper names (e.g.: <John>, <Gorbatschow>) or words of foreign origin (e.g.: <Ensemble>, <Aids>, <Team>) for which the usual German correspondence rules between letters and phones are not valid.

Table 9 Statistical evaluation of automatically generated phonetic transcriptions

Transcription Data	Types of Control Set I (n = 2,000)		Tokens of Control Set I (n = 9,532,135)		Types of Control Set II (n = 1,000)	
Completely with correct transcriptions, no incorrect transcriptions	1,952	(97.60%)	9,466,084	(99.31%)	913	(91.30%)
Incompletely with correct transcriptions, no incorrect transcriptions	13	(0.65%)	13,611	(0.14%)	28	(2.80%)
Completely with correct transcriptions, additionally incorrect transcriptions	22	(1.10%)	37,014	(0.39%)	27	(2.70%)
Incompletely with correct transcriptions, additionally incorrect transcriptions	1	(0.05%)	877	(0.01%)	0	(0.00%)
Only incorrect transcriptions	12	(0.60%)	14,549	(0.15%)	32	(3.20%)

Conclusion

In this paper, we described an approach to the automatic phonetic transcription of German words. The transcription process consists of the two main steps, *morphological segmentation* and morphology-based *phonetic transcription*. The potentiality of generating multiple transcriptions per word and the integration of morphological knowledge into the transcription system contributed to a substantial improvement of this system's linguistic performance.

This system is currently being used with great success for the automatic generation of phonetic transcriptions in the German version of the large-vocabulary speech recognition system TANGORA. Another potential application area of the system lies in the area of text-to-speech synthesis, where it could serve to provide intermediate phonetic transcriptions of the words to be acoustically synthesized. The morphological component of the system could possibly also be used in linguistically-based full-text retrieval systems in order to determine the word stems.

In the future, our research and development efforts may possibly concentrate on the improvement of the morphological segmentation rate for common German words and of the transcription quality for proper names and foreign words.

Acknowledgments

I wish to express my gratitude for the encouragement received from my manager Eric Keppel.

My special thanks go to Ute Breitinger (Technical University Darmstadt, Germany) who implemented extensive parts of the preprocessor and of the interpreter for letter-to-phone rules, Rudolf Schmidt who was substantially involved in the implementation of the software of the morphological segmentation system, Thomas Pachunke and Oliver Mertineit who developed the German word syntax and most of the German morph dictionary, and Johannes Heinecke who developed a vast section of the letter-to-phone rules for German and implemented the modules for the mapping of digits on their full orthographic equivalents.

Furthermore, I thank Georg Walch for his support. Christine Sperling gave me valuable support when proofreading this paper.

Of course none of the above is in any way responsible for any errors in this paper.

Appendix: Transcription examples

The examples of morphological segmentations and phonetic transcriptions that begin on the next page consist of three columns:

- Column 1 lists the input words.
- Column 2 contains for each input word, one or more automatically generated morphological segmentations. Unsegmented words are prefixed with <?>.
- Column 3 displays one or more automatically generated transcriptions for each segmentation. Two special features of these transcriptions are: (1) Stress markers are missing. Stress is neglected by the current letter-to-phone rules, because the speech recognition system, for which the transcriptions are generated as reference units, does not take into account stress during the recognition process. (2) For reasons of clarity, transcriptions with [ə]-elision are not listed below. (Our automatic transcription system also generates the appropriate transcription variants with [ə]-elision.)

<abgehefteter>	#+ab+ge=heft%et%er#	[ʔapgəhɛftətɐ]
<Abrollapparat>	#+Ab=roll=apparat#	[ʔaprɔlʔapara:t] [ʔaprɔlapara:t]
<achtarmig>	#=acht=arm%ig#	[ʔaxtʔarmɪç] [ʔaxtʔarmɪk] [ʔaxtarmɪç] [ʔaxtarmɪk]
<Alliierte>	#+Al=li%ier%te#	[ʔalɪʔi:ɐtə] [ʔali:ɐtə]
<Angora>	#=Angora#	[ʔaŋgo:ra]
<Angst>	#=Angst#	[ʔaŋst]
<Arbeitnehmeraktie>	#=Arbeit=nehm%er=akti%e#	[ʔarbaɐ̯tne:mɐʔaktsjə] [ʔarbaɐ̯tne:mɐaktsjə]
<archaisch>	#=archa%isch#	[ʔarçaɪʃ]
<Athen>	#?Athen#	[ʔate:n]
<außerdem>	#=außer=dem#	[ʔaʊ̯sɐde:m]
<Ausreiseantrag>	#+Aus=reis%e+an=trag#	[ʔaʊ̯sraɐ̯ezəʔantra:k] [ʔaʊ̯sraɐ̯ezəantra:k]
<Baal>	#?Baal#	[ba:l]
<Bettuch>	#=Bet=tuch# #=Bett=tuch#	[be:tu:x] [bɛtu:x]
<beurteilender>	#+be=urteil%end%er#	[bəʔurtaɐ̯ləndɐ] [bəurtaɐ̯ləndɐ]
<brachen>	#=brach%en#	[bra:xən]
<brachst>	#=brach%st#	[bra:xst]
<brachte>	#=brach%te#	[braxtə]
<Chance>	#=Chanc%e#	[ʃãsə] [ʃaŋsə]
<Charlotte>	#=Charlott%e#	[ʃaɐ̯lɔtə]
<Chauffeur>	#=Chauff~eur#	[ʃɔfø:ɐ]
<chic>	#=chic#	[ʃɪk]

< China >	# = Chin_a#	[çına] [kına]
< daher >	# = da = her#	[da:he:ɐ] [dahe:ɐ]
< Detail >	# = Detail#	[de:tai̯]
< Ehevertrag >	# = Eh%e + ver = trag#	[ʔe:əfɐtra:k]
< Entbindungspflegers >	# + Ent = bind%ung%s = pfleg%er%s#	[ʔɛntbɪnduŋspfle:gɐs] [ʔɛntbɪnduŋsfle:gɐs]
< erblichen >	# + er = blich%en# # = erb%lich%en#	[ʔɐblɪçən] [ʔɛrplɪçən]
< Familie >	# = Famili%e#	[fami:ljə]
< Fehlregulation >	# = Fehl = regul_at_ion#	[fe:lregulatsjo:n]
< flottgemachtes >	# = flott + ge = mach%tes#	[flɔtgəmaxtəs]
< Friseuse >	# = Fris~eus%e#	[frɪzø:zə]
< furchtbarerem >	# = furcht%bar%er%em#	[fʊrçtba:rərəm] [fʊrçtba:rɐm]
< Geldmarktpapiers >	# = Geld = markt = papier%s#	[gɛltmarktpapi:ɐs]
< Geschäftsabschlusses >	# + Ge = schäft%s + ab = schluss%es#	[gəʃɛfts?apʃlʊsəs] [gəʃɛftsapʃlʊsəs]
< großräumigste >	# = groß = räum%ig%st%e#	[gro:srɔ̯ømɪkstə] [gro:srɔ̯ømɪçstə]
< Häuschen >	# = Häus%chen#	[hɔ̯øsçən]
< Haushaltsersparnisse >	# = Haus = halt%s + er = spar%niss%e#	[haʊ̯saltsʔɐʃpa:rnɪsə] [haʊ̯saltsɐʃpa:rnɪsə]
< hinzuziehende >	# + hinzu = zieh%end%e#	[hɪntsʊtsi:əndə] [hɪnsʊtsi:əndə]
	# + hin + zu = zieh%end%e#	[hɪntsʊtsi:əndə] [hɪntsu:tsi:əndə]
< hochgeklapptes >	# = hoch + ge = klapp%tes#	[ho:xgəklaptəs]
< Ideen >	# = Idee%n#	[ʔɪde:n]
< Ingenieur >	# = Ingeni~eur#	[ʔɪnʒenjø:ɐ]

< initiierst >	# = initi%ier%st#	[ʔɪnɪʦɪʔiːɐ̯st] [ʔɪnɪʦiːɐ̯st]
< Interessenausgleiche >	# = Interess%en + aus = gleich%e#	[ʔɪntərɛsənʔa͡ʊsgla͡ɪçə] [ʔɪntərɛsənə a͡ʊsgla͡ɪçə]
< Italien >	# = Itali%en#	[ʔɪtaːljən]
< Kalkspat >	# = Kalk = spat#	[kalkʃpaːt]
< klargestellte >	# = klar + ge = stell%te#	[klaːrgəʃtɛltə]
< Korbflechterei >	# = Korb = flecht%er = ei#	[kɔrpflɛçtɐʔa͡ɛ] [kɔrpflɛçtɐa͡ɛ]
	# = Korb = flecht%er%ei#	[kɔrpflɛçtərə a͡ɛ]
< kurzzeitige >	# = kurz = zeit%ig%e#	[kʊrʦʦa͡ɛtɪgə]
< Lärmempfindlichkeit >	# = Lärm + emp = find%lich%keit#	[lɛːrmʔɛmp͡fɪntlɪçka͡ɛt] [lɛːrmɛmp͡fɪntlɪçka͡ɛt] [lɛːrmʔɛmfɪntlɪçka͡ɛt] [lɛːrmɛmfɪntlɪçka͡ɛt]
< leitetet >	# = leit%etet#	[la͡ɛtətət]
< makelloseste >	# = mak%el = los%est%e#	[maːkəlɔːzəstə]
< moussiert >	# = mouss%ier%t#	[mʊsiːɐt]
< Nacktbadestrand >	# = Nackt = bad%e = strand#	[naktbaːdəʃtrant]
< niederhiebst >	# = nieder = hieb%st#	[niːdɐhiːpst]
< notorischem >	# = not_or%isch%em#	[nɔtoːrɪʃəm]
< pflichteifrigere >	# = pflicht = eifr%ig%er%e#	[p͡flɪçtʔa͡ɛfrɪgərə] [p͡flɪçta͡ɛfrɪgərə] [flɪçtʔa͡ɛfrɪgərə] [flɪçta͡ɛfrɪgərə]
< plakativere >	# = plak_at_iv%er%e#	[plakatiːvərə]
< Prozeßautomation >	# = Prozeß = automat_ion#	[prɔʦɛsʔa͡ɔtɔmaʦjoːn] [prɔʦɛsa͡ɔtɔmaʦjoːn]
< querschlagt >	# = quer = schlag%t#	[kveːrʃlaːkt]
< Radachse >	# = Rad = achs%e#	[raːtʔaksə] [raːtaksə]
< Radar >	# = Rad_ar#	[radaːr]

< ringartige >	# = ring = art%ig%e#	[rɪŋʔartɪgə] [rɪŋartɪgə]
< Ruin >	# = Ruin#	[ruiːn]
< sahen >	# = sah%en#	[zaːən]
< schlagkräftigerer >	# = schlag = kräft%ig%er%er#	[ʃlaːkrɛftɪgərɐ]
< Schleuse >	# = Schleus%e#	[ʃlɔ͡øzə]
< Serie >	# = Seri%e#	[zɛrjə]
< strapazierbarste >	# = strapaz%ier%bar%st%e#	[ʃtraːpat͡siːrbaːrstə]
< Systemanschlüssen >	# = System + an = schlüss%en#	[zʏsteːmʔanʃlʏsən] [zʏsteːmanʃlʏsən]
< täuschen >	# = täusch%en#	[tɔ͡øʃən]
< telegraphische >	# + tele = graph%isch%e#	[teːləgraːfiʃə]
< Traummanager >	# = Traum = manag~er#	[traʊ̯mɛnədʒɐ] [traʊ̯mɛnədʃɐ]
< trübsinnigerer >	# = trüb = sinn%ig%er%er#	[tryːpzɪnɪgərɐ] [tryːpsɪnɪgərɐ]
< unerwartetste >	# + un + er = wart%et%st%e#	[ʔʊnʔɐvartət͡stə] [ʔʊnɐvartət͡stə]
< unsensibel >	# + un = sens_ibel#	[ʔʊnzɛnsiːbəl]
< Vakuumkammer >	# = Vaku_um = kamm%er#	[vaːkʊʊmkamɐ]
< vorzuschießendes >	# + vor + zu = schieß%end%es#	[foːrt͡sʊʃiːsəndəs] [foːrt͡suːʃiːsəndəs]
< wassergekühltes >	# = wass%er + ge = kühl%tes#	[vasɐgəkyːltəs]
< zielstrebigster >	# = ziel = streb%ig%st%er#	[t͡siːlʃtreːbɪçstɐ] [t͡siːlʃtreːbɪkstɐ]
< röm.-kath. >	# = röm%isch = kathol%isch#	[røːmɪʃkatoːlɪʃ]
< u.a. >	# = und = ander%es#	[ʔʊntʔandərəs] [ʔʊntʔandɐs] [ʔʊntandərəs] [ʔʊntandɐs]

	#=und=ander%e#	[ʔʊntʔandərə] [ʔʊntandərə]
	#=unter=ander%em#	[ʔʊntɐʔandərəm] [ʔʊntɐʔandɐm] [ʔʊntɐandərəm] [ʔʊntɐandɐm]
<23>	#=drei=und=zwanzig#	[drae̯ʔʊnt͡svant͡sɪç] [drae̯ʔʊnt͡svant͡sɪk] [drae̯ʔʊnt͡svansɪç] [drae̯ʔʊnt͡svansɪk] [drae̯ʊnt͡svant͡sɪç] [drae̯ʊnt͡svant͡sɪk] [drae̯ʊnt͡svansɪç] [drae̯ʊnt͡svansɪk] [drae̯ʔʊnsvant͡sɪç] [drae̯ʔʊnsvant͡sɪk] [drae̯ʔʊnsvansɪç] [drae̯ʔʊnsvansɪk] [drae̯ʊnsvant͡sɪç] [drae̯ʊnsvant͡sɪk] [drae̯ʊnsvansɪç] [drae̯ʊnsvansɪk]
<3.>	#=dritt%e# #=dritt%er# #=dritt%em# #=dritt%en# #=dritt%es# #=dritt%ens#	[drɪtə] [drɪtɐ] [drɪtəm] [drɪt̬m] [drɪtən] [drɪt̬n] [drɪtəs] [drɪtəns] [drɪt̬ns]
<3,42>	#=drei=Komma=vier=zwei#	[drae̯kɔmafiːɐt͡svae̯]
<->	#=minus# #=Bind%e=strich#	[miːnʊs] [bɪndəʃtrɪç]
<$>	#=Dollar# #=Dollar=zeich%en#	[dɔlaːr] [dɔlaːrt͡sae̯çən]
<#>	#=Doppel=kreuz#	[dɔpəlkrɔø̯t͡s]
<,>	#=Komma# #+Bei=strich#	[kɔma] [bae̯ʃtrɪç]
<KO-Großschreibung>	#?KO-Großschreibung#	[koːgroːsʃrae̯bʊŋ] [koːgroːʃrae̯bʊŋ]

Cited references and notes

1. K. Wothke, U. Bandara, J. Kempf, E. Keppel, K. Mohr, and G. Walch, "The SPRING Speech Recognition System for German," *Proceedings of Eurospeech '89*, European Conference on Speech Communication and Technology, Paris, September 1989, J. P. Tubach and J. J. Mariani, Editors, Vol. II (1989), pp. 9–12.
2. F. Jelinek, "The Development of an Experimental Discrete Dictation Recognizer," *Proceedings of the IEEE* **73**, No. 11, 1616–1624 (1985).
3. A survey of transcription components for text-to-speech systems is given in: K. Wothke, "Computergestützte Verfahren zur phonologischen Beschreibung," *Computational Linguistics: An International Handbook of Computer Aided Language Research and Applications*, I. S. Bátori, W. Lenders, and W. Putschke, Editors, de Gruyter-Verlag, Berlin, New York (1989), pp. 175–188.
4. An older and now partially obsolete description of the system for the automatic generation of phonetic transcriptions is given by Wothke in Reference 29.
5. M. Kommenda, *Automatische Wortstrukturanalyse für die akustische Ausgabe von deutschem Text*, doctoral thesis at the Technical University, Vienna, Austria (1991).
6. B. Schnabel and H. Roth, "Automatic Linguistic Processing in a German Text-to-Speech Synthesis System," *Proceedings of the ESCA Workshop on Speech Synthesis*, Autrans, France (September 25, 1990), pp. 121–124.
7. H. E. Wolf, F. Strecker, and G. Fries, "Text-Sprache-Umsetzung für Anwendungen bei automatischen Informations- und Transaktionssystemen," *Informationstechnik* **31**, No. 5, 334–341 (1989).
8. S. Koch, A. Küstner, and B. Rüdiger, "Deutsche Wortformensegmentierung ohne Lexikon," *Sprache und Datenverarbeitung*, No. 1, 35–44 (1989).
9. G. Thurmair, "Eine maschinelle morphologische Analyse des Deutschen," *Informationslinguistische Texterschließung*, Chr. Schwarz and G. Thurmair, Editors, Olms-Verlag, Hildesheim, Zürich, New York (1986), pp. 66–107.
10. S. Scarci and S. Taraglio, "Automatic Phonetic Transcription for Large-Vocabulary Speech Recognition," *Proceedings Speech '88*, R. Lawrence, Editor, Edinburgh, Great Britain, August 22–26, 1988, pp. 771–777.
11. *The Principles of the International Phonetic Association*, University College, Gower Street, London (1949; reprinted 1977).
12. P. Ladefoged, "Some Reflections on the IPA," *Journal of Phonetics* **18**, 335–346 (1990).
13. *Siebs. Deutsche Aussprache. Reine und gemäßigte Hochlautung mit Aussprachewörterbuch*, 19th Rev. Ed., H. de Boor, H. Moser, and Chr. Winkler, Editors, de Gruyter-Verlag, Berlin (1969).
14. Duden, *Aussprachewörterbuch*, 2nd Rev. Ed., Bibliographisches Institut, Mannheim, Wien, Zürich (1974).
15. *Großes Wörterbuch der deutschen Aussprache*, E.-M. Krech et al., Editors, Bibliographisches Institut, Leipzig (1982).
16. K. Wothke, "TETOS—A Text-to-Speech System for German," *Proceedings of the 1990 International Conference on Spoken Language Processing (ICSLP '90)*, Kobe, Japan, 1990, Vol. II (1990), pp. 849–852.
17. K. Wothke, *Letter-to-Phone Rules for German*, Technical Report 75.91.04, Heidelberg Scientific Center, IBM Germany, Heidelberg, Germany (February 1991).
18. R. Schmidt and K. Wothke, *A Multiple Backtracking Algorithm for Morphological Word Segmentation*, Technical Report 90.12.015, Heidelberg Scientific Center, IBM Germany, Heidelberg, Germany (December 1990).
19. K. Wothke and R. Schmidt, "A Morphological Segmentation Procedure for German," *Proceedings of the International Conference on Current Issues in Computational Linguistics*, Penang, Malaysia, June 10–14, 1991, pp. 137–147.
20. U. Breitinger, *Entwicklung eines Präprozessors sowie eines Interpreters zur phonetischen Transkription orthographischer Wörter*, diploma thesis, Technical University Darmstadt, Department of Computer Science (July 1990).
21. R. Schmidt, *Technical Aspects of a Phonetization Program Considering Morphological Peculiarities*, Technical Report 75.91.22, Heidelberg Scientific Center, IBM Germany, Heidelberg, Germany (August 1991).
22. W. D. Ortmann, *Wortbildung und Morphemstruktur hochfrequenter deutscher Wortformen*, Part I, Goethe-Institut, München, Germany (1985).
23. G. Wahrig, *Deutsches Wörterbuch*, Mosiak-Verlag, München, Germany (1987).
24. N. Chomsky, "On Certain Formal Properties of Grammars," *Information and Control*, No. 2, 137–167 (1959).
25. P. Hellwig, "Parsing natürlicher Sprachen: Realisierungen," *Computational Linguistics. An International Handbook of Computer Aided Language Research and Applications*, I. S. Bátori, W. Lenders, and W. Putschke, Editors, de Gruyter-Verlag, Berlin, New York (1989), pp. 378–432.
26. T. Pachunke, O. Mertineit, K. Wothke, and R. Schmidt, "Broad Coverage Automatic Morphological Segmentation of German Words," *Proceedings of the Fifteenth International Conference on Computational Linguistics*, Nantes, France, July 23–28, 1992, Vol. IV (1992), pp. 1218–1222.
27. N. Chomsky and M. Halle, *The Sound Pattern of English*, Harper & Row, New York (1968).
28. J. Heinecke and K. Wothke, *Letter-to-Phone Rules for German Taking into Account the Morphological Structure of Words*, Technical Report 75.92.03, Heidelberg Scientific Center, IBM Germany, Heidelberg, Germany (February 1992).
29. K. Wothke, *Automatic Phonetic Transcription Taking into Account the Morphological Structure of Words*, Technical Report 75.91.18, Heidelberg Scientific Center, IBM Germany, Heidelberg, Germany (July 1991; reprinted with corrections April 1992).

Accepted for publication September 15, 1992.

Klaus Wothke *IBM Heidelberg Scientific Center, Vangerowstrasse 18, D-69115 Heidelberg, Germany.* Dr. Wothke is currently a research staff member in the project on automatic speech recognition at the IBM Heidelberg Scientific Center. He joined IBM in 1988. His research interests lie in the area of computational linguistics and, currently, computational phonology, computational morphology, automatic speech

recognition, and text-to-speech systems. He holds an M.A. degree and a doctoral degree in communications research and phonetics (with main emphasis on computational linguistics), mathematical logic, linguistics, and audiology from the University of Bonn, Germany. He is author and coauthor of more than 20 scientific publications.

Reprint Order No. G321-5522.