Finite State Morphology and Left to Right Phonology

Jorge Hankamer
University of California, Santa Cruz

0. Introduction

Finnish, Turkish, and other agglutinative languages lend themselves to left-to-right morphological parsing by a strategy based on the fact that the morphology can be described by a finite state transition network (Koskenniemi 1983, Karttunen 1983, Hankamer 1984). This paper describes a morphological parser which analyzes words formed agglutinatively, and explores the theoretical implications of the parser's treatment of morphophonemic alternations.

The parser[1] employs a finite state transition network representation of morphotactics and a treatment of morphophonemic alternations which employs generative-type rules operating cyclically (i.e. left-to-right in a suffixing language). Morphophonemic alternation is accounted for by listing roots and affixes in a basic form, which is modified according to the surface environment when matching is done. These modifications correspond to the phonological rules of the language.

Because the parser proceeds strictly from left to right in constructing a morphological analysis, at any point in the analysis there can be no information available about the morphological environment to the right. This means that there can be no cases of morphophonemic alternation conditioned in terms of following morphological environment. One class of cases in Turkish appears to involve such right-to-left conditioning: the form of certain affixes and pronouns varies depending on whether a plural or case suffix immediately follows. I will show, however, that there is a reasonable reanalysis in which the right-to-left conditioning disappears, and that this analysis is no more complex than the standard one will have to be if it is to account for all the facts.

The parser thus presupposes a theory of phonology in which phonological rules apply to one morpheme at a time, starting with the root and progressing outward (rightward in an exclusively suffixing language). The phonology is strictly cyclic: once a morpheme has undergone the phonological rules and been matched to a surface string, it is never examined again; the cycle of phonological rules applies in a domain that is exactly the size of a morpheme.

Despite its conceptual simplicity and the strict requirement of left-to-right phonology, the parser appears adequate for all of Turkish morphology, with the exception of incorporation and compounding, which require certain embellishments.[2] It is also readily adaptable to other languages with agglutinating morphology.

The outline of the paper is as follows: section 1 briefly sketches the main problems confronting a morphological parser, particularly one designed to analyze words formed agglutinatively. Section 2 describes the treatment of morphotactics; section 3 describes the treatment of morphophonemic alternation. Section 4 contains a discussion of consequences of the assumptions underlying

the method of analysis, and a description of a particular problem in Turkish morphology where the assumptions force decisions regarding the analysis. Section 5 is a conclusion.

1. Morphological Parsing

The task facing a morphological analyzer is as follows: given as input a phonological representation of a word[3], to yield as output a set of morphemic representations consistent with the input representation. This output set may contain no members (the input string is not analyzable), one member (the input string is unambiguously analyzable into a morphemic representation), or several members (the input string is analyzable as several different combinations of morphemes).[4]

The major problems facing such an analyzer are two: (a) the input string contains no direct indication of where the morpheme boundaries are; (b) due to morphophonemic alternations, a given morpheme takes a shape dependent on its morphological and phonological environment. Thus it is not at all straightforward to specify how the string

yıkanabileceklerdi[5]

'they were going to be able to be washed'

is to be analyzed as the sequence of morphemes

wash-PASS-POT-FUT-PL-PAST

In particular, the PASS (passive) morpheme takes nine different shapes, depending on environment: *ıl*, *il*, *ül*, *ul*, *ın*, *in*, *ün*, *un*, and *n*. The conditioning is completely phonological, but the program has to know when an *n* can be counted as a realization of the passive morpheme and when it cannot.

Two factors are relevant to deciding whether an *n* in the input string can count as a realization of the passive morpheme. First, the morphological environment must be right—the preceding string must be analyzable as a verb stem (in fact, as a particular subcategory of verb stem). Second, the phonological environment has to be right—the stem must end in a vowel.

Keçi separates the problems of morphological and phonological environment. Morphotactic restrictions are encoded in a finite state transition network representation, which defines the class of well-formed morphemic representations. This is done by assigning each root to a basic stem category, which determines the class of affixes that may attach to it. Affixes are assigned to complex categories which combine with stem categories to yield other stem categories. The result is a categorial grammar which recursively defines the well-formed stems. Since the only syntagmatic operation is concatenation (of the segmental representation of the functor to the segmental representation of its argument), the categorial grammar is equivalent to a finite state transition network in which stem categories correspond to states and affix categories

correspond to transitions from states to states.

For example, in Turkish the suffix *-la* is in a category N0\V0, meaning that it combines with a stem of category N0 to yield a derived stem of category V0. V0 and N0 are the basic verb and noun categories. Hence *-la* [N0\V0] can combine with a noun *kilit* [N0] ('lock') to yield a verb *kilitle* [V0] ('lock'). Two stems are in the same category only if they accept exactly the same class of affixes, so the number of stem categories is rather large.

Parsing proceeds as follows: the parser begins in a designated initial state. It can transit to one of the states corresponding to a basic stem category by recognizing an initial substring of its input as a surface realization of one of the roots in its root lexicon. The root category determines a class of affixes that are permitted in next position. The parser searches an affix lexicon for an affix that is in the permitted class and matches the surface string at the current point. If one is found, a pointer is advanced to the new current point in the word and the parser jumps to the new state, corresponding to the derived stem category, determined by the affix. This process iterates. If the end of the input string is reached and the parser is in a designated final state, the string is successfully parsed. Section 2 describes the morphotactics in more detail.

Morphophonemic alternation is accounted for by a phonological component which mediates between the lexical and surface forms of morphemes. Both roots and affixes are listed in the lexicon in a basic ("lexical") form, which is subjected to a set of phonological rules which convert lexical representations of candidate morphemes into forms consistent with surface environment. These rules are described in section 3.

2. Morphotactics

Diagram 1 is a somewhat simplified graphic representation of the morphotactic network.[6] Transitions from the start state 00 represent root categories. The root categories shown in the diagram are N0 (noun roots), Q0 (predicate roots), V0 (verb roots), A0 (adjective roots), and Z0 (adverb roots). There are of course several further root categories which are not shown.

Here is an example of the parsing of an only moderately complex word, according to this morphotactics. Given the input

çöplüklerimizdekilerdenmiydi

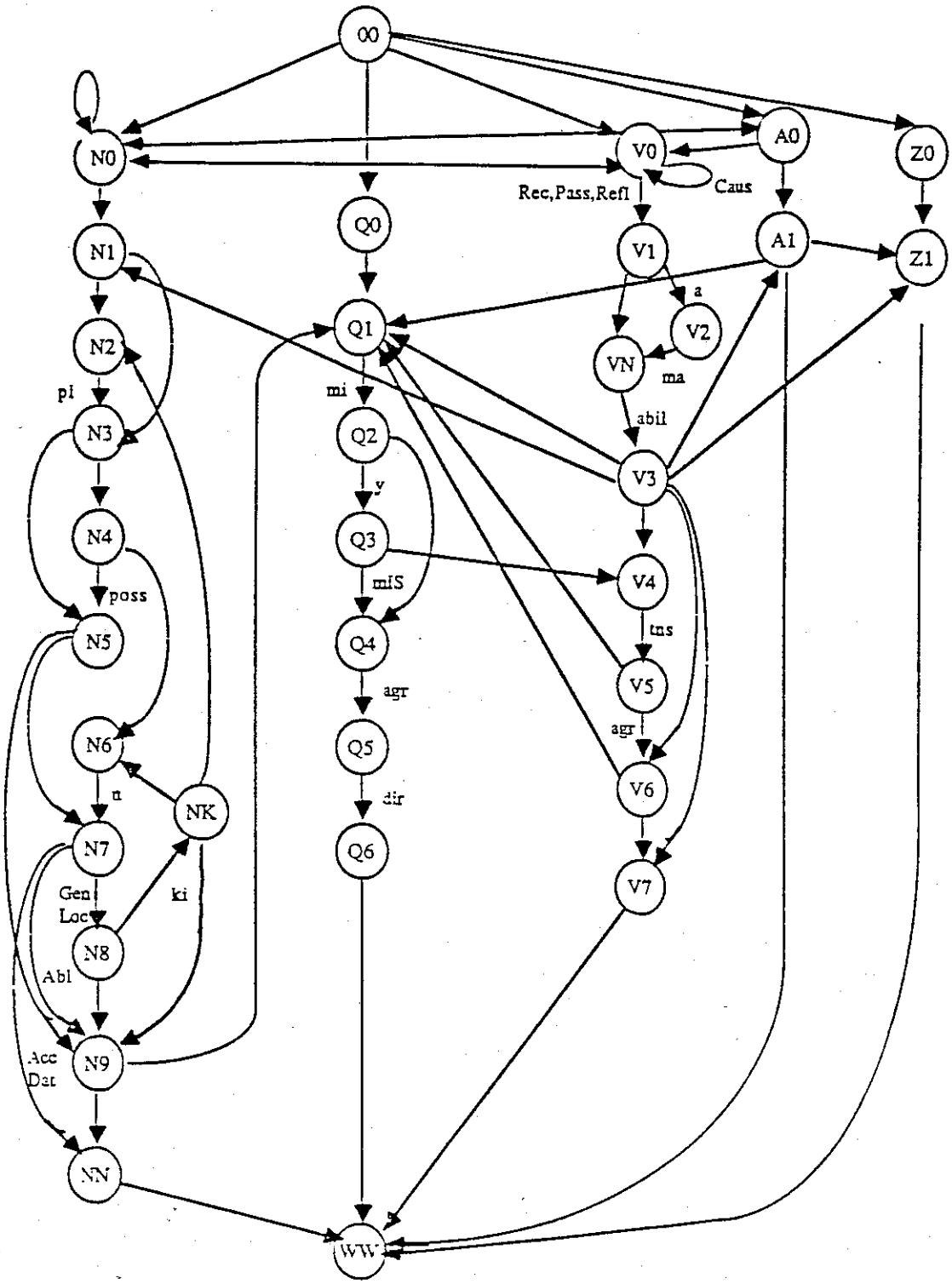
'was it from those that were in our garbage cans?'

the analysis proceeds as follows:

First the root lexicon is consulted, and the form *çöp* 'garbage' is found to match the first three segments of the input.

The root *çöp* determines a stem category N0. Now an affix is sought that can combine with a stem of category N0 and matches the initial substring of the remainder of the input form. Such an affix is the affix *-lıg* which matches the input due to vowel harmony and final stop devoicing. *-lıg* is in the category N0\N0, i.e. it combines with a stem of category N0 to yield a new stem of category N0.

Diagram 1


00
N0
V0
A0
Z0
Caus
Rec,Pass,Refl
Q0
N1
V1
A1
Z1
a
Q1
N2
V2
VN
ma
pl
mi
N3
Q2
abil
V3
y
N4
Q3
V4
poss
mIS
tns
N5
Q4
V5
agr
agr
N6
Q5
V6
n
dir
NK
N7
Q6
V7
Gen
Loc
ki
N8
Abl
Acc
Dat
N9
NN
WW

Hence *çöplük* is analyzed as a stem of category N0. The process now iterates.

Among the affixes that can combine with a stem of category N0 is the affix *-la* which matches *le* in the input. This will be tried, but the attempt will lead to failure because *-la* is an N0\V0 affix,[7] and there are no affixes in the affix lexicon that match at the new point in the input and can combine with a stem of category V0.

No overt affix leads to a successful parse from this point. The morphotactics, however, provides for certain free jumps between categories. In particular, any stem of category N0 counts as a stem of category N1, and there is thus a free jump from N0 to N1 in the morphotactic network.[8] Thus *çöplük* counts as a member of category N1.

From N1 there are jumps to N2 and to N3. If the jump to N2 is taken, the next affix must be the plural affix *-lar*; if the jump to N3 is taken, the next affix must not be the plural.

Both paths will be attempted, but the one which commences with a jump to N3 will fail, since there is no way to reach the end of the word along that path. The path commencing with a jump to N2 leads to the correct analysis.

The only affix that can attach to an N2 is the plural affix *-lar*, which is the sole occupant of the category N2\N3; this matches the *ler* after *çöplük*, so *çöplükler* is analyzed as a stem of category N3.

Omitting unnecessary details, the rest of the analysis proceeds with successive recognition of the affixes *-ımız* (N4\N5, 1pl possessive), *-da* (N7\N8, locative), *-ki* (N8\NK, relative), *-lar* (N2\N3, plural), *-dan* (N7\N9, ablative), *-mı* (Q1\Q2, interrogative), *-y* (Q2\Q3, auxiliary), and *-dı* (V4\V5, past). There are free jumps from V5 through V6 and V7 to the final state WW, so the word is successfully parsed.

```
çöp-lük---ler--imiz--de-ki--ler---den--mi-y--di----
   N    NNN    NN     NN  N  NN    NNN    NQ  Q QV  VVVW
   0    012    34     57  8  K2    357    91  2 34  567W
```

The morphotactics contains several loops. Note that in the example just given, the function of the "relative" affix *-ki* is to provide a path back to earlier levels in the nominal stem hierarchy. Speakers of Turkish do not, of course, make use of all of the words provided for by this recursion, but it does appear to be productive.

Somewhat more dubious is the indirect recursion V4-V5-Q1-Q2-Q3-V4, which is there to capture some of the more frightening complexities in the verbal and predicate inflection, and is almost certainly not right. I don't want to talk about it.

3. Morphophonemics

The morphophonology is encoded in the functions which determine whether a given stretch of the surface string matches a root or suffix entry. There are two different functions, one for roots and one for suffixes, because the rules are not exactly the same for roots and suffixes. What these functions do is modify the

basic form of the morpheme to make it compatible with its surface environment.

When the matching function is called, it is passed a copy of the lexical form of the morpheme to be matched and a pointer to the current location in the surface string (i.e. to the beginning of the unanalyzed portion of the word).

The morphophonemic rules themselves are conditional statements within the matching functions. These rules have access to the lexical form of the morpheme currently under consideration, which they may modify (and consequently affect the action of subsequent rules applying to that same morpheme) and to the surface string.

The action taken by a morphophonemic rule may thus in principle depend on any property of the lexical representation of the morpheme, as modified by previous morphophonemic rules, and on any property of the surface string. The rules do not have access to non-surface representations of previous or following morphemes, however. Given the design of the parsing strategy, it is impossible for rules to refer to nonsurface context to the right, since that part of the string will not have been analyzed yet. It is possible to allow them to refer to lexical context to the left, and that is probably going to be necessary in order to deal with lexical exceptions to vowel harmony (*saat-i*, *harb-i*). The current implementation does not treat such exceptions correctly, since the matching function does not receive any information about lexical context at all.

The morphophonemic rules are of two kinds: those which apply only at morpheme boundaries, and those which must inspect each segment in the form. Final stop devoicing and suffix-initial stop voicing assimilation are examples of the former, and vowel harmony is an example of the latter. Boundary rules are applied only to the appropriate (initial or final) segment, looking at the context in the surface string to determine what action to take. Every-segment rules inspect each segment in the form, proceeding from left to right, modifying those segments that meet the conditions of application.

To take a simple example, suppose the surface form is *simitçi* ('simit-seller'). The rootsearch function will find an entry in the lexicon with the form *simid* and category N0. A copy of this form will be passed to a matching function along with a pointer to the beginning of the word. The only rule applicable to this form is final stop devoicing, which looks at the *d* which needs to be matched to the fifth segment in the surface form, inspects the sixth segment in the surface form to see if it is a vowel, finds that it is not and consequently replaces the *d* with its voiceless counterpart.

After all the morphophonemic rules have applied as necessary, the cooked form is matched against the corresponding surface substring. If the match is exact, the analysis proceeds with a call to the suffix search routine, which in this case will be called with a pointer to the *ç* in the surface string and the stem category N0. The suffix search routine looks in the suffix inventory for suffixes that sanction transitions from N0, attempting to match each one to the surface string starting at the *ç*.

One such suffix has the form *cı* and transition category N0\N0 (combines with an N0 to form an N0). The matching function receives a copy of this form and a

pointer to the *ç*. The rule for suffix-initial stop voicing assimilation is applicable. It turns a suffix-initial voiced stop into its voiceless counterpart if the preceding segment in the surface form is voiceless. The *c* is thus replaced by *ç* and the form is now *çı*. Then vowel harmony applies, modifying each suffix vowel in accord with the preceding vowel in the surface form. This replaces *ı* by *i* in this case.

Finally the cooked form is matched against the surface substring of equivalent length starting at the current location, and the match is perfect.

When a successful match is found, the suffix search function calls itself, passing as parameters a pointer to the new current location in the surface string and the current stem category, determined by the transition category of the just-analyzed suffix. In this example, the pointer will be pointing at the end of the string. When the search function is called with a null string, it consults the suffix inventory to see if the current stem category is a legitimate final category (i.e. if the stem can be a word). If so, the parse is a success.

The parser deals correctly with final stop devoicing, the *k* ~ *ğ* alternation, suffix-initial stop voicing assimilation, vowel harmony except for lexical exceptions, buffer consonants, suffix-initial vowel deletion, vowel raising, and several minor kinds of alternations. Cases of suppletion, as in the personal pronouns (*ben* ~ *ban-a*), are dealt with by listing the suppletive forms separately in the lexicon and controlling the possible continuations in the morphotactics.

One interesting class of alternations involves the insertion of an epenthetic vowel in certain roots. For example, the root meaning 'city' has the form *şehir* in isolation or when followed by a consonant, but *şehr* when followed by a vowel. The right way to deal with this is clearly to recognize the form *şehr* just when the *r* can be syllabified with a following syllable, but the program currently has no direct access to syllable structure. It should not be difficult to fix this, since the syllable canon is very simple and syllabification can be read directly from the surface representation.

The current implementation works as follows: the root is listed in the lexicon as *şehr*. The root matching function checks final consonant clusters in underlying forms of roots, and when the cluster exhibits rising sonority, as in the case of *şehr*, a high vowel is inserted between the elements of the cluster unless a vowel follows in the surface form.

This epenthetic vowel must harmonize with the preceding vowel of the root, so after the vowel is inserted the vowel harmony rule is applied to it. Thus *şehr* becomes *şehir*, but *oğl* becomes *oğul*, etc.

There are also some alternations that depend on the number of preceding syllables, such as the aorist suffix (lexical form *-ır* for all polysyllabic stems, *-ar* for most monosyllabic stems). These are currently listed separately, and the parser cannot reject a word for containing the wrong one. This will be fixed once a syllabification routine is implemented.

4. Consequences: 'pronominal *n*'

As pointed out in the preceding section, the keçi model does not permit phonological rules to be sensitive to following morphological context across

morpheme boundaries.[9] There is a phenomenon in Turkish which might appear to be an instance of such morphological conditioning from the right. Certain pronominal roots (*bu* 'this', *o* 'that', and *şu* 'that right there') and affixes (*ki* 'relative', *sı* '3sg possessive') exhibit a stem-final *n* (so-called "pronominal *n*", cf. Lewis (1967, p. 40)) when followed by certain affixes. Consider the following forms derived from *bu* as compared with corresponding forms from *maymun* 'monkey' and *kutu* 'box':

bu		maymun	kutu	
bunu	(acc)	maymunu	kutuyu	(kutu+yı)
buna	(dat)	maymuna	kutuya	(kutu+ya)
bunda	(loc)	maymunda	kutuda	(kutu+da)
bundan	(abl)	maymundan	kutudan	(kutu+dan)
bunlar	(pl)	maymunlar	kutular	(kutu+lar)
bunca	(adv)	—	—	
budur	'it's this'	maymundur (maymun+dır)	kutudur	(kutu+dır)
buymuş	'it is alleg-edly this'	maymunmuş (maymun+y+mış)	kutuymuş	(kutu+y+mış)

The forms with *n* appear when what follows is a case or plural affix, or the adverbializing affix *-ca*. The *n* does not appear when there is no affix, or when an affix other than the plural affix, the adverbializing affix, or one of the case affixes follows.[10]

A seemingly straightforward treatment of these facts would be to assume that the underlying form of the pronoun is *bun* and posit a rule deleting the final segment just in case a plural or case or adverbializing affix does not follow.[11]

This, however, is precisely what keçi phonology does not permit. The parser has no way to know, at the point where the *n*-deletion rule would have to apply, whether an affix of a particular category follows or not.

There are, fortunately, other alternatives. Several assumptions regarding the morphemic structure are possible, including the following three (using the accusative as an example):

(a) bun-u
(b) bu-nu
(c) bu-n-u

Under (a), the "pronominal *n*" is assumed to be part of the lexical representation of the root morpheme. It would be deleted sometimes by a phonological rule sensitive to following morphological context. Under (b), it is assumed to be part of the lexical representation of the affix morpheme, and would have to be deleted sometimes by a phonological rule sensitive to preceding morphological context. The third possible assumption is that the *n* represents neither a part of the preceding morpheme, nor a part of the following morpheme, but is rather a morpheme all its own.

Assumptions (a) and (b) both treat the "pronominal *n*" alternation as a phonological phenomenon, in the sense that a deletion rule mediates between a lexical representation containing the *n* and a surface form lacking it. Under assumption (c) the "pronominal *n*" is treated as a buffer morpheme, and where it is allowed to appear can be left entirely to the morphotactics.

A closer look reveals that the deletion rule of treatments (a) and (b) could not be quite so simple as it first appeared, and that for a phonological rule it doesn't have much phonological about it. To begin with, it is clear that the rule cannot just delete any underlying *n* in the prescribed morphological context. The root-final *n* of *maymun* 'monkey' doesn't ever vanish, and there are numerous affixes that begin with *n* and never lose it. Either this disappearing *n* must be represented by some more abstract lexical segment which surfaces as *n* when it hasn't been deleted, or the rule must refer specifically to the morpheme containing it as well as to the preceding or following morpheme. Let us assume the former tack is taken and "pronominal *n*" is represented in lexical forms as *N*, i.e. *bu* has lexical representation *buN*.

The picture gets worse when another ugly fact comes into view. The pattern in the case of the relative *ki* is not exactly the same as that of the pronouns:

bu		kutudaki	(kutu+da+ki 'the one in the box')
bunu	(acc)	kutudakini	
buna	(dat)	kutudakine	
bunda	(loc)	kutudakinde	
bundan	(abl)	kutudakinden	
bunlar	(pl)	kutudakiler	
budur	'it's this'	kutudakidir	(kutu+da+ki+dır)
buymuş	'it's allegedly this'	kutudakiymiş	(kutu+da+ki+y+mış)

The pattern is the same except that the *n* in the case of *ki* appears when a case affix follows, but not when the plural affix follows.

This means that a distinct abstract segment would have to be posited to underly this differently behaving *n*, and a distinct rule to delete it. The *ki* affix could have *kiM* as its lexical representation, and the rule deleting *M* would differ from the rule deleting *N*.

We might suspect by now that the phonology is the wrong place to try to deal with this phenomenon. Consider what may be done if we treat the phenomenon as a matter of morphotactics, as suggested by assumption (c).

The treatment of *ki* is indicated in diagram 1. Notice that *ki* labels a transition from N8 to NK, and that from NK there are three escapes, to N2, to N6, and to N9. From N2 there is no escape except via the transition to N3, which is sanctioned only by the plural affix; the only escape from N6 is the transition to N7, sanctioned only by the buffer morpheme *n*. Every path from N7 to WW is via a case affix. Thus the appearance of *n* with *ki* is governed entirely by the morphotactics.

The appearance of *n* with pronominal stems is treated similarly. A basic stem category (o0), not shown in the diagram, is the category of those pronouns that take pronominal *n*. From this category there are three transitions: one to N2, sanctioned by *n*; one to N6, which is a free jump; and one to N9, also a free jump. The difference between the pronominal stems and the *ki* stems is that the pronominal stems require an overt buffer *n* to sanction the transition to N2, while the *ki* stems get there by a free jump. There are in a sense two different *n*'s in this description, but there are no phonological rules involved at all. It is difficult to compare complexity when the devices used by two treatments are radically different, but the morphotactic solution is certainly not obviously more cumbersome than the "phonological" one.

5. Conclusion

The preceding two sections have illustrated some of the capabilities and limitations of the keçi parser, and exposed differences, which will be apparent to those who care, between it and the KIMMO system (Koskenniemi 1983, Karttunen 1983). Keçi rules, unlike KIMMO rules, are "generative" in the sense that they do things to underlying representations to derive surface representations. There is the possibility for interaction between rules within the domain of a morpheme, allowing for certain kinds of feeding and bleeding relations. On the other hand, keçi rules cannot refer to intermediate stages in the derivation of adjacent morphemes, nor can they refer to any property of following context (beyond the current morpheme) except what is represented in the surface form. As seen in section 4, this restriction forces a morphotactic treatment of certain phenomena which one might otherwise be tempted to treat phonologically.

The restriction on the directionality of morphological conditioning may well turn out to be too strong. One obvious thing to do now is to make a thorough search for potential counterexamples, and consider whether palatable morphotactic alternatives are available. The interesting prediction is that no generalizations will be lost, no needless complexity introduced, if what looks at first like a phonological rule with inward morphological conditioning is interpreted as a matter strictly of morphotactics.

Footnotes

[1]The parser's name is *keçi*, after the UCSC Linguistics Department's unofficial mascot. I will henceforth call it that. The name *keçi* should never be capitalized, but I will defer to standard usage and capitalize it when it begins a sentence.

[2]A treatment of nominal compounding is proposed in Hankamer (ms).

[3]In practice, a graphological representation. The Turkish analyzer takes as input a string of ASCII characters representing the standard orthographic representation of the word, which is very close to a classical phonemic representation.

[4]The parser described here makes no attempt to choose among multiple analyses of an input string. The problem of disambiguation in context is beyond its scope.

[5]In the citation of Turkish forms, I adhere to standard Turkish orthography. The parser is implemented using the following transliterations:

I the Turkish orthographic undotted i (ı)
U u umlaut (ü)
O o umlaut (ö)
C c with cedilla (voiceless palatal affricate) (ç)
S s with cedilla (voiceless palatal spirant) (ş)
G yumuşak ge (ğ)

[6]The representation is simplified in several ways: (a) many minor categories have been omitted; (b) forms and glosses have been omitted except for a few landmarks; (c) no distinction is indicated between free jumps and overt affixes; (d) several complex portions of the network have been omitted entirely.

[7]The convention I use in the representation of complex categories is such that the rule of functional application reads

$$X + X\backslash Y = Y$$

[8]Category N1 is distinguished from category N0 because certain adjectival stems (in particular participles derived from verbs) can function as noun stems, receiving the plural, possessive, and case affixes that ordinary noun stems can receive, but cannot receive the N0\N0 derivational affixes. The category N1 is the union of all the forms contained in category N0 together with forms derived from V3 by affixes in the V3\N1 category. In general, free jumps are used to represent such category merging, as well as to represent optionality of an affix category.

[9]That is, in a suffixing language like Turkish, no phonological rule may be dependent on the morphological structure to the right of the morpheme on which the rule is currently applying. Carstairs (1984) has advanced a similar claim, proposing that morphological conditioning factors always appear closer to the root than their effects.

[10]I have not given the genitive or the possessive forms because the genitive suffix begins with an *n* of its own when attached to a vowel-final stem, and hence would show no contrast, and the possessive affix cannot combine with these pronominal stems for semantic reasons.

[11]Interestingly, no traditional or modern grammarian that I could find appears to have clearly espoused this position. Lewis (1967) adopts essentially the position I take below; Underhill (1976, p. 90) suggests that there are two base forms, *bu* and *bun*, which would lead to a description not involving leftward morphological conditioning if the two forms were assigned to different morphotactic categories, which was clearly his intention.

References

Carstairs, Andrew. 1984. Constraints on Allomorphy in Inflection. Indiana University Linguistics Club.

Hankamer, Jorge. 1984. Turkish Generative Morphology and Morphological Parsing. Paper presented at the Second International Conference on Turkish Linguistics, Istanbul. August 1984.

Hankamer, Jorge. ms. Parsing Turkish Nominal Compounds. University of California, Santa Cruz.

Karttunen, Lauri. 1983. KIMMO: A General Morphological Processor. Texas Linguistic Forum 22:217-228.

Koskenniemi, Kimmo. 1983. Two-level Morphology. U. Helsinki, Department of General Linguistics. Publication No. 11.

Lewis, Geoffrey. 1967. Turkish Grammar. Oxford: Oxford University Press.

Underhill, Robert. 1976. Turkish Grammar. Cambridge, MA: MIT Press.

Stevenson College
University of California, Santa Cruz
Santa Cruz, California 95064