Backwards Phonology

John Bear
Artificial Intelligence Center
SRI International

Abstract

This paper constitutes an investigation into the generative capabilities of two-level phonology with respect to unilevel generative phonological rules. Proponents of two-level phonology have claimed, but not demonstrated, that two-level rules and grammars of two-level rules are reversible and that grammars of unilevel rules are not. This paper makes "reversibility" explicit and demonstrates by means of examples from Tunica and Klamath that two-level phonology does have certain desirable cababilities that are not found in grammars of unilevel rules.

1 Introduction

Since Koskenniemi proposed using two-level phonology in computational morphological analysis in 1983, it has enjoyed considerable popularity [Koskenniemi, 1983]. It seems to be both expressively powerful and computationally tractable. Two-level phonological grammars have been written for a dozen or more languages, and written in a form that is interpretable by a program. One question that arises fairly frequently however, at least in the context of discussion about two-level morphology, is roughly, "Why don't you use *normal* generative phonological rules?" i.e., rules of the type that are taught in elementary linguistics classes. A slightly more positive way to ask the question is, "In what way or ways does Koskenniemi's notion of two-level phonological rule represent a theoretical advance?" This paper addresses that question by extending the notion of unilevel rule system to cope with the same types of phenomena that two-level rule systems were designed to handle, and then contrasting the two different systems.

At the annual meeting of the Linguistic Society of America (LSA) in 1981, Ron Kaplan and Martin Kay presented a paper describing results about equivalences between what they call a cascade of finite-state transducers and a set of normal, ordered phonological rules [Kaplan and Kay, 1981]. At the LSA's 1987 annual meeting, Lauri Karttunen gave a paper attempting to show that, when viewed a certain way, Koskenniemi's two-level rules possess a certain elegance that cannot be ascribed to ordered sets of rules, namely their independence from order per se [Karttunen, 1986].

In spite of Karttunen's paper and Koskenniemi's, and perhaps to some extent because of Kaplan and Kay's paper, it is still not obvious to people who are interested in this field what, if anything, two-level phonology offers that cannot already be found in the linguistic literature under the heading of generative phonology. Koskenniemi has made some claims about grammars of two-level rules being reversible whereas sets of ordered rules are not. However these claims are not backed up by solid argumentation, and the Kaplan and Kay paper seems to argue otherwise.

From a linguistic point of view, there may be good reason to think that people use two different sets of rules or procedures for generation and recognition. From a computational point of view, however, it is interesting to ask, "What needs to be done in order to use the same grammar for generation and recognition; does a single reversible grammar lead to more or less work in terms of writing the grammar and in terms of run-time speed; and finally, does a reversible grammar lead to a more or less elegant presentation of the phenomena?" Another reason for asking about reversibility is to make a comparison of these two rule formalisms possible. The main novelty in Koskenniemi's system is the reversibility of the system, so we may well question what would be necessary to view unilevel rules as reversible.

In short, there are very good reasons for being interested in properties of reversibility, and these properties will serve as the basis for this paper's comparison between the two different types of phonological rule formalisms mentioned above. The discussion here will focus more on concrete examples of generative capacity, and much less on issues of what is involved in building an acceptable linguistic theory. [For more on global concerns of linguistic theory, see, for example, Eliasson, 1985]. The questions addressed here will be, "What assumptions need to be made to use a grammar of unilevel generative rules to do recognition?"

and "How does the resulting combination of grammar plus rules-of-interpretation compare with a two-level style grammar?"

2 Reversibility of Unilevel Rule Systems

The question of grammar reversibility involves two interrelated but separate issues. The first is whether the notational or descriptive devices of a grammar are in general amenable to being reversed, and what is involved in the reversal. The second is whether individual accounts of the phenomena of a particular language are reversible, and, again, if so, what is involved in the reversal.

The remarks in this paper are mainly concerned with the general paradigm of generative phonology, in particular, segmental phonology as is described in elementary texts – e.g., Kenstowicz and Kisseberth (1979), Halle and Clements (1983), Schane (1973), Mohanan (1986) – rather than any particular linguistic theory. The main techniques discussed are rewrite rules, orderings of rules, features, and variables for feature values (e.g., the alpha and beta of assimilation rules). The problems of suprasegmental phonology will be left for another paper.

3 Backwards Rules

I shall start by making explicit what it means to apply a phonological rule in the backwards direction. The basic idea is extremely straightforward and will be, I think, uncontroversial.

$$a \rightarrow b \;/\; \alpha \_ \beta \qquad (1)$$

A rule like the one in (1) transforms the string $/\alpha a \beta/$ into the string $/\alpha b \beta/$. Here α and β are strings of characters over some alphabet, e.g., the phonemes of a language. I take it that such a rule can also be interpreted as mapping the string $/\alpha b \beta/$ into the string $/\alpha a \beta/$, when it is applied backwards.

To take a more linguistically realistic rule, let us consider the simple rule in (2).

$$n \rightarrow \eta \;/\; \_ \; g \qquad (2)$$

From a recognition point of view, this means that if we have the sequence [ŋg] in a surface form of a word, then the underlying sequence could be /n g/. In slightly more general terms, we look for the segment on the right side of the arrow to see whether it appears in the context given in the rule. If so, we can transform that segment into the segment on the left side of the arrow.

4 Obligatory Versus Optional

The rule in (2) says nothing about whether it is optional or obligatory in the backwards direction. Optionality in the backwards direction is entirely independent of optionality in the forward direction. In English the rule in (2) seems to be obligatory in the reverse direction, i.e., every surface [ŋ] seems to come from an underlying /n/. In the forward direction, it does not always apply. This is demonstrated by the pair: co[ŋ]gress vs. co[n]gressional.[1]

In a language that had phonemic /ŋ/ and /n/, the rule might be obligatory in the forward direction and optional in the backward direction.[2] That is, if [ŋ] on the surface can come from either /n/ or /ŋ/, then the rule would necessarily be optional in the reverse direction.

The point here then is that one needs to specify in the grammar not just whether a rule is obligatory or optional in the forward direction, but also whether it is obligatory or optional in the backwards direction.

5 Reversibility and Rule Ordering

The previous example describes the case of a single rule and points out that attention must be paid to whether a rule is optional or obligatory in the backwards direction as well as in the forward direction. The following case of rule ordering shows that there is more to the issue of reversibility than the distinction between "optional" and "obligatory."

There is a beautiful example in the *Problem Book in Phonology* by Halle and Clements (1983) of the elegance of rule ordering. In this section I will show that the device of ordered rules is not generally reversible using their example from Klamath.

The data from Klamath together with five rules are taken from Halle and Clements (1983), who in turn give their source as being *Klamath Grammar* by Barker (1964):

[1] Mohanan (1986) p. 151.

[2] That obligatory rules need not be obligatory when applied in the backwards direction has been pointed out by Ron Kaplan (in a course at the LSA Summer Institute at Stanford, 1987)

$nl \rightarrow ll$
/honli:na/ $\rightarrow$ *holli:na* *'flies along the bank'*

$n\underset{\circ}{l} \rightarrow lh$
/honl̥y/ $\rightarrow$ *holhi* *'flies into'*

$nl' \rightarrow l?$
/honl'a : l'a/ $\rightarrow$ *hol?a : l'a* *'flies into the fire'*

$l\underset{\circ}{l} \rightarrow lh$
/pa : l̥la/ $\rightarrow$ *pa : lha* *'dries on'*

$ll' \rightarrow l?$
/yalyall'i/ $\rightarrow$ *yalyal?i* *'clear'*

Halle and Clements also say that Barker assumes that all phonological rules are unordered and that all rules apply simultaneously to underlying representations to derive surface representations.[3] They then give the following exercise: "Show how Barker's set of rules can be simplified by abandoning these [Barker's] assumptions and assuming that phonological rules apply in order, each rule applying to the output of the preceding rule in the list of ordered rules. Write the rules sufficient to describe the above data, and state the order in which they apply."[4]

The rules that one is supposed to arrive at are roughly these:

$$n \rightarrow l \,/\, \_ \left\{ \begin{array}{c} l' \\ \underset{\circ}{l} \\ l \end{array} \right\} \qquad (3)$$

$$\underset{\circ}{l} \rightarrow h \,/\, l\_ \qquad (4)$$

$$l' \rightarrow ? \,/\, l\_ \qquad (5)$$

The ordering to impose is that Rule (3) applies before Rules (4) and (5), and that Rules (4) and (5) are unordered with respect to each other. The reader can verify that the rules give the correct results when applied in the forward (generative) direction. In the backwards (recognition) direction, the derivations for the five forms are as given below. The rule numbers are superscripted with a minus one to indicate that these rules are inverses of the rules listed above.

holli:na $\rightarrow$ *honli:na*
Rule 3^{-1}

holhi $\rightarrow$ *holl̥i* $\rightarrow$ *honl̥i* [5]
Rule 4^{-1} *Rule* 5^{-1}

hol?a:l'a $\rightarrow$ *holl'a:l'a* $\rightarrow$ *honl'a:l'a*
Rule 5^{-1} *Rule* 3^{-1}

pa:lha $\rightarrow$ *pa:ll̥a* $\rightarrow$ **pa:nl̥a*
Rule 4^{-1} *Rule* 3^{-1}

yalyal?i $\rightarrow$ *yalyall'i* $\rightarrow$ **yalyanl'i*
Rule 5^{-1} *Rule* 3^{-1}

What we see here is that in order to recognize the form *holli:na* correctly, Rule (3) must be obligatory in the reverse direction. However, in order to get the correct results for the forms *pa:lha* and *yalyal?i*, Rule (3) may not apply at all; i.e., it is not correct to say that the results can be obtained by correctly stipulating whether a rule is optional or obligatory. Rule (3) works well in the forward direction, but gives incorrect results when applied in the backwards direction. In short, the elegant set of ordered rules makes incorrect predictions about recognition. In contrast, Barker's original unordered set of rules correctly describes the data regardless of direction of application (i.e., generation vs. recognition).

This is a result about ordering of rules. I have not shown that a set of ordered rules is never reversible, only that such a set is not necessarily reversible.

6 Variables and Deletion

The previous example used extremely plain rules: no features, no alphas or betas, and no deletion. The next example I shall present involves some of these commonly used devices. I shall try to make clear when they can be used in a reversible way (though they need not be), and when they just do not seem amenable to reversal. Before discussing reversal further, I will present the data and the set of rules for describing the data in the generative framework. The data and analysis were taken from Kenstowicz and Kisseberth (1979).[6] Their data come from the language Tunica.

The rules and data deal with two phenomena: vowel assimilation and syncope. The rules, given below, are ordered, with (6) occurring before (7). [Note on transcription: the question mark represents glottal stop.]

[3] Halle and Clements (1983) p. 113
[4] Ibid.
[5] This is correct modulo the change of i back into y which Halle and Clements assure us is not part of the issue at hand. For purposes of discussing reversibility it merely provides more support for the argument that unilevel rules are not easily reversed.
[6] p. 292. They cite their source as Haas (1940).

$$\begin{bmatrix} +syll \\ +low \end{bmatrix} \rightarrow \begin{bmatrix} \alpha \text{ back} \\ \beta \text{ round} \end{bmatrix} / \begin{bmatrix} +syll \\ \alpha \text{ back} \\ \beta \text{ round} \end{bmatrix} \; ? \; \_ \qquad (6)$$

$$\begin{bmatrix} + & syllabic \\ - & stress \end{bmatrix} \rightarrow \emptyset \; / \; \_ \; ? \qquad (7)$$

Rule (7) says (or was meant to say) that unstressed vowels are deleted before glottal stops. Rule (6) was intended to mean that /a/ assimilates to [ɛ] or [ɔ] when it is separated by a glottal stop from a preceding /i/ or /u/ respectively.

In addition to the two rules just given, Kenstowicz and Kisseberth mention but do not formulate a rule of Right Destressing that follows both rules. The rules are in accord with the following data, also taken from Kenstowicz and Kisseberth. The following forms show assimilation.

To verb	He verbs	She verbs	She is v-ing	Gloss
pó	*pó?uhki*	*pó?ɔki*	*póhk?aki*	look
pí	*pí?uhki*	*pí?ɛki*	*píhk?aki*	emerge
yá	*yá?uhki*	*yá?aki*	*yáhk?aki*	do
čú	*čú?uhki*	*čú?ɔki*	*čúhk?aki*	take

These forms show syncope and assimilation.

To verb	He verbs	She verbs	She is v-ing	Gloss
hára	*hár?uhki*	*hár?aki*	*hárahk?áki*	sing
hípu	*híp?uhki*	*hípɔki*	*hípuhk?áki*	dance
náši	*náš?uhki*	*náš?ɛki*	*nášihk?áki*	lead

As a sample derivation, Kenstowicz and Kisseberth give the following:

/náši?áki/
↓ Vowel Assimilation
náši?éki
↓ Syncope
náš?éki
↓ Right Destressing
[*náš?ɛki*]

For the purpose of going through a backwards derivation, I will make explicit a few assumptions. First, I assume that the Vowel Assimilation rule is really as in (8) below.

Vowel Assimilation (Modified)

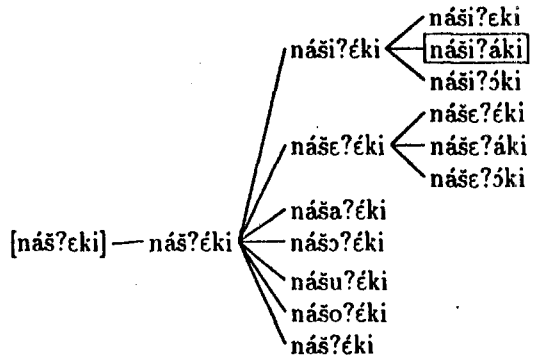
$$\begin{bmatrix} +syll \\ +low \end{bmatrix} \rightarrow \begin{bmatrix} +syll \\ +low \\ \alpha \text{ back} \\ \beta \text{ round} \end{bmatrix} / \begin{bmatrix} +syll \\ \alpha \text{ back} \\ \beta \text{ round} \end{bmatrix} \; ? \_ \qquad (8)$$

It is a matter of style that the features [+ syll, + low] were left out of the feature bundle to the right of the arrow in Kenstowicz and Kisseberth's formulation of the rule. Although it is considered good style to do so, the omission of such information makes it unclear how the rule should be applied for recognition. Hence I have included this information in Rule (8).[7]

Another assumption I will make is that the unformulated rule of Right Destressing lends nothing to my argument here. I assume that the rule when applied in the reverse direction puts stress on the appropriate syllable and nowhere else.[8]

Finally, I will spell out what I consider to be a reasonable interpretation of how to use the rules for recognition. When interpreted backwards, Rule (8) says that a low vowel that is separated by a glottal stop from another vowel with which it agrees in backness and rounding might have come from some other low vowel. The syncope rule in (7), when interpreted backwards, says to insert an unstressed vowel before glottal stops. As was pointed out before, there is no way to deduce whether these rules are obligatory or optional in the reverse direction. Indeed, it is not at all obvious what "obligatory" even means in terms of the assimilation rule taken backwards.

Given these assumptions, we can now produce a reverse derivation for [*na's?ɛki*].



First Reverse Destressing is applied to give *náš?éki*. Then Reverse Syncope applies to insert various hypothesized vowels in forms in the column to the right. Finally, the rightmost column shows the results of

[7] Presumably Kenstowicz and Kisseberth want to treat [ɛ] as being [+ low] to keep the rule simple and still contrast [ɛ] with [i]. If they treat [ɛ] as [- low] and [ɔ] as [+ low], the assimilation rule becomes messier. This assumption about [ɛ] becomes important later.

[8] It seems clear that segmental accounts will fall short when dealing with suprasegmental issues like stress. The goal of this paper is to contrast two different ways of doing segmental phonology. Both would presumably benefit from autosegmental extensions.

applying the reverse of the Assimilation rule to the preceding forms. A box is drawn around the correct underlying form.

What we end up with are 14 or 15 possible forms – clearly too many. One problem is that the assimilation rule in (6) and (8) was formulated with only generation in mind. If we change it slightly, adding the features [+back, -round] to the bundle to the left of the arrow as in (9),

$$\begin{bmatrix} +syll \\ +low \\ +back \\ -round \end{bmatrix} \rightarrow \begin{bmatrix} +syll \\ +low \\ \alpha back \\ \beta round \end{bmatrix} / \begin{bmatrix} +syll \\ \alpha back \\ \beta round \end{bmatrix} \; ? \; \_ \qquad (9)$$

we have a better rule. Now it says that [ɛ] and [ɔ], when they result from assimilation, come specifically from /a/. This makes the results better. The previous version of the rule just mentions low vowels, of which there are three that we know about: ɛ, a, ɔ.[9] When we specify that of these three we always want /a/, we have a more accurate grammar. Now instead of recognizing 14 or 15 possible underlying forms for the word *náš?ɛki*, the grammar only recognizes ten.

There is a very simple but subtle point at issue here, having to do with writing reversible rules. The grammar writers knew when they were formulating the assimilation rule that [ɛ] and [ɔ] were never going to come up *as input to the rule* because these two vowels do not exist in the underlying representations. They also knew that there were no other rules applying before the assimilation rule which would introduce [ɛ] or [ɔ]. Hence they did not need to distinguish between the various possibilities for low vowels. In short, the grammar writers made use of fairly subtle information to write a rule which was as pared down as possible. Leaving out the features in (9), as Kenstowicz and Kisseberth do, looks elegant, but turns the two-way rule into a one-way rule that works only for generation. This is a case where leaving out some features obscures the content of the rule and prevents one from correctly applying the rule for recognition. In short, this is a case where the rule could have been written in a way that was reversible, or at least more reversible, but in the name of "brevity" or "elegance" it was not.

The vowels [ɛ] and [ɔ] also provide complications for the reversal of the vowel deletion rule. We have no reason to believe from the data given that the deleted vowel is ever [ɛ] or [ɔ]. However there is not a good way of saying, using standard rule writing techniques, that any vowel that is introduced in the recognition must be one of the underlying ones. *In ordered sets of rules, there is not typically a distinction made between the segments that can occur as input to a rule and segments that can only occur as output.* One of the unhappy consequences is that [ɛ] and [ɔ] have the same status with respect to the rules of Tunica as the other, underlying, vowels in the language.

An even more serious problem revealed by this Tunica example is the inability of the standard generative rule-writing mechanism to specify the interrelationship between rules. The rules apply based only on strings of characters they get as input, not on what rules came before. In the case at hand, however, we would like to be able to relate the two rules to one another. What we would really like to be able to say is that when in the course of recognition it becomes necessary to reintroduce the deleted vowel, if there is an [ɛ] on the surface the reintroduced vowel must be [i], and if there is an [ɔ] the reintroduced vowel must be [u] or [o]. This is a problem with alpha (assimilation) rules. There is no way to say that if there is an [ɛ] or [ɔ] on the surface, then the reverse of the syncope rule must apply, when doing recognition, and, furthermore, that it must apply in such a way that the assimilation rule can then apply (again in reverse) and, lastly, that the reverse of the assimilation rule *must* then apply. In simpler terms, there is no way to say that if there is an [ɛ] (respectively [ɔ]) on the surface, then it must be preceded by an underlying /i/ (respectively /u/ or /o/).

When dealing with cases of deletion, and mergers in general, it is not generally possible to write a set of rules that maps surface forms unambiguously to a single underlying form. In the case of the Tunica vowel deletion, there are occurrences of surface forms in which the phonological rules cannot tell which vowel to reintroduce when doing recognition. There are, however, cases where it *is* clear which vowel should be reintroduced, e.g., the case above, and in these cases, both the grammar formalism and the individual analysis should be able to express this information. The mechanism of using alphas and betas, for instance in assimilation rules, does not appear to have this expressive capacity.

The problem could be ameliorated by writing less elegant rules. For instance, the syncope rule in (7) could be written as in (10).

$$\begin{bmatrix} +syllabic \\ +underlying \\ -stress \end{bmatrix} \rightarrow \emptyset \; / \; \_ \; ? \qquad (10)$$

This would ensure that the nonunderlying vowels [ɛ] and [ɔ] would not be introduced when applying the rules in the reverse direction. *It still would not be as*

[9] As mentioned in an earlier footnote, Kenstowicz and Kisseberth seem to treat [ɛ] as [+ low].

restrictive as one could be using two-level rules.

One could argue that all one needs to do is use the lexicon to weed out the forms that are wrong. Yet one would not consider suggesting the same thing if a grammar generated too many surface forms, although one could imagine using a surface lexicon as a filter. The technique of using the lexicon to weed out the forms that are wrong is a perfectly good efficiency measure, but has no bearing on the question of how well a formalism maps underlying forms to surface forms and vice versa.

In the rest of this paper I will present and discuss two-level accounts of phonological phenomena described earlier, and show the merits of such an approach.

7 Two-level Rules

In the two-level accounts that have been proposed [Koskenniemi 1983, Karttunen and Wittenburg 1983, Bear 1986, etc.], there are two alphabets of segments, underlying and surface. There are constraint-rules about which underlying segments may be realized as which surface segments, and vice versa, based on context. The rules' contexts are strings of pairs of segments, each underlying segment paired with a surface segment. Deletions and insertions are handled by pairing a segment with a null segment. What is crucial about the rules is that each element of a context is actually a pair of segments, an underlying and a surface segment. The ability to refer to both surface and underlying contexts in a rule allows the rule writer to describe phenomena that are handled with ordered rules in the unilevel approach.

The other powerful device in two-level phonology is an explicit listing of the two alphabets and the feasible mappings between them. These mappings are simply pairs of segments, one surface segment paired with one underlying segment. This list of feasible pairs typically contains many pairs of identical segments such as (a,a) or (b,b), representing that there are segments that are the same underlyingly as on the surface. The list also contains pairs representing change. For the Tunica example, (a,ɛ) and (a,ɔ) would be in the list, but (a,u) and (i,u) for example would not be. The feasible pairs can be thought of as machinery for generating strings of pairs of segments that the rules either accept or reject. An accepted string of segment pairs constitutes a mapping from an underlying form to a surface form *and from surface to underlying form.*

8 Rule Ordering

In a paper presented at the 1986 annual meeting of the Linguistic Society of America, Lauri Karttunen proposed this solution for the Klamath data above:[10]

$$n \rightarrow l \,/\, \_ \begin{Bmatrix} l':= \\ \underset{\circ}{l}:= \\ l:= \end{Bmatrix} \qquad (11)$$

$$\underset{\circ}{l} \rightarrow h \,/\, =:l \,\_ \qquad (12)$$

$$l' \rightarrow ? \,/\, =:l \,\_ \qquad (13)$$

The contexts of the rules should be read as follows. Each pair separated by a colon is a lexical segment followed by a surface segment. The equals sign is a place holder used when the rule writer does not want to make any commitment about what some segment must be. So, for instance, l':= is an underlying /l'/ paired with some surface segment, and the rule doesn't care which. Similarly, =:l is a way of stipulating that there is a surface [l] in the context, and we don't care, for the purposes of this rule, which underlying segment it corresponds to. The right arrow, →, is being used in the way described in Bear [1986, 1988 a,b]. For example, Rule (11) should be construed as allowing the pair of segments n:l (underlying n corresponding to surface l) to occur in the rule's environment, while disallowing the pair n:n. Although the right arrow rule is reminiscent of the arrow in unilevel rules, this interpretation is nondirectional. There are two other kinds of constraints to allow one to deal effectively with the asymmetries involved in pairing underlying forms with surface forms. In Bear [1986, 1988] the two other kinds of constraints are (1) to allow a pair of segments to occur in a certain context without disallowing the default pair (e.g. n:n in the previous example is a default pair), and (2) to disallow a pair in some context without allowing some other pair. For example, the rule types in (14) and (15) are allowed.

$$a{:}b \text{ allowed here: } \alpha \,\_\, \beta \qquad (14)$$

$$a{:}b \text{ disallowed here: } \alpha \,\_\, \beta \qquad (15)$$

In Koskenniemi [1983, 1984] the constraints are slightly different, but have roughly the same functionality. In Koskenniemi's system, one may stipulate that if a lexical segment occurs in some context, then it must correspond to some particular surface segment. One may also stipulate that a certain lexical/surface segment pair may only occur in a certain environment.

[10] I'm using an amalgamation of notations from Koskenniemi, Karttunen and Wittenburg, and Bear.

Karttunen [1986] pointed out that the three rules in (11), (12), and (13) work correctly to give the right results when generating surface forms from underlying forms, and made the point that they do so without recourse to the device of rule ordering. Another point he could have made about these rules which I will make here is that they are just as effective in producing the right underlying forms from surface forms. There is not the problem of multiple intermediate levels of representation, where one is faced with the choice of whether to continue applying [reversed] rules or to stop and call the form a result.

9 Combining Assimilation With Deletion

One solution for the Tunica data is given below.[11]

$$a \rightarrow ɔ \;/\; \{\, u{:=} \mid o{:=} \,\}\; ?\; \_ \qquad (16)$$

$$a \rightarrow \varepsilon \;/\; i{:=}\; ?\; \_ \qquad (17)$$

$$\begin{bmatrix} \text{Vowel} \\ \text{-stress} \end{bmatrix} \rightarrow \text{V} \;/\; \_? \text{ where Vowel} \in \left\{ \begin{matrix} i \\ a \\ o \\ u \end{matrix} \right\} \qquad (18)$$

Rules (16) and (17) say that /a/ assimilates to the *underlying* vowel preceding it, with a glottal stop intervening. One other crucial element of the two-level way of doing things is that in addition to rules, a grammar contains a list of feasible segment pairs. For this Tunica case, there presumably would not be a feasible pair /ɛ/:[ɛ], nor would there be /ɔ/:[ɔ] since [ɛ] and [ɔ] do not seem to occur as underlying vowels. Hence the surface [ɛ] in our example word [*náš?ɛkí*] would be forced unambiguously to correspond to an underlying /a/. This is exactly what we want.

Rule (18) specifies that unstressed vowels are deleted when they occur before a glottal stop. The rule makes clear that only the four vowels i, a, o, and u are deleted, and also that when doing recognition, only those vowels are allowed to be inserted.

These rules make it clear that the underlying form for [*náš?ɛkí*] must be /*náši?áki*/ modulo details of the rule of Right Destressing.

10 Analysis by Synthesis

There is one system for doing computational morphology, specifically for recognizing Turkish, which uses unilevel rules [Hankamer, 1986]. The system first invokes an ad hoc procedure to find the first heavy syllable of a Turkish word. This substring and perhaps a few carefully constructed variants of it are considered as possible stems for the word. Next, based on the morphotactic information about the stem found in the lexicon, assuming one of the possible stems is in the lexicon, several possible suffixes are proposed as possible. A set of phonological rules is applied to the hypothesized underlying forms consisting of stem+suffix. Whichever of them results in a string that matches the input surface form is considered to be right. The process is repeated until the entire string is analyzed.

Since Turkish is exclusively suffixing and has strong phonotactic constraints on what can be a stem, it is possible to write an ad hoc routine to pick the stem out. It remains to be seen how this method of analysis can be made general enough to be applied successfully to other languages. While Hankamer's paper is interesting in its own right, it would be a mistake to construe it as demonstrating anything very general about reversibility of unilevel rule systems.

11 Conclusion

The question has been asked, "What is so good about Koskenniemi's two-level phonology?" The answer is that it allows one to write reversible, nonprocedural descriptions of phonological phenomena with much more accuracy than does the conventional unilevel formalism. The point I have stressed here is the reversibility. From a computational point of view, this represents a step forward. There are no published accounts of reversible grammars written in a unilevel formalism so far as I know and there are many written in two-level rules. Koskenniemi's proposal was made with computation in mind as opposed to linguistic theory. It may, in the long run, have an impact on linguistic theory. It definitely has had a large impact on computational morphology.

Acknowledgements

The bulk of this work was done while I was a visiting scientist at the IBM LILOG project in Stuttgart, Federal Republic of Germany, in the summer of 1988. This work was also made possible by a gift from the System Development Foundation as part of a coordinated research effort with the Center for the Study of Language and Information, Stanford University. I would like to thank the people at IBM, Stuttgart, SRI, and CSLI for supporting this work. I would also like

[11] It is a common abbreviatory convention that any pair of idendical segments, e.g., a:a, can be written simply as a single segment, e.g., a. So, in these rules the glottal stop character represents the pair: ?:?.

to thank the following people for many helpful discussions and comments: Meg Withgott, Martin Emele, Mary Dalrymple, Petra Steffens, Bob Mugele, and Hans Uszkoreit.

I would not have been able to produce this paper had it not been for Emma Pease who has done considerable work defining phonetic fonts and graphics macros for TEX which she made available. I would also like to thank Mary Dalrymple for helping me with LATEX.

References

[1] Barker, M.A.R. (1964) *Klamath Grammar*, University of California Press, Berkeley and Los Angeles, California.

[2] Bear, John (1985) "Interpreting Two-Level Rules Directly," presented at a Stanford workshop on finite-state morphology.

[3] Bear, John (1986) "A Morphological Recognizer with Syntactic and Phonological Rules," *COLING 86*, pp. 272-276.

[4] Bear, John (1988) "Two-Level Rules and Negative Rule Features," *COLING 88*, pp. 28-31.

[5] Eliasson, Stig (1985) "Turkish k-Deletion: Simplicity vs. Retrieval," in *Folia Linguistica XIX*, 3-4, pp. 289-311, Mouton Publishers, The Hague.

[6] Gazdar, Gerald (1985) "Finite State Morphology: A Review of Koskenniemi (1983)," Technical Report No. CSLI-85-32 of the Center for the Study of Language and Information, Stanford University, Stanford, California.

[7] Haas, Mary (1940) *Tunica. Handbook of American Indian Languages*, Vol. 4. Smithsonian Institution, Bureau of American Ethnography, Washington, D.C.

[8] Halle, Morris, and G.N. Clements (1983) *Problem Book in Phonology: A Workbook for Introductory Courses in Linguistics and in Modern Phonology*, The MIT Press, Cambridge, Massachusetts, and London, England.

[9] Hankamer, Jorge (1986) "Finite State Morphology and Left-to-Right Phonology," in *Proceedings of the West Coast Conference on Formal Linguistics*, published by Stanford Linguistics Association, Stanford, California.

[10] Kaplan, Ronald, and Martin Kay (1981) Paper presented at the annual meeting of the Linguistic Society of America.

[11] Karttunen, Lauri (1983) "Kimmo: A General Morphological Processor," in *Texas Linguistic Forum #22*, Dalrymple et al., eds., Linguistics Department, University of Texas, Austin, Texas.

[12] Karttunen, Lauri (1986) "Compilation of Two-Level Phonological Rules," presented at the Annual Meeting of the Linguistic Society of America in San Francisco, California.

[13] Karttunen, Lauri, Kimmo Koskenniemi and Ronald Kaplan (1987) "TWOL: A Compiler for Two-Level Phonological Rules," distributed at the 1987 Summer Linguistic Institute at Stanford University, Stanford, California.

[14] Karttunen, Lauri and Kent Wittenburg (1983) "A Two-Level Morphological Analysis Of English," in *Texas Linguistic Forum #22*, Dalrymple et al., eds., Linguistics Department, University of Texas, Austin, Texas.

[15] Kay, Martin (1983) "When Meta-rules are not Meta-rules," in K. Sparck-Jones, and Y. Wilks, eds. *Automatic Natural Language Processing*, John Wiley and Sons, New York, New York.

[16] Kay, Martin (1987) "Nonconcatenative Finite-State Morphology," paper presented at a workshop on Arabic Morphology, Stanford University, Stanford, California.

[17] Kennstowicz, Michael, and Charles Kisseberth (1979) *Generative Phonology*, Academic Press, Inc., Harcourt, Brace, Jovanovich, Publishers, Orlando, San Diego, New York, Austin, Boston, London, Sydney, Tokyo, Toronto.

[18] Koskenniemi, Kimmo (1983) *Two-Level Morphology: A General Computational Model for Word-form Recognition and Production.* Publication No. 11 of the University of Helsinki Department of General Linguistics, Helsinki, Finland.

[19] Koskenniemi, Kimmo (1983) "Two-Level Model for Morphological Analysis," *IJCAI 83*, pp. 683-685.

[20] Koskenniemi, Kimmo (1984) "A General Computational Model for Word-form Recognition and Production," *COLING 84*, pp. 178-181.

[21] Mohanan, K.P. (1987) *A Theory of Lexical Phonology*, D. Reidel Publishing Company, Dordrecht, Holland.

[22] Schane, Sanford (1973) *Generative Phonology*, Prentice Hall, Englewood Cliffs, New Jersey.