A NOTE ON MORPHEME STRUCTURE IN GENERATIVE PHONOLOGY

Frances Karttunen
The University of Texas at Austin
Austin, Texas U.S.A. 78712

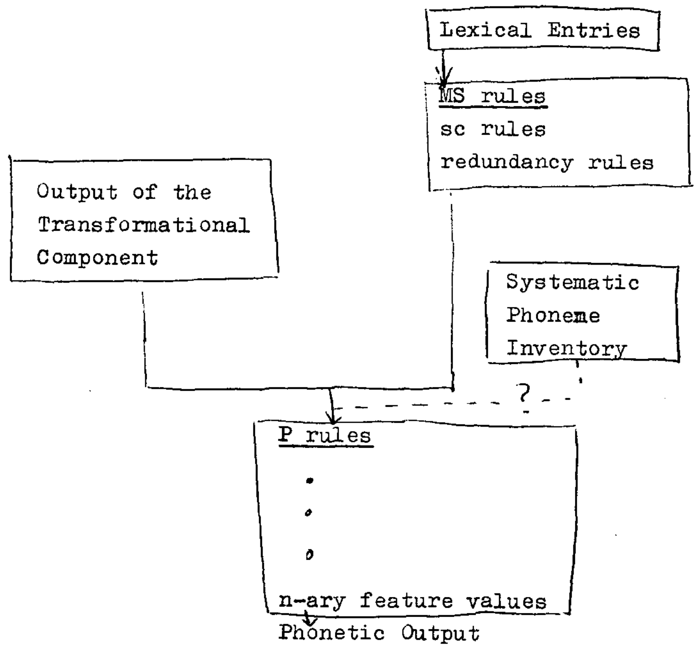
In an early model of generative phonology the lexicon of a language contained entries with as few feature specifications as possible in the interest of economy. The blank feature specifications representing both nondistinctive features and those rendered redundant by sequential constraints were filled in by the same phonological rules. At this point, the concept of P rules changing feature values was unclear.

When the distinction between rules that fill in blanks and those that change feature values became clear, it was embodied in the concept of morpheme structure rules and P rules. The MS rules were further split into feature redundancy (segment structure) rules and sequential constraint rules. The MS component bore a striking resemblence to the earlier "phonotactic" sections of autonomous phonemic analyses, but the claim was made for MS rules that they explained what phonotactics merely described. The MS rules formed a major part of Chomsky's "readjustment component" which rendered the output of the syntactic component fit to be the input to the phonological component. A fairly current version of this model is the following one from Harms' Introduction to Phonological Theory:
(1968)

(1) Sequential constraint rules and blank filling rules fill in redundant features in lexical items.
(2) These lexical items are inserted into the output of the transformational component.
(3) Phonological rules, some utilizing syntactic information, operate on these strings.

This model takes the lexical entry to be a prime of sorts. Within the lexical entry, segmental features are not specified if they can be predicted. If all obstruents in a language are voiceless, voice is specified for the obstruents in all lexical items by the redundancy rule [+obs] → [-vc] . If the only possible initial consonant in a triple cluster is s, then only [+cons] is given in the lexical entry. The rest of the distinctive features for s are filled in by a sequantial constraint rule and the additional nondistinctive ones by a redundancy rule or rules. Harms' model orders all redundancy rules before the phonological rules. This ordering excludes Halle's solution for the exception of i and e from Finnish vowel harmony, which places the harmony rule before the redundancy rule specifying gravity for i and e. ("On the Bases of Phonology", The Structure of Language, 1964, p. 332)

The following diagram represents Harms' model.


Lexical Entries
MS rules
sc rules
redundancy rules
Output of the
Transformational
Component
Systematic
Phoneme
Inventory
?
P rules
n-ary feature values
Phonetic Output

Grouped together as morpheme structure rules (MS rules), sequantial constraint rules and redundancy rules, according to proponents of this model, account for the acceptability of some strings as possible morphemes in a language and the rejection of others.

If we consider the problem of the acceptability of strings of a language's sequential phonemes (English has θ and l, but θlin is not possible.), then we might justifiably characterize morpheme structure rules as generating all and only the possible morphemes (stems, affixes, and uninflected particles) of the language. Is this equivalent to the function of the MS rules in the model above?

There seems to be the following conceptual difference. The model described starts with the lexical entry, and the sequential constraint rules fill in distinctive features. The inventory of systematic phonemes is realized in the output of the SC rules.

It is not clear where or how the inventory of a language's systematic phonemes fits into this model. The fully specified phonemes are realized in the output of the MS rules. The lexical entries in some sense preexist the phoneme inventory, because they consist of incomplete feature matrices which are filled in by the MS rules. How does one arrive at the existing and possible lexical items? To effect a saving of features in the lexicon, all features that can be determined from context are left out of an entry. Then MS rules fill them in. In this way, according to Harms, "Morpheme structure rules can account for the fact that native speakers of a language agree with great consistency on which of several nonoccurring forms could be admitted as new morphemes in their language." (Intro. to Phonol. Th., p. 88)

Is it possible that this phonological model and its MS rules can account for all and only the acceptable morphemes in a language? If the MS rules are only blank-filling rules, a completely new phoneme can be added at will to a language via this model by simply introducing it fully specified into a lexical entry before the MS rules.

If, on the other hand, the MS rules are able to reverse features in lexical entries, then ill-formed entries will be corrected. In this case, the lexicon may be full of impossible morphemes, and the MS rules act as a filter to pass only well-formed morphemes on to the phonological rules.

The argument against this is that the lexicon must be as economical as possible. No features, right or wrong, which are predictable by rule, are specified in the lexicon.

Where do the lexical items come from then? And since the model under discussion applies rules to items from the lexicon, not to strings in general, what is the status of the acceptable strings which do not, at a given moment, exist in the lexicon? A test for acceptability of a hypothetical string might be whether the MS rules could fully specify its matrices without changing any specified features. If blanks remained or features had to be changed, then the string would be judged unacceptable in the language. This test would require that not only existing lexical entries but also hypothetical strings of feature matrices be input to the MS rules. To whatever degree the model rejects or changes faulty inout, it is clearly an acceptor or filter rather than a generator.

Unless the lexicon-MS component relationship is completely circular (take out of the morphemes what can be put in by the MS rules, list the remains in the lexicon, and then fill them in again by the MS rules), there is no way to account for the well-formedness of the input to the MS rules.

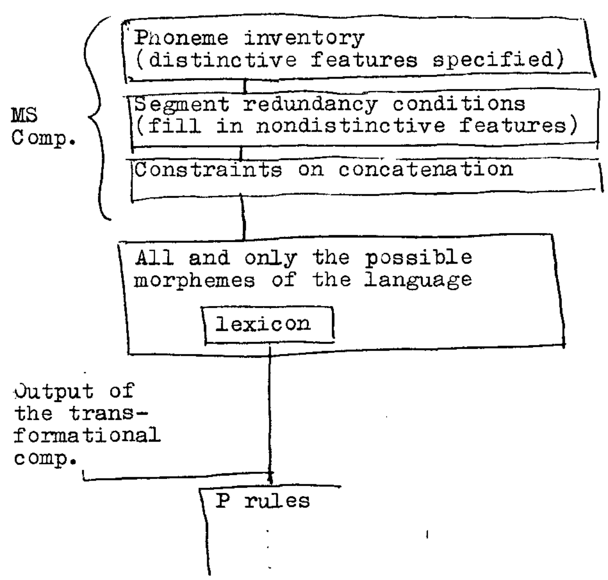
In his elegant article, "Redundancy rules in phonology", Language 43.2, 1967, Richard Stanley clearly demonstrated the different nature of redundancy and P rules (the former predicting feature values, the latter changing them) and the danger of misusing feature blanks. His proposal was that phonological redundancy be embodied not in rules but in Morpheme Structure Conditions. The former had only the lexicon as their domain and were ordered before the P rules. The latter have all matrices in their domain and are not ordered with respect to the P rules.

To quote from Stanley:

... A grammar of each natural language will have, in place of a set of MS rules, an unordered finite set M of MS conditions. This set will include, in general, conditions of each of the three types. The set of all matrices m in U, such that m is accepted by <u>every</u> MS condition in M, is well defined; we call this set M(U).

Since each MS condition in M represents a generalization about the morphemes of the language, it follows that the set M(U) represents all matrices which violate none of these generalizations. .. In short, the set M(U) is exactly the set of possible morphemes of the language. The segment structure conditions in M will guarantee that M(U) contains only those matrices whose columns are systematic phonemes of the language; the sequence structure conditions in M will guarantee that no sequential constraints of the language are violated in matrices of M(U). The set M of MS conditions may thus be thought of as filtering out, from the set U of all matrices, those matrices which do not form a possible morphemes of the language, leaving the set M(U). (Language 43.2 p. 428)

An alternative model of the phonological component, differing from Stanley's in that it views the MS component as a generator rather than as a filter, assumes the language's fully specified systematic phonemes as primitives. The MS conditions are then constraints upon concatenation of submatrices of these matricies of phonetic features.



The MS component of this model generates all and only the possible morphemes of the language. Of these, only a subset are existing lexical items associated with syntactic and semantic information.

Given a hypothetical string, its acceptability can be determined on two criteria. First, is it composed of phonemes of the language? Check the inventory. If so, are the phonemes concatenated in a manner permitted by the MS conditions? If these conditions are met, it may be a lexical item. Is it associated with semantic and syntactic information?

The practical use of morpheme structure conditions in computation brings to mind an old example of the commercial aspects of phonotactics, namely to generate brand names for detergents, beauty products, etc. More seriously, it is desirable to have a system report immediately a recognizable misprint or foreign word rather than to search fruitlessly through dictionary storage for the item. Moreover, since dictionary storage must be continually updated, it is important for a system to report possible new lexical items for inclusion.