PHONOLOGICAL ANALYSIS BY COMPUTER:
PROSPECTS AND DIRECTION[*]

by

Lee A. Becker[**]
Computer Science Department
Indiana University
Bloomington, Indiana 47405

TECHNICAL REPORT No. 121

PHONOLOGICAL ANALYSIS BY COMPUTER:
PROSPECTS AND DIRECTION

LEE A. BECKER

DECEMBER 1981

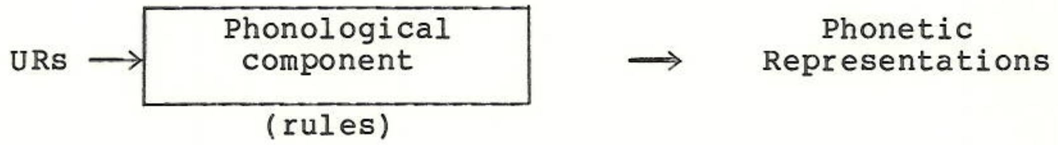
---

[*] This material is based on work supported in part by the National Science Foundation under Grant MCS 81-02291.

[**] Author's present address, Tulane University.

INTRODUCTION

In the generative model of grammar, the function of the phonological component is to assign a phonetic representation to an utterance by modifying the underlying representations (URs) of its constituent morphemes. Morphemes are the minimal meaning units of language, e.g. mean+ing+ful. URs are abstract entities which contain the idiosyncratic information about pronunciations of morphemes.



Phonological analysis attempts to determine the URs and to discover the general principles or rules that relate them to the phonetic representations.

The purpose of this paper is to discuss prospects and directions for developing an intelligent system which does phonological analysis. The structure of the paper will be as follows. First, the potential value of the endeavor will be considered. Second, reasons for the lack of previous work in this area will be explored. Next, a pilot study done by the author will be outlined and the prospects of its approach will be examined in light of other work in learning and grammatical inference. Finally, the particular approach proposed will be justified.

This work would be of potential theoretical interest to the general study of learning systems and of grammatical inference. Its primary application within Artificial Intelligence would be in speech synthesis, which uses the results of phonological analysis. It would also be relevant to linguistic theory especially in connection with the comparison of various versions of Generative Phonology which propose different constraints on the class of possible phonological components and the way in which these constraints relate to learnability. It has a clear connection with work on the acquisition of language by children, and also relates to work in Cognitive Psychology on learning in general.

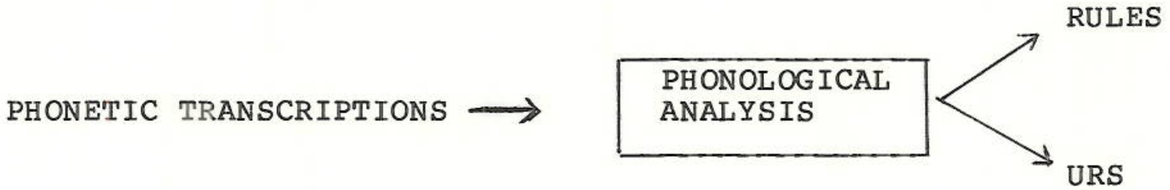
REASONS FOR LACK OF RESEARCH IN THIS AREA

The author knows of no other researchers at the present time who are working on phonological analysis by computer; some work was done by Petrick in 1963 [16]. One might suggest two factors which may have contributed to this--the first peculiar to phonology, the second pertaining to linguistics in general. The first factor is the stigma connected with the term "discovery procedures" which for phonologists is associated with the discovery procedures of the American Structuralists, the predominant school of linguistics in the United States prior to the Chomskyan revolution. The discovery procedures proposed by these linguists disallowed "mixing of levels", e.g., the use of information about syntax and semantics in doing phonology. The interdependence of levels and the necessity for grammatical prerequisites to phonological analysis is a basic tenet of Generative Phonology.

The second factor can be attributed to Chomsky's writings. In Aspects of the theory of syntax (1965) regarded the construction of a linguistic theory that could select a descriptively adequate grammar on the basis of primary linguistic data as unachievable at that time. This property of a theory he referred to as "explanatory adequacy." Chomsky recognized [6, p. 25], "As a precondition for language learning, he [the child] must possess, first, a linguistic theory that specifies the form of the grammar of a possible human language, and, second, a strategy for selecting a grammar of the appropriate form that is compatible with the primary linguistic data." Chomsky continued his efforts to define the form of the grammar, i.e. the class of possible grammars. He translated the second precondition into a requirement for the child to have a method for selecting one grammar out of those within the class of possible grammars that are compatible with the primary linguistic data. Corresponding to this, he suggests [6, p. 31] that a linguistic theory aiming for explanatory adequacy must have a way of evaluating alternative proposed grammars. Instead of a learning strategy or procedure, an evaluation procedure was sought.

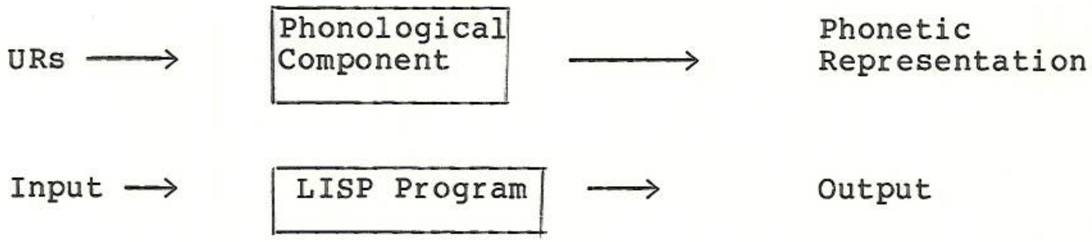
PHONY - A PILOT STUDY

Phonological analysis proper may be considered to take as its input phonetic transcriptions of pronunciations of words and phrases upon which a preliminary morphological analysis has been done. They have been divided into morphenes, and different instances of the same morphene have been associated. (This is done partly on the basis of meaning.)



The output of phonological analyses is a set of rules or regularities in the data, as well as a set of 'underlying representations' for the morphemes. The phonological rules generate the various pronunciations of the morphenes from their underlying representations.

The task of phonological analyses can be contrasted to that of inducing LISP-programs from input-output pairs [18].



Phonological analyses essentially takes only the output and must infer both the input(URs) and set of functions (phonological rules). This task is possible because of (1) the very restricted type of rules possible and (2) a constraint of the 'distance' between a UR and its pronunciations. To understand these it is necessary to consider the representation of sounds and rules in Generative Phonology.

In Generative Phonology sounds are represented as matrices of feature specifications, the phonetic symbols being a shorthand for these matrices. For example, the phonetic symbol <u>d</u> is shorthand for the following set of feature specifications.

$$\begin{bmatrix} -\text{ syllabic} \\ +\text{ consonantal} \\ -\text{ continuant} \\ +\text{ voice} \\ -\text{ nasal} \\ +\text{ anterior} \\ +\text{ coronal} \\ \vdots \end{bmatrix}$$

The set of 'distinctive features' proposed by Chomsky and Halle were claimed to be sufficient to distinguish the sounds in any language. Further these features were all claimed to have two values; the feature was either present or absent. There has been a fair amount of dispute about the specific features, and several additional ones have been proposed, e.g. gravity, advanced tongue root. There has also been considerable dispute about whether the features are all binary. Nevertheless most phonologists use the original binary features, often with a few additional ones. Phonological rules are operations upon sets of these feature matrices by which feature specifications are assigned to the matrix when it appears in a certain context. The rule expressed (in shorthand) normally as s --> š / _i (read s becomes š in position immediately before i) would be expressed as follows using feature matrices.

$$\begin{bmatrix} +\text{ continuant} \\ +\text{ coronal} \\ +\text{ anterior} \\ +\text{ strident} \end{bmatrix} \longrightarrow \begin{bmatrix} -\text{ anterior} \\ +\text{ high} \end{bmatrix} \Big/ \;\_\_\_\; \begin{bmatrix} +\text{ syllabic} \\ +\text{ high} \\ -\text{ back} \end{bmatrix}$$

The representation provides a language in which to express hypotheses. The task is to find statements in this language to express the data. Thus the representation implicitly defines the search space. The search space is restricted by the following constraint on the 'distance' between a UR and its pronunciations. Every feature specification in the UR must be present in a example, a morpheme that has three pronunciations [sarap], [sarab], [sarav]. This constraint restricts its possible URs to /sarap/, /sarab/, /sarav/, /saraf/. Even [f] does not appear in any of the pronunciations of this morpheme, its -continuant specification occurs in [v] and its -voice specification occurs in [p]; its other feature specifications are common to [p], [b], [v]. This constraint is weaker than the "strong alternation condition" (cf. [14]), which would restrict the final UR segment to be /p/, /b/, or /v/. The term "alternation" will be important to the discussion below; here [p] vs. [b] vs. [v] is an alternation.

Given, then the pronunciations # ad +a# and # at #, where '#' stands for word boundary and '+' stands for morphene boundary, the UR must be /ad/ or /at/ (since d and t differ by only a single feature VOICE. The structural change of the single rule necessary to account for this alternation must be either --> [-voice] or --> [+voice], and the environment of the rule must be a subset of the feature specifications in these two pronunciations, i.e.

$$\begin{bmatrix} \text{feature} \\ \text{specifications} \\ \text{of} \\ \text{d} \end{bmatrix} \text{-->} \quad [-\text{voice}] \Big/ \begin{bmatrix} \text{feature} \\ \text{specifications} \\ \text{of} \\ \text{a} \end{bmatrix} \text{---} \begin{bmatrix} \text{feature} \\ \text{specification} \\ \text{of} \\ \text{a} \end{bmatrix}$$

or

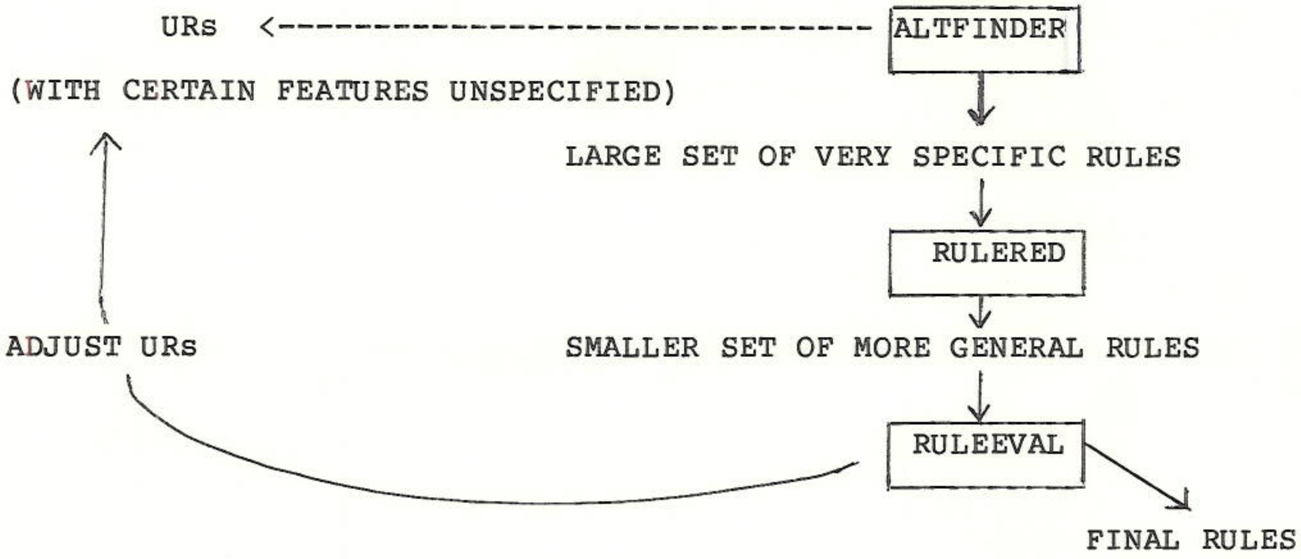
$$\begin{bmatrix} \text{feature} \\ \text{specifications} \\ \text{of} \\ \text{t} \end{bmatrix} \text{-->} \quad [+\text{voice}] \Big/ \begin{bmatrix} \text{feature} \\ \text{specifications} \\ \text{of} \\ \text{a} \end{bmatrix} \text{---} \begin{bmatrix} \text{feature} \\ \text{specifications} \\ \text{of} \\ \text{a} \end{bmatrix}$$

It should be pointed out that most often several sets of combinations of underlying representations and phonological rules can be used to derive the same pronunciations. This could happen in several ways. It could be unclear what the UR is, and different URs together with different rules could derive that same pronunciation, i.e. the directionality of the rule could be unclear. Also some phenomena could be explained by a single more general rule or by several more specific rules.

In Generative Phonology two approaches have been taken to deal with the problem of multiple possible solutions--attempting to impose restrictions on what could constitute a valid solution, and using an evaluation procedure to decide in cases of multiple possible solutions. One could also use both of these approaches. In that case the more restriction, the less evaluation is necessary. No particular proposed restrictions have been adopted by the vast majority of phonologists other than the constraint on the 'distance' between the UR and its pronunciations (as discussed above). An original single evaluation criterion--'simplicity', as manifested by the number of feature specifications used, has not proved completely satisfactory [12]. Probably the most commonly used criterion aside from simplicity is 'naturalness'. 'Naturalness' is primarily a function of cross-linguistic commonness, which is most often related to phonetic considerations. This is not always the case.

To demonstrate the possibility of doing phonological analysis by computer the author designed a program called PHONY. In order to discuss possible extensions to PHONY, as well as the limitations of the approach it embodies, its structure will be considered in some detail.

The structure of the bare machine of PHONY reflects the process of inductive inference and scientific hypothesis formation in general. First, hypotheses or generating principles, the "rules", are proposed as the basis of the data. Second, the rules are generalized with testing to eliminate those which are invalid. Third, the remaining rules are evaluated. As the figure below indicates, in PHONY these three parts are called ALTFINDER, RULERED, RULEEVAL.



ALTFINDER generates the rules. For every feature of a segment of a morpheme that alternates, ALTFINDER produces two lists - one with the strings where a positive value for that feature occurred and the other where a negative value occurred.

It should be noted that the elements of these lists, i.e. strings, together with the feature alternating, its value, and an indication of which segment in the string contains the feature, are all potentially rules. They bear the same information as standard phonological rules. Compare the representations below.

```
#    a    d    +    a    #             #    a    t    #

1                        1             1              1
0              1         0             0              0
0              0         0             0              0
0              0         0             0              0
0              0         0             0              0
0    1         0    1    0             0    1         0
0    0    1    0    0    0             0    0    1    0
0    1    0    0    1    0             0    1    0    0
0    0    0    0    0    0             0    0    0    0
0    1    0    0    1    0             0    1    0    0
0    1    0    0    1    0             0    1    0    0
0    0    1    0    0    0             0    0    1    0
0    1    1    0    1    0   VOICE     0    1    0    0
0    1    0    0    1    0             0    1    0    0
0    0    0    0    0    0             0    0    0    0
0    0    0    0    0    0             0    0    0    0
0    0    0    0    0    0             0    0    0    0
0    0    0    0    0    0             0    0    0    0
```

to the rules t --> d / # a _ + a # d -> t / # a _ #, i.e. respectively, one can't pronounce t in the environment # a _ + a # but rather must pronounce d, and one can't pronounce d in the environment # a _ # but rather must pronounce t. The latter rule and the second representation are juxtaposed below.

```
    1000000000000000 1010110110000 1000011000000 1000000000000000

D --> T /     #              a              ___              #
```

It is often the case that one or both of these potential 'rules' will be valid, i.e. would be generalizations that would hold over the pronunciations represented in the input. However, these 'rules' would be much less general than those which are found in phonological analyses. It is assumed that speaker/hearer/language learners can and do generalize from these specific cases to form more general rules. If this were not the case how could speakers correctly pronounce morphemes in new environments? Within the theory the criterion of simplicity is sensitive to these generalizations in that such generalizations reduce the number of feature specifications. Within PHONY the preference for more general rules is manifested repeatedly in RULERED by trying to generate and test more general rules resulting from the coalescing or combining of two or more specific rules.

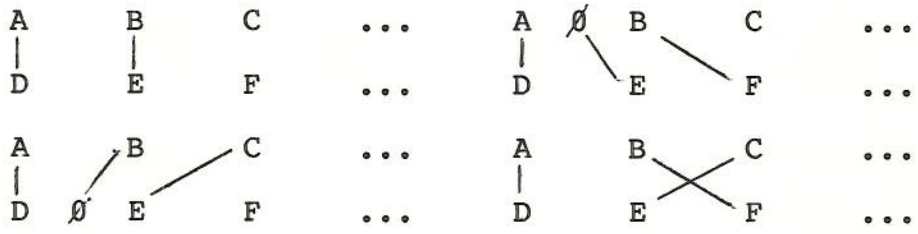
Recall that the representation of the segments involved a feature matrix with positive or negative specifications for each feature. In order to generate more general rules this representation is modified to two matrices for each segment - one representing those features which must be positive in the environment and the other for those features which must be negative. (Other features can assume either value.)

ALTFINDER also sets up URs with an indication of which feature specifications are alternating and need to be decided. ALTFINDER proceeds by comparing in turn each instance of a given morpheme with the current hypothesized UR for that morpheme. The identification and isolation of alternations is complicated by the common processes of epenthesis (insertion of a segment) and elision (deletion of a segment), and occasionally by the much more rarely occurring methathesis (interchange in the positions of two segments). These procedures are illustrated below.

Given UR / t a r i s k /, [tarisak] would involve Epenthesis 0 --> a [trisk] would involve Elision a --> 0 [tariks] would involve Methathesis sk --> ks

Therefore in cases where the segments being compared are not identical it is necessary to ascertain whether they are variants of a single underlying segment of one of these processes has applied. The possibilities are illustrated below.

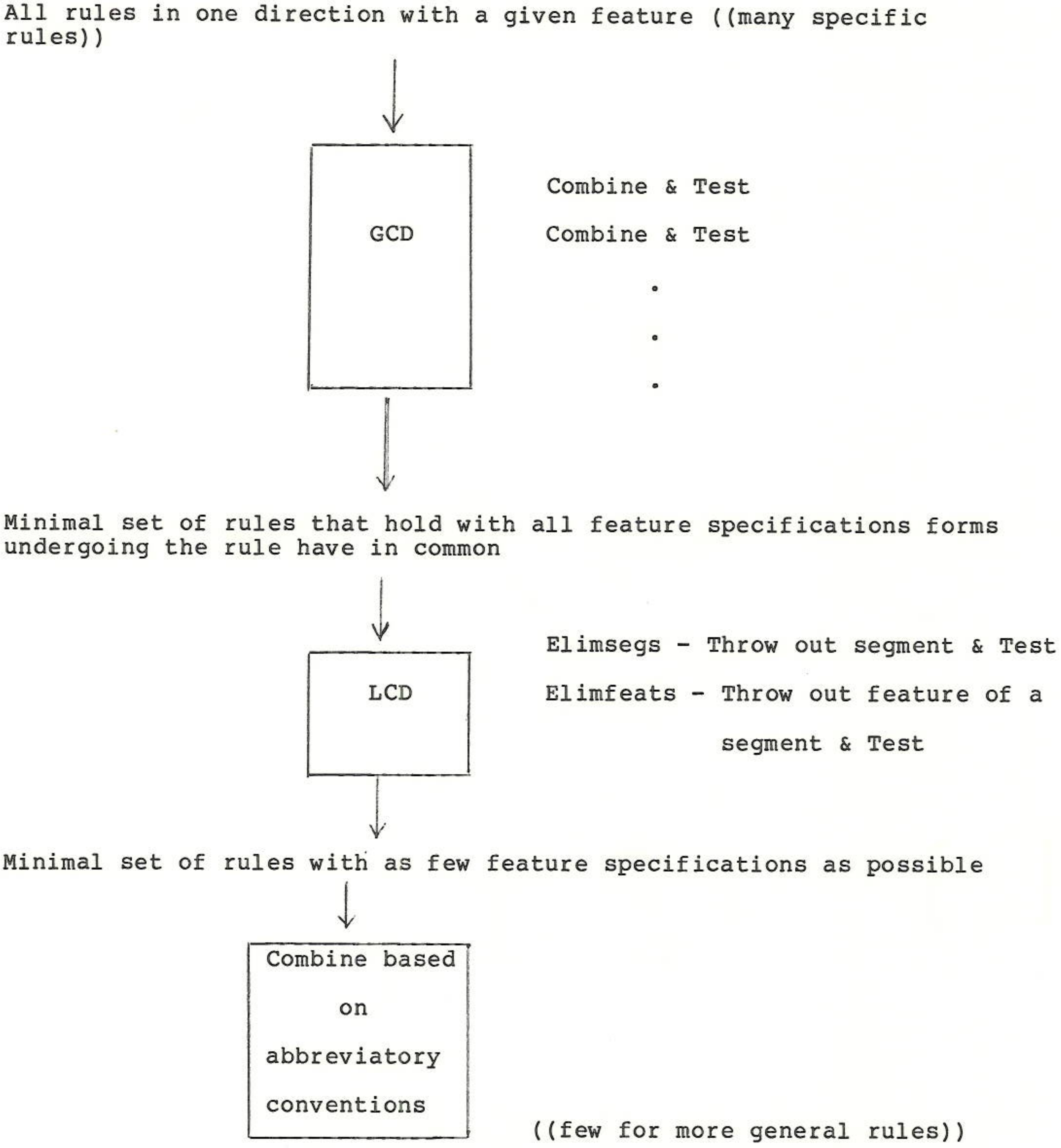
Given two pronunciations of the same morpheme [A B C . . .] where A is associated with D [D E F . . .] where B is not identical to E. there are four possible relationships:



The criteria used to decide between these relationships are (a) degree of similarity in each of the conceivable associations, and (b) a measure of the similarity of the rest of the strings for each of the conceivable associations.

RULERED attempts to generalize the rules. Its structure is represented below.

RULERED



All rules in one direction with a given feature ((many specific rules))

Combine & Test

Combine & Test

.
.
.

Minimal set of rules that hold with all feature specifications forms undergoing the rule have in common

Elimsegs - Throw out segment & Test

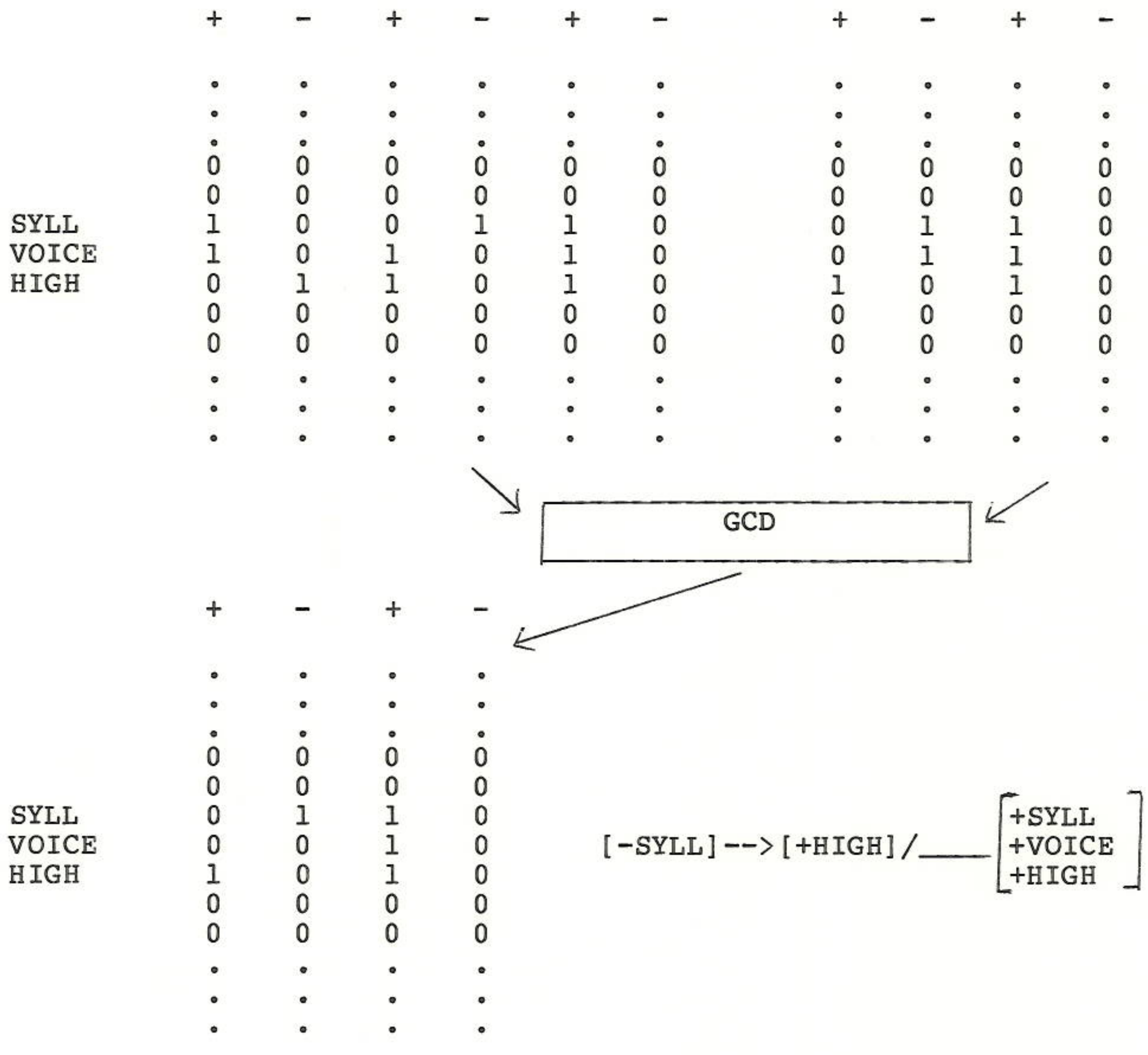
Elimfeats - Throw out feature of a segment & Test

Minimal set of rules with as few feature specifications as possible

Combine based on abbreviatory conventions

((few for more general rules))

The first process in RULERED involves going through all the rules involving a certain feature and generating the minimal number of equivalence classes of "rules" and combined (GCDed) "rules" which are valid. The GCD (greatest common denominator) operation is essentially an intersection of matrices made up of linear order of vectors (feature matrices) of the segments of two rules, lined up on the basis of the segment with the alternation.



$$
\begin{array}{l|cccccc}
 & + & - & + & - & + & - \\
 & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
 & 0 & 0 & 0 & 0 & 0 & 0 \\
 & 0 & 0 & 0 & 0 & 0 & 0 \\
\text{SYLL} & 1 & 0 & 0 & 1 & 1 & 0 \\
\text{VOICE} & 1 & 0 & 1 & 0 & 1 & 0 \\
\text{HIGH} & 0 & 1 & 1 & 0 & 1 & 0 \\
 & 0 & 0 & 0 & 0 & 0 & 0 \\
 & 0 & 0 & 0 & 0 & 0 & 0 \\
 & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots
\end{array}
\qquad
\begin{array}{cccc}
+ & - & + & - \\
\vdots & \vdots & \vdots & \vdots \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & 1 & 1 & 0 \\
0 & 1 & 1 & 0 \\
1 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
\vdots & \vdots & \vdots & \vdots
\end{array}
$$

$$
\begin{array}{l|cccc}
 & + & - & + & - \\
 & \vdots & \vdots & \vdots & \vdots \\
 & 0 & 0 & 0 & 0 \\
 & 0 & 0 & 0 & 0 \\
\text{SYLL} & 0 & 1 & 1 & 0 \\
\text{VOICE} & 0 & 0 & 1 & 0 \\
\text{HIGH} & 1 & 0 & 1 & 0 \\
 & 0 & 0 & 0 & 0 \\
 & 0 & 0 & 0 & 0 \\
 & \vdots & \vdots & \vdots & \vdots
\end{array}
$$

$$[-\text{SYLL}] \dashrightarrow [+\text{HIGH}] / \underline{\qquad} \begin{bmatrix} +\text{SYLL} \\ +\text{VOICE} \\ +\text{HIGH} \end{bmatrix}$$

The GCD operation has generated a more general rule. If the original two rules are a manifestation of a more general rule, the generalized rule must not involve or make reference to the initial segment of the former rule. Notice also that in the GCD the VOICE feature does not have to be positive or negative; if the two original rules are a manifestation of a single rule the specification of the VOICE feature in the alternating segment must not be relevant.

The rules resulting from the first process in RULERED have the largest matrices, i.e. the largest set of feature specifications which all the forms undergoing these rules have in common. (These are similar to Vere's "maximal common generalizations" [19].) However, the elimination of some of these feature specifications might still result in valid rules. The so-called consistent with the input corpus [13, p. 336]. Rules with minimal matrices, i.e. minimal number of feature specifications are produced by LCD (lowest common denominator) attempting in turn to eliminate each segment in GCDed rule; the new rule is generated and tested, and if valid the segment is out, otherwise it remains. Then an attempt is made to eliminate in turn each feature specification in the remaining segments, again generate and test.

The output of LCD is the minimal set of rules with as few specifications as possible with the rules still holding over the input corpus. The third part of RULERED involves combining the remaining rules, where possible on the basis of the abbreviatory conventions of Generative Phonology. This is done on the basis of the formal properties of the rules. For example, if two generated rules are identical except that one has an additional segment not present in the other, these can be into a single rule; parentheses allow the inclusion of optional segments in the environment of a rule. For example, [-son]-->[-voice]/___# and [-son]-->[-voice]/___[+cons]# could be combined as [-son]-->[-voice]/___([+cons])#. In addition, all the rules generated above involve a change of only a single feature specification. If there are several rules which are identical except that a different feature specification is changed, i.e. the two changes occur in the same environment, they can be combined into a single rule; in this particular environment both specifications change.

RULEEVAL takes as its input (1) the maximally combined and reduced rules with an indication of the alternations for which each could account and (2) the URs involved in alternations, i.e. those for which feature specifications must be decided. For each alternation of a feature of morpheme, the total of all the feature specifications in all the rules in one direction which could account for that alternation is compared to the total for the rules in the other direction. The set of rules with the smaller total are established and the UR is adjusted, i.e. the feature specification involved in the alternation can be filled in. Rules can also be established in connection with members of "minimal pairs:, i.e. where two morphemes which can occur in the same environment differ by a single alternation, or when one of the other rules in set of these to account for an alternation in one direction has already been established.

Above the structure of PHONY's bare machine was discussed. The purpose of PHONY was to demonstrate the possibility of doing phonological analyses by computer. Its basic approach is similar to that of the phonologist, except that the bare machine is almost exclusively data-driven. A phonologist is guided in his analysis by his knowledge about what types of rules are common cross-linguistically. This property has been referred to as "naturalness". "Natural" processes are generally assumed to be phonetically motivated, although there are cases where the phonetic explanation is not known. When a phonologist encounters an alternation which could be the result of a natural rule, he examines the entire input corpus to see if the rule holds. Thus naturalness is a heuristic reducing the search.

Naturalness considerations could be easily incorporated into PHONY's bare machine by including a catalogue of natural processes. It would then be possible to match the rules as soon as they were generated in ALTFINDER against this catalogue, and if a match occurred, the whole input corpus could be checked to determine if the rules were valid. Alternatively, one could initially check the input corpus against the catalogue of natural processes to see if any of these were valid. This is a common approach in model-driven inference programs (cf.[15]). In either case, this incorporation would both serve as a heuristic to shortcut the search and also to reflect the theory's preferences for certain solutions which are not strictly simplest in terms of feature specifications. Another reflection of the theory would be in the order in which attempts are made to eliminate features. In addition, other evaluation criteria could be incorporated, for example, a facility for pattern congruity (cf.12, pp. 93-7).

The bottleneck in terms of run time for PHONY is in the LCD stage. It uses a very crude and uninformed technique for eliminating unnecessary feature specifications. The time involved in this stage could be reduced if many of the alternations were resolved as a result of matching against a catalogue of natural processes. Depending on the success rate of matching and testing and on the size of the catalogue, this could result in an overall savings in time. It would also help if heuristics guided the attempt to eliminate segments and features en masse. The elimination could be model-driven by expectations of the theory. Alternatively, PHONY could be incorporated as the performance element in a learning system which could optimize elimination of segments and features en masse (cf. [4]). The critic of such a system could be sensitive to the number of tests of the input corpus required with different orders and combinations of attempted feature eliminations.

A fundamentally different approach to the problem of the bottleneck would be to employ a bidirectional search. It would be data-driven in one direction, as PHONY's bare machine, from specific to more general processes, and model-driven in the other direction from the general natural processes to more specific rules. This would be similar to Mitchell's 'version space' approach [15].

None of the extensions could overcome the most significant limitation of PHONY - size of input corpus. In order to isolate the phonological rules of a language a certain number of utterances are required. The minimal number would be enough to exhibit the alternations which result from application of the rules as well as examples sufficient to show the limits of applicability of the rules, i.e. how specific the rules must be. In certain cases the set of forms necessary may not be large, but somehow this data must be selected from a larger corpus. In this respect PHONY was essentially a 'toy' program, it worked upon data found in phonological problems found in textbooks. The forms given in these problems were carefully chosen to be sufficient from which to isolate the rules.

PROPOSED APPROACH AND ITS JUSTIFICATION

PHONY with extensions for heuristics would essentially be an expert system, reflecting how a linguist might do phonological analysis. (This reflection would be more exact if PHONY were also extended to give it a capacity for "instance selection" (cf. [4]).) In the following discussion the methodological approach taken by linguists will be called into question.

Recall that PHONY processed its entire input corpus, generating specific rules, before it attempted to combine and reduce the rules and evaluate them. This method differs from that of most data-driven learning systems, which produce a current best hypothesis after each new training instance (cf.[15]). This process might continue until the set of training instances were exhausted or until the pattern, concept, or grammar has been learned in some sense (e.g. learnability of formal grammars was initiated by Gold [9]. Phonological analysis, i.e. the acquisition of the phonological rules and the URs (which can also be thought of as rewrite rules), would constitute a form of grammatical inference (cf. [2], [8]).

Recall that Chomsky regarded the development of an evaluation procedure to be a worthwhile subgoal for linguistic theory. The purpose of the evaluation procedure or metric was to choose between multiple grammars within the class of possible grammars which were compatible with the primary linguistic data. For phonology, the original proposed evaluation metric was the total number of feature specifications in the proposed set of rules. this did not prove adequate, and in The Sound Pattern of English (1968) Chomsky and Halle proposed the "linking conventions" as an extension to this metric which essentially allowed certain specifications or rules to be disregarded in the totaling of the feature specifications. This approach did not prove fruitful or popular and since that time there have been no proposals for evaluation procedures or metrics. It can be suggested that the difficulty in finding an evaluation metric may be a reflection of the fact that such an evaluation metric plays no role in language learning by the child. In this connection it should be pointed out that such an evaluation procedure comparing two or more prospective hypothesized grammars generally plays no role in AI learning systems or in formal procedures for grammatical inference.

Since no evaluation procedure was available to linguistic theory, all effort to resolve the problem of which of the multiple possible grammars consistent with the primary linguistic data should be chosen has involved attempting to restrict the class of possible grammars. Unless it is possible to limit the class of possible grammars consistent with the primary linguistic data of a language to only one grammar, some learning strategy will still be required to allow the choice of just one grammar. It should be borne in mind that a learning procedure may yield a unique grammar even in conjunction with a much larger class of grammars. In fact, there may be no way to characterize a class of grammars other than being discoverable by a certain effective learning procedure. This realization has very significant methodological implications for linguistic theory. Since it may not be possible to limit the class of possible grammars beyond a certain point, searching for a discovery procedure may in fact be necessary, and some effort in this direction would certainly seem to be justified.

Little work was done in th search for a formal set of procedures to discover or induce grammars for natural languages from the start of the Chomskyan revolution until the mid-seventies. Since then the work has centered on the learning of syntactic and semantic aspects of natural languages (cf. [17] for a review of the literature). Some of the most significant work has been done by Wexler and his co-workers on the learning of transformation grammars ([10], [20], [21]). This work involved the formal definition of the class of possible grammars, of the input available to the learner, of the language learning procedure, and of the criterion of success.

A particularly interesting aspect of this work was the proposal of "learnability constraints". These are constraints on the class of formal grammars which allow the proof of success of the learning process; it is assumed that the function of these constraints is to allow learnability ([20]). Some of these constraints make empirical claims and predictions as to the nature of linguistic data, and some of these constraints have been proposed independently by linguists concerned with descriptive adequacy ([21]).

The proposed approach is to first concentrate on defining a learning procedure and proving learnability, with the class of grammars as defined by the representation (discussed above) and with input similar to that of PHONY, i.e. with division into morphemes and association of different instances of the same morpheme. In order to prove learnability, it may be necessary to establish learnability constraints, as in Wexler's work discussed above. These would be compared with (and possibly would have derived their impetus from) the various constraints proposed by phonologists. The learning procedure would then be incorporated into a program to do phonological analysis. Subsequently, an effort would be made to improve the performance of the program by proposing (additional) learnability constraints, again comparing how these limits on the class of possible grammars compare to the constraints proposed by phonologists.

References

1. Becker, L. 1981. "PHONY: A heuristic phonological analyzer". In *Proceedings of the 19th Annual Meeting of the Association for Computational Linguistics*.

2. Biermann, A. W. & J.A. Feldman. 1972. "A survey of results in grammatical inference." In S.Watanabe (ed.), *Frontiers of Pattern Recognition*.N.Y. : Academic Press.

3. Block, B. 1948. "A set of postulates for phonemic analysis". *Language* 24: 3-46.

4. Buchanan, B. G., T. M. Mitchell, R.G. Smith and C. R. Johnson, Jr. 1979. "Models of learning systems." In J. Belzar, A. Holtzman, A. Kent (eds.), *Encyclopedia of Computer Science and Technology* N.Y. : Marcel Dekker, Inc. Vol 3, 24-51.

5. Chomsky, N. 1957. *Syntactic Structures* The Hague: Mouton.

6. Chomsky, N. 1965. *Aspects of the Theory of Syntax* Cambridge, Mass.: MIT Press.

7. Chomsky, N. and M. Halle. 1968. *The Sound Pattern of English*. N.Y.: Harper & Row.

8. Fu, K.S. and T. L. Booth. "Grammatical inference: Introduction and Survey - Part I". *IEEE Trans. Syst., Man., Cybern. SMC 5.1* 95555-111. - Part II, SMC 5.4: 409-423.

9. Gold, E. M. 1967. "Language identification in the limit." *Information Control 10*: 447-474.

10. Hamburger, H. and K. Wexler. 1975. "A mathematical theory of learning transformational grammar." *Journal of Mathematical Psychology 12*:137-177.

11. Hockett, C. F. 1942. "A system of descriptive phonology". *Language 18*:3-21.

12. Hyman, L. M. 1975. *Phonology: Theory and Analysis*. N.Y.: Holt, Rinehart and Winston.

13. Kenstowicz, M. and C. Kissebertz 1977. *Topics in Phonological Theory* N.Y.: Academic press.

14. Kiparsky, P. 1968. "How abstract is phonology?" In O. Figimura (Ed.), *Three Dimensions in Linguistic Theory* 1973. Tokyo: TEC.

15. Mitchell, J. M. 1979. "An analysis of Generalization as a search problem" *Proceedings of the Sixth International Joint Conference on Artificial Intelligence* 577-582.

16. Petrick, S. R. 1963. "Minimization procedures in phonology." *AFCRL-63*-324.

17. Pinker, S. 1979. "Formal models of language learning." *Cognition* *7*:217-283.

18. Shaw, D.W. Swartout, and C. Green. "Inferring LISP programs from examples." *Proc. of the Fourth International Joint Conference on Artificial Intelligence*, 260-267.

19. Vere, S. A. 1977. "Induction of relational productions in the presence of background information." *Proc. of the Fifth International Joint Conference on Artificial Intelligence* 349-355.

20. Wexler, K. 1981. "Some issues in the Theory of Learnability." In L. Baker and J. McCarthy (Eds.), *The Logical Problem of Language Acquisition* Cambridge, Mass.: MIT Press.

21. Wexler, K. and P. Culicover. 1979. *Formal Principles of Language Acquisition* Cambridge, Mass: MIT Press.