

DTIC FILE COPY

2

AD-A219 095

**A CONNECTIONIST IMPLEMENTATION OF
COGNITIVE PHONOLOGY**

Technical Report AIP -75

Deirdre W. Wheeler & David S. Touretzky

Department of Psychology
Carnegie Mellon University
Pittsburgh, PA 15213-3890

May 26, 1989

**The Artificial Intelligence
and Psychology Project**

Departments of
Computer Science and Psychology
Carnegie Mellon University

Learning Research and Development Center
University of Pittsburgh

DTIC
ELECTE
MAR 16 1990
S B D

Approved for public release; distribution unlimited.

90 03 12 036

A CONNECTIONIST IMPLEMENTATION OF COGNITIVE PHONOLOGY

Technical Report AIP -75
Deirdre W. Wheeler & David S. Touretzky
Department of Psychology
Carnegie Mellon University
Pittsburgh, PA 15213-3890

May 26, 1989

This research was supported by the Computer Sciences Division, Office of Naval Research, under contract number N00014-86-K-0678, by National Science Foundation grant EET-8716324, and by a contract from Hughes Research Laboratories.

Reproduction in whole or in part is permitted for any purpose of the United States Government. Approved for public release; distribution unlimited.

DTIC
ELECTE
MAR 16 1990
S B D

REPORT DOCUMENTATION PAGE				
1a. REPORT SECURITY CLASSIFICATION Unclassified		1b. RESTRICTIVE MARKINGS		
2a. SECURITY CLASSIFICATION AUTHORITY		3. DISTRIBUTION / AVAILABILITY OF REPORT Approved for public release; Distribution unlimited		
2b. DECLASSIFICATION / DOWNGRADING SCHEDULE				
4. PERFORMING ORGANIZATION REPORT NUMBER(S) AIP - 75		5. MONITORING ORGANIZATION REPORT NUMBER(S)		
6a. NAME OF PERFORMING ORGANIZATION Carnegie-Mellon University	6b. OFFICE SYMBOL (If applicable)	7a. NAME OF MONITORING ORGANIZATION Computer Sciences Division Office of Naval Research		
6c. ADDRESS (City, State, and ZIP Code) Department of Psychology Pittsburgh, Pennsylvania 15213		7b. ADDRESS (City, State, and ZIP Code) 800 N. Quincy Street Arlington, Virginia 22217-5000		
8a. NAME OF FUNDING / SPONSORING ORGANIZATION Same as Monitoring Organization	8b. OFFICE SYMBOL (If applicable)	9. PROC. INSTRUMENT IDENTIFICATION NUMBER N00014-86-K-0678		
8c. ADDRESS (City, State, and ZIP Code)		10. SOURCE OF FUNDING NUMBERS p4000ub201/7-4-86		
		PROGRAM ELEMENT NO N/A	PROJECT NO. N/A	TASK NO. N/A
		WORK UNIT ACCESSION NO N/A		
11. TITLE (Include Security Classification) A Connectionist Implementation of Cognitive Phonology				
12. PERSONAL AUTHOR(S) Wheeler, Deirdre W. and Touretzky, David S.				
13a. TYPE OF REPORT Technical	13b. TIME COVERED FROM 86Sept15 to 91Sept14	14. DATE OF REPORT (Year, Month, Day) May 26, 1989		15. PAGE COUNT 18
16. SUPPLEMENTARY NOTATION [redacted] paper presented at the Berkeley Workshop on Cognitive Phonology (May 26, 1989)				
17. COSATI CODES			18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)	
FIELD	GROUP	SUB-GROUP	cognitive phonology	
			connectionist symbol	
			linguistics.	
			processing.	
			many-maps model	
19. ABSTRACT (Continue on reverse if necessary and identify by block number) SEE REVERSE SIDE				
20. DISTRIBUTION / AVAILABILITY OF ABSTRACT ☐ UNCLASSIFIED/UNLIMITED ☒ SAME AS RPT ☐ DTIC USERS			21. ABSTRACT SECURITY CLASSIFICATION	
22a. NAME OF RESPONSIBLE INDIVIDUAL Dr. Alan L. Meyrowitz			22b. TELEPHONE (Include Area Code) (202) 696-4302	22c. OFFICE SYMBOL N00014

ABSTRACT

This paper reports on initial results of an effort to actually implement Lakoff's theory of cognitive phonology in a connectionist framework. For all sorts of reasons, standard generative phonological theories cannot be implemented in connectionist frameworks. Lakoff's theory of cognitive phonology offers solutions to some of these problems in that it offers an alternative way to think about derivations and ordered rules, as well as eliminating the need for right-to-left iterative rule application.

We will begin by describing our "many maps" model and show how Lakoff's cross-level phonological constructions may be implemented. As will become clear, our basic assumption is that all phonological constructions should express correlations between levels and be satisfied in parallel, simultaneously, across the entire input domain.

After describing the general properties of our model and how mappings between levels are actually implemented, we will consider a number of specific cases. In particular, we will focus on those apparently involving iterative application of rules: Slovak shortening, Gidabal shortening, and vowel harmony in Yawelmani. Our challenge is obviously to provide alternative accounts of those cases involving intra-level rules in Lakoff's theory. We believe that the clustering mechanism allows us to do this. Finally, the complex rule interactions in Icelandic will be addressed, and we will show that our theory, though very tightly constrained, can handle this case as well.

INTERIM REPORT per Alan Meyrowitz
ONR/Code 1133
TELECON

3/15/90

VG



Accession For	
NTIS GRA&I	☒
DTIC TAB	☐
Unannounced	☐
Justification	
By <i>per telecon</i>	
Distribution/	
Availability Codes	
Dist	Avail and/or Special
<i>A-1</i>	

A Connectionist Implementation of Cognitive Phonology

paper presented at the
Berkeley Workshop on Cognitive Phonology
May 26, 1989

Deirdre W. Wheeler

Department of Linguistics
University of Pittsburgh
Pittsburgh, PA 15260
(wheeler@oberon.lcl.cmu.edu)

and

David S. Touretzky

School of Computer Science
Carnegie Mellon
Pittsburgh, PA 15213
(dst@cs.cmu.edu)

1. Introduction

This paper reports on initial results of an effort to actually implement Lakoff's theory of cognitive phonology in a connectionist framework. For all sorts of reasons, standard generative phonological theories cannot be implemented in connectionist frameworks. Lakoff's theory of cognitive phonology offers solutions to some of these problems in that it offers an alternative way to think about derivations and ordered rules, as well as eliminating the need for right-to-left iterative rule application.

We will begin by describing our "many maps" model and show how Lakoff's cross level phonological constructions may be implemented. As will become clear, our basic assumption is that all phonological constructions should express correlations between levels and be satisfied in parallel, simultaneously, across the entire input domain.

Not all of cognitive phonology is quite so easy to implement. Lakoff draws a distinction between cross-level constructions and intra-level well-formedness constraints. It is the intra-level well-formedness constraints which turn out to be problematic to implement since they do not actually express correlations between levels. In addition, Lakoff allows cross level constructions to be stated with environments at the input level (e.g. Slovak), the output level (e.g. Gidabal), or at either level (e.g. Icelandic). While these possibilities are seemingly innocent formal variations on paper, they are very significant (and problematic) when it comes to actually trying to implement them in a connectionist network.

In Lakoff's 1988 LSA paper, there are several technical asides in which he appeals to Smolensky's 'Harmony theory', saying that he conceives of constructions as increasing harmony both within and across levels. Constraints are simultaneously satisfied within a domain in such a way as to achieve the maximally harmonious state -- the state which best satisfies any and all well-formedness constraints. The major problem which arises here is that it is not in fact clear how to implement this notion of maximal harmony.

The problem here is that an appeal to "harmony" skirts an important computational question: how is the network able to find the most harmonious state? That is, how does the P-level, for example, 'automatically' settle into the representation that best meets all the constraints. It could deterministically search all possible states and pick the best one, but this would take a very long time. Nervous systems are too slow for sequential search. Smolensky's Harmony Theory is based on parallel, stochastic (simulated annealing) search, but annealing search also takes too much time in order to get valid results. Our goal is to avoid reference to a 'harmony' process that is unimplementable in current neural network theories.

Besides the time argument, it's not clear how to express phonological operations, which may perform complex chains of insertions and deletions, in ways that a stochastic neural net could deal with. (In other words, it's not clear how to wire such a network.) Nor is it clear how the constraints on phonological well-formedness can be efficiently encoded in a neural net. We are not saying it can't be done, but we don't at the moment know how to do it.

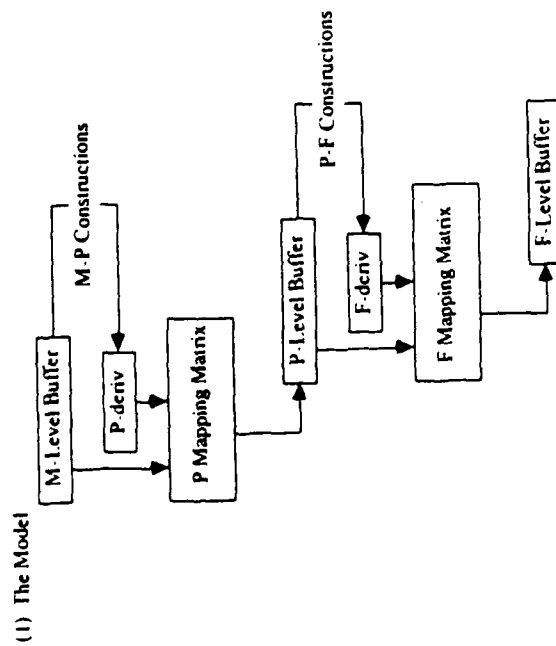
The specific implementation described here does not appeal to harmony theory, but rather is strictly deterministic in character. Cross-level constructions sanction changes to be made to segments in the input buffer to satisfy the constraints of the output level. We solve all the implementation problems associated with (iterative) intra level rules by introducing a single clustering mechanism which operates on the input buffer and defines the domain of application of rules. In effect, there are no intra level rules. As a consequence, it appears to be possible to constrain the phonological theory so that only cross level constructions are

permitted, and at the same time offer an account of iterative processes which can in fact be implemented in a connectionist framework.

After describing the general properties of our model and how mappings between levels are actually implemented, we will consider a number of specific cases. In particular, we will focus on those apparently involving iterative application of rules: Slovak shortening, Gidabal shortening, and vowel harmony in Yawelmani. Our challenge is obviously to provide alternative accounts of those cases involving intra-level rules in Lakoff's theory. We believe that the clustering mechanism allows us to do this. Finally, the complex rule interactions in Icelandic will be addressed, and we will show that our theory, though very tightly constrained, can handle this case as well.

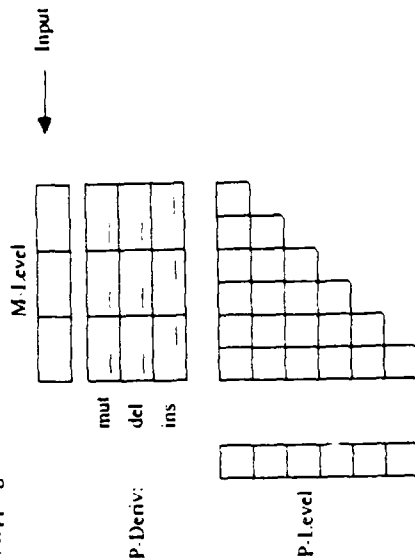
II. Many Maps Model

This section will outline the general architecture of our 'many maps' implementation. The overall model has the following structure:



First, consider the mapping from M-level to P-level.

(2) M-P Mapping



The M-level buffer contains phonemic segments. The P-deriv buffer describes the changes necessary to derive the P-level form of the utterance from the underlying M-level form. The input to the P-mapping matrix consists of M-level segments plus the changes recorded in P-deriv. The output of the matrix forms the contents of the P-level buffer. Segments at M-level are by default mapped to identical segments at P-level. Each M-level segment has an entry in P-deriv where a change may be recorded if required by some construction of the grammar. Through P-deriv, phonological constructions will have the effect of overriding the default identity mapping (c.f. Lakoff 1988). Mutation, insertion, and deletion are all supported in the current implementation. A segment may be mutated by specifying the features that are to change in the first row of P-deriv. Segments may be deleted by turning on the deletion bit in the second row, in which case the M-level segment will not appear in the P-level buffer. And segments may be inserted by specifying features in the third row. In the current implementation, segments are always inserted to the right of some segment, though that is not a necessary limitation. One constraint which is imposed by the hardware, however, is that only one segment may be inserted in the mapping process between two input segments. We know of no counterexamples.

The function of the mapping matrix is to provide a 'clean' output representation where all segments are adjacent and right justified. The upper-diagonal matrix in the figure represents an array of connectionist mapping units. When one of the units is active (which will be represented by showing a segment in the appropriate unit), the segment in that input column is copied to the corresponding output row. At the same time, any changes recorded in P-deriv are made. At most one unit may be on in any row or column. Thus, the matrix ensures that the order of input segments is preserved in the output, and that there are no 'gaps' in the case of deletions or 'collisions' in the case of insertions. For each segment in the input buffer there are two units in the matrix, with the right

hand unit being activated in the case of insertions. If this unit is not turned on, the input segments are adjacent in the output representation.

We will draw on alternations in Yawelmani to illustrate these processes. Consider, first, epenthesis. Lakoff (1988:21) states the epenthesis construction as:

(3) Yawelmani epenthesis:

M: C | C [C,N]
P: | [+syl] +high |

Here, Lakoff is assuming a general theory of markedness and underspecification (Archangeli 1984). Specifying that the epenthetic segment is simply a high vowel is sufficient in this case on the assumption that the following default rule will fill in the value for [back]:

(+1) High vowel markedness (default): if [+sv, +high], then [-back]

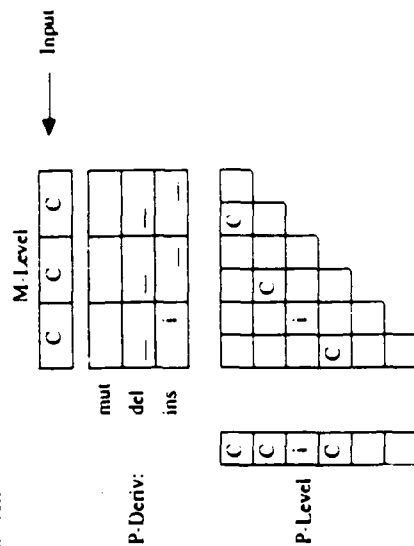
We have no doubt that underspecification plays an important role in phonological analyses and that the model should allow for feature values to be unspecified in cases where the value is predictable. In our implementation, segments are represented as sets of bits, roughly corresponding to phonological features, which may be on or off. One way to incorporate underspecification is to assume that for each feature $[F]$, there is one bit for $[+F]$ and one bit for $[-F]$. If a segment is unspecified for $[F]$, then both bits are off.

A subset of Lakoff's intra-level rules (e.g. High Vowel Markedness) can be implemented straightforwardly as post-F-level rules which serve the function of filling in unmarked values for any features which are not specified at that point. The processing which needs to be done to fill in feature values can all be done completely in parallel across the entire domain, given the very constrained form of these rules.

It is worth noting that high vowel markedness is very different in character from some of Lakoff's other intra-level rules (e.g. Yawellmani vowel harmony). Markedness rules relate features within segments, and are not dependent on broader contexts. Allowing this class of rules in no way compromises our position that there are no intra-level rules. These markedness principles do not interact with the other rules of the grammar in any way. As a consequence, all the problems encountered with trying to implement iterative, intra-level rules do not arise. Thus, we allow for a final component, after the mapping to F-level, where correlations between features may be expressed.

In our model, the insertion is accounted for in the following manner. If a string of three consonants or two word-final consonants appears in the input buffer (M-level), then epenthesis takes place. This is accomplished by recording the features [+svl, +high] (represented for convenience as [i]) in P-deriv as shown below:

(S) Epenthesis



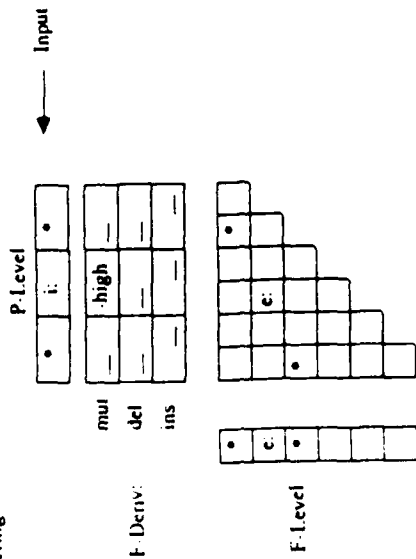
The lowering rule in Yawelmani provides an example of a mutation process. Lakoff posits the following P-F construction to account for the lowering process, whereby all long vowels at P are non-high at F.

(6) Yawelmani lowering (Lakoff 1988:22)

P: [+syl, +long]
F: [-high]

In our model, this construction sanctions a change to be recorded in F-deriv, making all long vowels [-high], as illustrated in the following figure. An asterisk is used here to represent any segment.

(7) Lowering

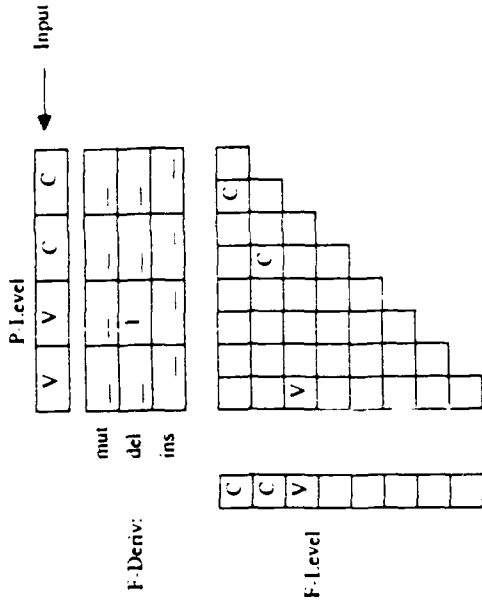


The shortening process in Yawelmani could be implemented in exactly the same way, with the following construction (Lakoff 1988:21) sanctioning changes to be recorded in P-deriv.

(8) Yawelmani shortening:
P: [+syl, +long] C C
F: [-long]

Alternatively, if long vowels are represented as a sequence of identical vowels (VV), then shortening would offer an example of a deletion process, as illustrated in the following mapping:

(9) Vowel deletion



In this case, since the deletion bit is turned on in P-deriv for the segment corresponding to the second half of the long vowel, no units fire in the mapping matrix and hence there is no trace of the vowel at F-level. Here again, the mapping matrix serves the function of guaranteeing that there are no gaps at F-level.

III. Left-to-right vs. Simultaneous

In Lakoff's discussion of iterative rules, he points out that in cognitive phonology, iteration is a natural consequence of the fact that a single construction may be simultaneously satisfied more than once in a word. He discusses vowel shortening processes in Slovak and Gidabal to illustrate this point. In a standard generative analysis, the same rule could apply in both languages, with the differences in the derived forms being attributed to the fact that the rule applies iteratively from right-to-left in Slovak, but left-to-right in Gidabal. The rule is given below in (10), with schematic examples in (11).

- (10) [+syl, +long] → [-long] / [+syl, +long] C₀ —
- (11) Slovak (R to L iterative) Gidabal (L to R iterative)
/V/ C V: C V: C V: / /V/ C V: C V: C V: /
V: C V: C V: C V: V: C V: C V: C V:
V: C V: C V: C V: V: C V: C V: C V:
[V: C V C V C V]

Lakoff's theory has two very desirable consequences with respect to cases like these. First, it is possible to account for the pattern of shortening in Slovak without having to assume that rules can iterate from right-to-left. Second, the theory does not require a series of ill-formed intermediate steps in the derivation. According to Lakoff's analysis, Slovak

and Gidabal differ only in the level at which the environment is stated. Lakoff states the following construction for Slovak:

- (12) Slovak Shortening
M: $[+syl, +long]$ C_0 $[+syl, +long]$
P: $[-long]$

What is particularly significant about this case is that by stating the construction as in (12) above, it is not necessary to assume that the 'rule' applies iteratively from right-to-left. As shown below, all non-initial long vowels meet the M-condition of the construction and therefore must be short at P.

- (13) M: V: C V: C V: C V:
P: V: C V C V C V

On the other hand, the shortening construction in Gidabal is stated as in (14), with the environment at P. The consequence of this apparently simple difference is that only alternate vowels may shorten, as illustrated in (15).

- (14) Gidabal Shortening
M:
P: $[+syl, +long]$ C_0 $[-long]$

- (15) M: V: C V: C V: C V:
P: V: C V C V: C V

Lakoff speaks of the processing proceeding from left-to-right in both Slovak and Gidabal, saying: "The cognitive phonology approach permits processing in real time left-to-right in both cases -- with no unnecessary intermediate stages." (Lakoff 1988: 13)

In fact, the analyses in (13) and (15) are the only ones which are well-formed according to the constraints imposed by the shortening constructions, so it is not in fact necessary to assume that strings are processed from left-to-right. In Gidabal, for example, if the construction is interpreted as imposing a constraint prohibiting sequences of long vowels, then the only way to satisfy the constraint is to shorten the second and fourth vowels. If the third vowel was shortened, then the first two vowels would violate the constraint.

So, in these cases, cognitive phonology appears to offer a very elegant solution to the problem of the computational complexity of both iterative application of rule: and right-to-left processing. There is one significant difference between these constructions, however, that detracts from the explanation. A closer look at the Gidabal construction reveals that both its environment and change are stated at P-level. While the construction is stated as an M-P rule, it could just as well be a P-level rule, proceeding from left-to-right, perhaps stated as follows:

- (16) Intra-level version of Gidabal shortening:
P: If $[+syl, +long]$ C_0 X, then if X = $[+syl]$ then X = $[-long]$

Thus, at least as it was originally formulated, the theory is not constrained enough to provide a unique analysis of the shortening process in Gidabal. In general, the theory leaves open the question of whether processes should be stated as cross-level constructions or intra-level constructions.

In terms of implementing constructions in the many maps model described here, the shortening construction in Gidabal is particularly problematic. While shortening in Slovak can be implemented straightforwardly in the current model, it is impossible to implement Gidabal without making major modifications. The problem is that with the Gidabal construction it is not possible to simply look at the M-level representation and determine what changes need to be effected at P-level. Part of the environment is actually stated at P, and thus information from P-level must be accessible for changes to be made correctly.

Before going on to describe our clustering mechanism which will offer a way out of this bind, we will consider Yawelmani vowel harmony -- a case where Lakoff's model really does need to rely on left-to-right processing. In a standard generative analysis, the harmony rule would be stated as an iterative rule, applying from left-to-right. Lakoff treats vowel harmony as an intra-level construction, applying at P-level. He formulates the construction as follows (Lakoff 1988: 22):

- (17) P: If $[+syl, +rd, +high]$ C_0 X, then if X = $[+syl, +high]$ then X = $[+ind, +back]$

Unlike cross-level constructions which clearly describe correlations between levels, constructions like the vowel harmony above have much more of a derivational flavor to them. They do not establish a mapping between levels, but rather serve as constraints on the well-formedness within a level. In Lakoff's description of Harmony he gives the following 'derivation' for [do:so]: 'report (dubitative)':

- (18) M: do:s+al
P: do:s+ol
F: do:s+ol

And, says simply that: "the harmony constraint is met at P" (Lakoff 1988: 22). Here again we don't know how to implement that kind of rule in a neural network with stochastic search. Expressing the constraints would be hard to do, and searching the space of possible P-level representations would take too long. Iterative rules solve the problem, but only by reintroducing sequentiality and intermediate states. While (18) does not actually involve iteration, it does illustrate the problem. We will consider cases involving iteration shortly.

However, before going on to explain how to account for vowel harmony without having to make any further modifications in the model, it is worth considering why Lakoff is forced to state Yawelmani vowel harmony as an intra-level construction rather than a cross-level construction. Recall that vowel harmony must follow epenthesis because epenthesis both feeds and bleeds harmony. He gives the example in (19) below to illustrate the feeding relation, with the epenthetic vowel undergoing harmony.

- (19) /ʔugn+hin/ "drinks"
ʔuginhin epenthesis
ʔuginhin harmony on epenthetic vowel
ʔuginhun harmony on the final vowel
[ʔuginhun]

And the following example shows how epenthesis can block harmony.

- (20) /logw + xa/ "let's pulverize"
 logiwxa epenthesis
 ... harmony
 [logixwa]

Given that epenthesis is an M-P construction, if harmony is also an M-P construction then this would incorrectly predict that epenthetic vowels do not undergo or block harmony since they are not present at M-level. If harmony is a P-F construction, then its interaction with the other P-F constructions, namely lowering and shortening, must be considered. Recall that lowering and shortening both have their environments at P-level, with changes at F-level. It is the lowering rule which is relevant in this case. In standard generative terms, harmony must precede lowering since lowering (incorrectly) bleeds harmony. Consider the following derivation, taken from Lakoff (1988:20):

- (21) /sudu:k*hin/ "removes"
 sudu:k*hin harmony
 sudok*hin lowering
 sudok*hin shortening
 [sudok'hun]

If lowering applies before harmony then harmony will not apply and the final vowel will surface as unrounded. With constructions expressing correlations between levels, it might appear as though harmony could be stated as in (22) below, a P-F mapping with environment at P-level.

- (22) Yawelmani Harmony
 P: [+syl, +rd, ahigh] C₀ [+syl, ahigh]
 F: [+md, +back]

This does allow us to correctly account for the fact that lowering does not bleed harmony. The environments for both constructions are at P-level, and consequently the effects of lowering are 'invisible' to harmony. All three constructions may be satisfied simultaneously in the mapping between P-level and F-level.

- (23) P: sudu:k + hin
 F: sudokhun

There is an obvious problem with the harmony rule in (22), however, which is that with the environment stated at P-level it will not apply iteratively. It will incorrectly predict that:

- (24) M: ?ugn*hin
 P: ?uginhin by epenthesis
 F: ?uginhin by harmony
 e[?uginhin]

No doubt, this is one of the reasons Lakoff analyzed harmony as a P-level construction rather than a P-F construction. We will now show how cases like Yawelmani and Gidabal can be handled without having to allow intra-level constructions.

IV. Clustering

In an earlier paper, Turetzky 1989, the Many Maps model was in fact modified to allow for a loop from P-level back through 'P-deriv'. This added significantly to the complexity of the model, but seemed to be necessary in light of the fact that processes like Gidabal shortening seemed to be sensitive to the effects of the construction applying elsewhere in the string. Our current position, however, is that such a loop is not in fact necessary. Lakoff's theory does not clearly distinguish between the 'changes' sanctioned by cross level constructions and their domain of application. We propose that there is a clustering mechanism which identifies the domain. Changes specified in P- or F-deriv to satisfy the constraints imposed by constructions will take effect within the domain of the cluster, recording changes in P-deriv for all elements of the cluster.

We are drawing heavily here on the insights of autosegmental phonology (Williams 1971/76, Goldsmith 1976, etc.) as will become clear shortly. In effect, our clustering mechanism will provide a means of identifying 'tiers' (e.g. a vowel tier), the beginning of a cluster, and the elements within the cluster. First, the mapping architecture already described is sufficient for establishing a vowel tier -- only those segments which are [+syllabic] are mapped to the output level. Thus, the 'filter' for clustering in Yawelmani is the feature [+syllabic]. We also need to identify the triggers and elements of clusters, as shown below. By definition, elements must be adjacent to either the trigger or another element on a given tier. To complete the rule, a change is specified, which applies to all elements of the cluster (but not to the trigger).

- (25) Yawelmani Vowel Harmony:
 Filter: [+syllabic]
 Trigger: [+md, ahigh]
 Element: [ahigh]
 Change: [+md]

The clustering mechanism has a left-to-right processing preference and given the specifications above turns on the 'trigger' and 'element' bits where appropriate. Round vowels are triggers and the 'element' bit is turned on if a vowel is the same height as the trigger. Sequences of vowels in which the 'element' bit is activated form a cluster. In a hypothetical example with five vowels, we have the following representation, where a '+' is intended to indicate that the bit is turned on.

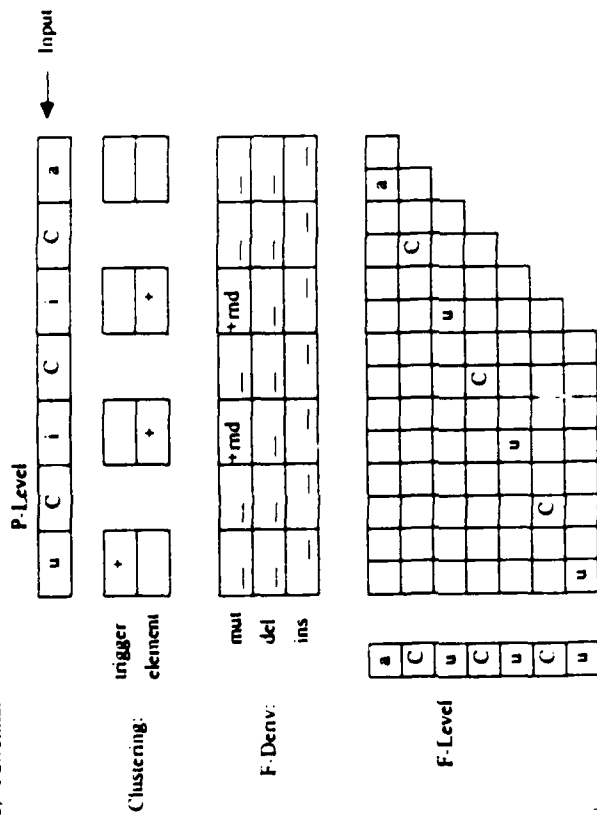
- (26) i u i i a
 trigger + + +
 element

We are assuming two general (universal) conventions on clustering. First, triggers cannot also be elements, and second, there is a preference for segments to be elements rather than triggers -- other things being equal. This will mean that in a string like /u i i u a/, for example, all vowels after the initial one will be elements of a single cluster. The final [a] cannot be an element in the following case even though it is a vowel at the same height as the trigger, because elements must be adjacent to triggers or other elements.

- (27) o i a
 trigger +
 element

Thus, vowel harmony in Yawelmani 'iterates in cases like (26) because sequences of high vowels have been clustered. This fairly minimal extension of the model allows us to maintain the position that all 'rules' apply simultaneously, in parallel, to whole words (or phrases). The following figure illustrates the effects of harmony.

(28) Yawelmani



The exact clustering mechanism may be used to explain the alternating vowel length patterns of Cidabal and Slovak. Here again, there is no need to assume any left-to-right application of the shortening rules. Vowel clustering in Slovak works as follows:

(29) Slovak shortening:

Filter:	[+syllabic]
Trigger:	{+long}
Element:	[+long]
Change:	[-long]

On the assumption that triggers cannot be elements and that there is a preference for segments to be elements if possible, this yields the following clustering pattern:

- (30)
- | | |
|----------|---------------|
| Trigger: | V: V: V: V: V |
| Element: | + + + + |

Vowel shortening then applies to all vowels which are elements of the cluster. The alternating pattern found in Gidabal results from specifying that elements are adjacent to triggers.

(31) Gidabal shortening:

Filter:	{+syllabic}
Trigger:	{+long}
Element:	{+trigger}
Change:	{-long}

(32) **Gidabal vowel clusters**

	V: V:	V: V:	V: V:	V: V:
Trigger:	+	+	+	+
Element:				

Just as in Slovak, vowel shortening applies to all the vowels which are elements of a cluster. In this case, though, the difference is that clusters consist of only single vowels. Although our clustering algorithm apparently involves a left-to-right preference for building clusters by grouping elements together, this does not introduce sequentiality into the model the way self-feeding intra-level rules do. The reason is that clustering happens in parallel over the entire buffer. All rules fire together. In parallel, after clustering is complete, Lakoff's Gidabal solution (if you don't buy the appeal to Harmony), rules must fire sequentially from left to right in order to achieve the desired outcome.

It is worth noting here that requiring that elements be adjacent to triggers in *Crishabul* limits clusters to binary constituents. This is reminiscent of the distinction in standard metrical theory between bounded and unbounded constituents (Hayes 1981). We believe that the clustering mechanism will extend straightforwardly into the domain of stress assignment, though a thorough discussion of stress is beyond the scope of this paper.

V. Icelandic

Before concluding, we will consider one final case which appears to pose a real challenge for our constraint limiting rules to cross-level correlations. Standard generative analyses of Icelandic involve a complex interaction of unlaunt, syncope, and vowel reduction. There seems to be no strict order of application of the rules which is descriptively adequate. In Lakoff's analysis, syncope and u-unlaunt are both M-P constructions. The complex interaction is accounted for by assuming that the environment for u-unlaunt may hold at either M- or P-level. The constructions are stated as follows:

(33) Syncope (Lakoff 1988:15)

syncope (Lakoff 1986:15)

M: V D + V
P: |
(D = [-syl, +cor, +lax])

(34) Umlaut (Jakoff 1988:18)

M: [+sy], +low, +back
P: [-low, -back]

The umlaut construction says that an M-level /a/ corresponds to a nonlow, nonback vowel at P-level when it is followed by a /u/ at either level. It should be pointed out here that this construction does not actually make the correct predictions with respect to the derived segment. If the M-level segment is /a/, which is [-round], then specifying that it corresponds to a segment which is [-low, -back] at P-level should mean that the P-level segment is [e], not [o]. We assume that this is a minor oversight and that the feature [-round] should be included. At any rate, Lakoff claims to be able to account for the traditionally problematic cases like the following:

(35) M: bagg + ul + i
P: böggli
(syncope, and u-umlaut with env. at M)

(36) M: bagg + i + u
P: bögglu
(syncope, and u-umlaut with env. at P)

An additional rule interacts with u-umlaut, namely vowel reduction, which Lakoff states as follows:

(37) Vowel Reduction (Lakoff 1988:18):
P: If [+syl, -str, -low, -back], then [+high]

Actually, this is not an accurate formulation of this 'rule' either since specifying the value [+high] in a nonlow, nonback vowel yields [i] not [u]. Again, we assume that this is a minor typographical error in Lakoff's manuscript.

Together, u-umlaut (two occurrences) and vowel reduction sanction the following correspondence in Lakoff's analysis:

(38) M: fainað + um
P: fönuðum

While this analysis appears to offer a very nice explanation for a complex array of facts, it suffers from the same problems as discussed earlier in terms of implementation. Lakoff is again appealing to the notion of "harmony" in the system, and it is not at all clear how this can actually be implemented in a connectionist framework.

Our solution to this complex interaction of rules is to assume that u-umlaut is both an M-P and a P-F rule. A vowel which undergoes syncope may still trigger u-umlaut since both constructions have their environments at M-level. And, with u-umlaut as a P-F rule, syncope vowels will not interfere. Thus, we have a slightly different picture of the derivation of [ölnum] than Lakoff offers.

(39) M: alin + um
P: al num
F: ölnum
(syncope, u-umlaut)

Having u-umlaut as a P-F construction does not interfere in any way with the u-epenthesis rule which inserts a [u] before an unsyllabified /v/. Since both rules have their environments at P-level and changes at F-level, the epenthetic [u] is invisible to u-umlaut

(40) M: dag + r
P: dag r
F: dagur
(u-epenthesis)

The derivation of forms like [fönuðum] in (38) is also straightforward assuming the clustering mechanism described earlier. Clusters are built according to the following specifications, with the umlaut rule applying to all vowels in a cluster simultaneously. The only special stipulation which needs to be made is that triggers occur to the right of elements within a cluster.

(41) Icelandic u-umlaut -- with R-to-L clustering
Filter: [+syllabic]
Trigger: [u]
Element: [a]
Change: [-low, -back, +round]

One problem does remain, however, and this has to do with the interaction of u-umlaut and vowel reduction as P-F constructions. In cases where u-umlaut applies as a P-F construction, then we would not expect vowel reduction to apply since vowel reduction says that an unstressed /ö/ at P-level corresponds to a /u/ at F-level. What this suggests is that vowel reduction is not actually a P-F construction, but rather is part of the set of default mapping rules which apply at the very end of any derivation to fill in default values for phonological features which have remained unspecified, and generally, to map feature representations to the 'closest' well-formed phonetic segment. In this case, we can posit the following default mapping rule.

(42) If [+syl, -str, -low, -back, +round], then [+back, +high]

This adjustment is forced by the fact that there are no instances of an unstressed [o] in the surface phonetic representations of Icelandic. Thus, unstressed /a/ corresponding to an /u/ at either P or F will ultimately surface as [u], correctly characterizing the interaction of u-umlaut and vowel reduction. Note that while the rule in (42) looks very similar to Lakoff's P-level rule, there are important differences between these processes and Lakoff's intra-level rules. In our model, default mapping takes place only at the very end, not within each level. Also, and more importantly, these rules are very tightly constrained. They may only specify features to be filled in on the basis of features specified for that segment. They may not refer to, or in any way be conditioned by, adjacent segments. They simply characterize constraints on the realization of segments, guaranteeing well-formedness in the surface phonetic representation.

VI. Conclusion

Our goals here have been twofold. We have focussed on describing our connectionist implementation of Lakoff's cognitive theory of phonology, and at the same time have considered several theoretical issues and have argued for theoretical constraints. The implementation of cross-level constructions is straightforward in our 'many maps' model, even when there are apparently several which need to be satisfied simultaneously. However, iterative rules which need to refer to their own output are more difficult to account for. Appeals to harmony theory, in Smolensky's sense, do not offer an implementable solution. The clustering mechanism which we have proposed here offers a means of achieving this goal. Our constraint on constructions is actually stronger than just saying that only cross-level constructions are permitted. Given Lakoff's formalism, a cross-level construction (e.g. Gidebal shortening) may be stated with the environment at the output level. We suggest that constructions be limited to those which may be stated as cross-level constructions with environments at the input level.

Our mapping and clustering mechanisms offer a means of implementing autosegmental processes in general. While we have focussed on the vowel tier here, we are confident that these mechanisms will also provide a basis for implementing other processes as well. In addition it is, in principle, possible to incorporate the insights of underspecification theory into our model. While we do not currently draw very heavily on underspecification, we fully support theoretical work in this area.

At an intuitive level it feels nice to think in terms of 'harmony' within the system and simultaneous satisfaction of all constructions (and/or constraints) within a level. Implementing this is a different story. In our model, the most harmonious state is reached in a deterministic fashion by recording the necessary changes in P-deriv or F-deriv, as appropriate. Constructions may be thought of as well-formedness constraints on the output level, with changes being recorded only where constraints would otherwise be violated. In effect, the system reaches the most harmonious state directly, without taking the time to consider all the other possible dis-harmonious states.

One final note. And we leave this as an open-ended research question. Two major thrusts of a connectionist approach to phonology are implementability and learnability. To the best of our knowledge, no one has seriously tackled the problem of learnability in this domain yet. Perhaps the reason that linguists have been reluctant to worry about how rules are actually acquired is that there was no computational model powerful enough, yet constrained enough, to allow efficient rule induction. We certainly have not shown learnability yet, but a system as simple as the one modelled here, with no iterative rules and no intermediate states, should be easier for a language acquisition device to formulate rules for.

ACKNOWLEDGEMENTS

This work was sponsored by the Office of Naval Research under contract number N00014-86-K-0678, by a contract from Hughes Research Laboratories, and by National Science Foundation grant EET-8716324.

References:

- Archangeli, D. (1984) "Underspecification in Yawelmani phonology and morphology". Ph.D. Dissertation. M.I.T.
- Goldsmith, J. (1976) "Autosegmental phonology". Ph.D. Dissertation. M.I.T.
- Hayes, B. (1981) "A metrical theory of stress rules". Ph.D. Dissertation. M.I.T.
- Lakoff, G. (1988) "Cognitive phonology". Draft of paper presented at the LSA Annual Meeting.
- Smolensky, P. (1986) "Information processing in dynamical systems: foundations of harmony theory". In D.E. Rumelhart and J.L. McClelland (eds.) *Parallel Distributed Processing*, Cambridge: MIT Press.
- Toutretsky, D. S. (1989) "Towards a connectionist phonology: the 'many maps' approach to sequence manipulation." *Proceedings of the Eleventh Annual Conference of the Cognitive Science Society*. Hillsdale, NJ: Lawrence Erlbaum Associates.
- Williams, E. (1971/76) "Underlying tone in Margi and Igbo". *Linguistic Inquiry* 7, pages 463-484.