A Connectionist Implementation of Cognitive Phonology

Deirdre W. Wheeler[1]
David S. Touretzky[2]

October 1989

CMU-CS-89-144 3

[1]Department of Linguistics
University of Pittsburgh
Pittsburgh, PA 15260

[2]School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

An earlier version of this paper was presented at the Berkeley Workshop on Constraints vs. Rules in Phonology, May 26-27, 1989.

Abstract

This paper reports on an initial implementation of Lakoff's theory of cognitive phonology in a connectionist network. Standard generative phonological theories require serial application of rules, which results in derivations with numerous intermediate states. This is incompatible with the connectionist goals of psychological and biological plausibility, and may also hinder learnability. Lakoff's theory of cognitive phonology offers a solution to some of these problems by providing an alternative way to think about derivations and ordered rules, and by eliminating the need for right-to-left iterative rule application. On the other hand, Lakoff's proposal presents certain computational difficulties due to its appeal to Harmony Theory. We present a reformulation of cognitive phonology using a novel clustering mechanism that completely eliminates iteration and permits an efficient feed-forward implementation.

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

The views and conclusions contained in this document are those of the authors, and should not be interpreted as representing the official policies, either expressed or implied, of Hughes Research Laboratories, the National Science Foundation, the Office of Naval Research, or the U.S. Government.

I. Introduction

This paper reports on initial results of an effort to actually implement Lakoff's theory of cognitive phonology (Lakoff 1988a, 1988b, 1989) in a connectionist framework. Standard generative phonological theories requiring serial application of rules typically result in long derivations with numerous intermediate states. This is incompatible with the connectionist goals of psychological and biological plausibility. Lakoff's theory of cognitive phonology offers solutions to some of these problems by providing an alternative way to think about derivations and ordered sets of rules, and by eliminating the need for right-to-left iterative rule application.

We will begin by describing our "many maps" model and showing 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 distinct 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 constraints which turn out to be problematic to implement. 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 1989 Berkeley 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 all the well-formedness constraints. The major problem which arises here is that it is not at all clear how to implement this notion of maximal harmony.

An appeal to "harmony" skirts an important computational question: How is the network able to find the most harmonious state? That is, how do the P and F-levels automatically settle into representations that best meet all applicable constraints? Smolensky's Harmony Theory is based on simulated annealing performed by simple neuron-like computing elements. It's not clear how to express phonological operators, which may produce complex chains of insertions, deletions, and mutations, in a way that would be amenable to stochastic search by a neural net. Nor is it clear how constraints on phonological well-formedness could be encoded in a neural net. In other words, we don't know how to wire a device that could meet Lakoff's specifcations. We are not saying that it can't be done, only that we don't at the moment see how to do it.

Another problem with appeals to "harmony" is the processing time involved. Simulated annealing search requires many iterations of local state updates, at gradually decreasing temperatures, in order to find the most harmonious or "minimum energy" global state. Real neurons operate much too slowly to be performing this kind of search. Assume, for example, a rapid adult speech rate of ten phonemes per second, and a maximum neuron firing rate below 1000 Hertz. At 100 milliseconds per phoneme, a chain of fewer than 100 firings is all that is possible before the next phoneme must be articulated. Annealing would seem to require far more than this. One of our goals in seeking an alternative to harmony theory is to avoid relying on computational processes that are clearly incompatible with human biological constraints.

The specific implementation described here does not appeal to harmony theory, but rather is strictly deterministic and feed-forward 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 be implemented efficiently in a connectionist framework.

After describing the general properties of our model and how mappings between levels are 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, vowel harmony in Yawelmani, and voicing assimilation in Russian. 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.

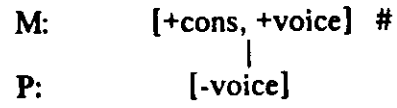
II. Many Maps Model

Lakoff's theory of cognitive phonology recognizes three distinct levels of representation: the morphemic level (M), the phonemic level (P), and the phonetic level (F). Generative phonological rules are replaced by 'constructions', which state well-formedness constraints within levels and correlations between levels. To illustrate, consider a process such as word-final devoicing in German, which would be accounted for by the following rule in the standard generative framework.

(1) [+cons, +voice] --> [-voice] / __#

In Lakoff's theory there would be a cross-level construction stated between the morphemic and phonemic levels (an M-P construction), as shown in (2). This construction would serve as a well-formedness constraint between levels, sanctioning the representations shown in (3).

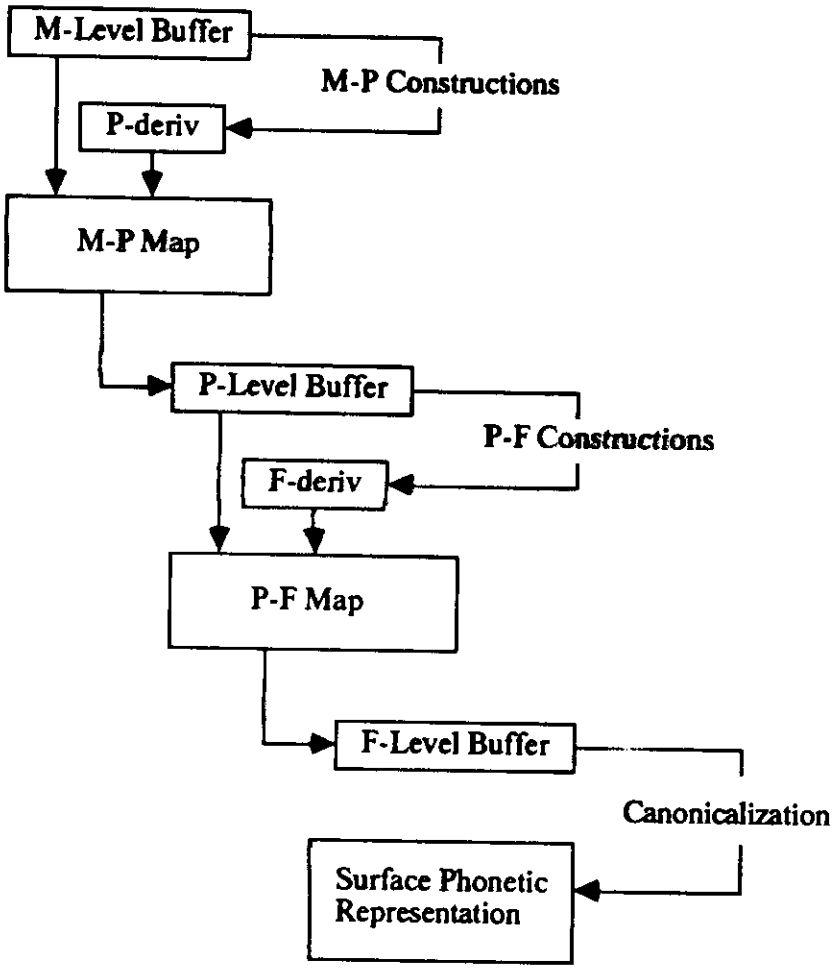
(2) German final devoicing:



(3) M: # r a d # 'wheel (sg.)'
P: r a t

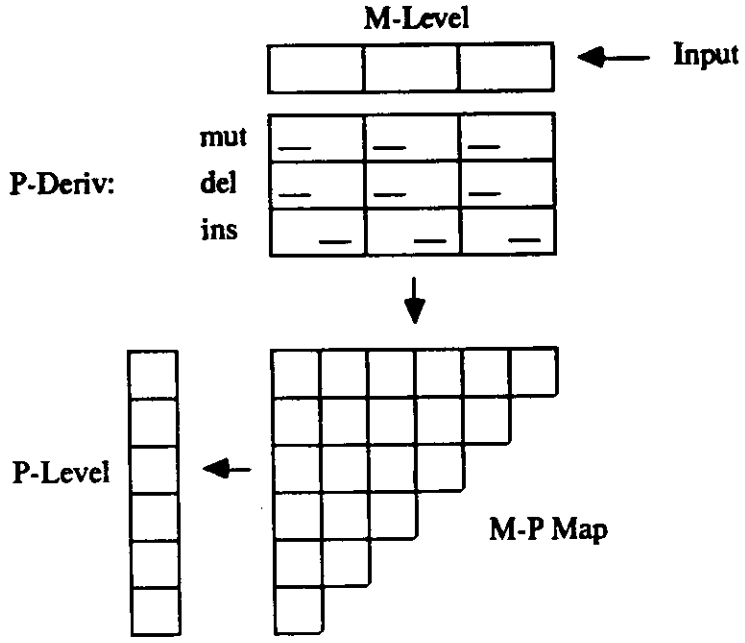
The remainder of this section will outline the general architecture of our 'many maps' implementation. The overall model has the following structure:

(4) The Model


M-Level Buffer
M-P Constructions
P-deriv
M-P Map
P-Level Buffer
P-F Constructions
F-deriv
P-F Map
F-Level Buffer
Canonicalization
Surface Phonetic Representation

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

(5) M-P Mapping

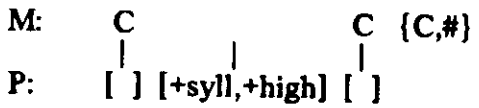


The M-level buffer contains phonemic segments. The P-deriv buffer describes the changes necessary to derive the P-level form of the utterance from the underlying M-level form. The input to the M-P map consists of M-level segments plus the changes recorded in P-deriv. The output of the matrix forms the 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 (cf. Lakoff 1989). 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 any two adjacent input segments. The significance of this constraint will be discussed later.

The function of the mapping matrix is to perform the indicated changes and 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 columns in the matrix, with the second column being activated in the case of insertions. If this column is empty, the input segments are adjacent in the output representation.

We will draw on isolated rules from Lakoff's analysis of Yawelmani for illustrative purposes. The full analysis, with data, will be presented later in the paper. Consider, first, epenthesis. Lakoff (1989: 26) states the epenthesis construction as:

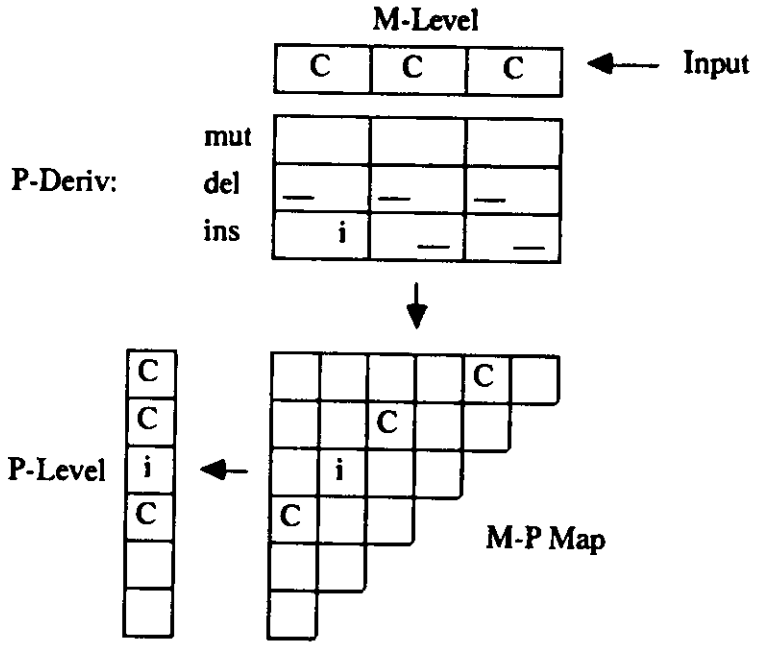
(6) Yawelmani epenthesis:



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 there is a default rule which will fill in the value for [back]; ultimately yielding [i].

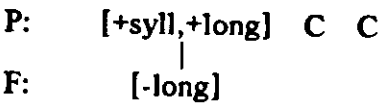
In our model, the insertion is accounted for in the following manner. If a string of three consonants or two word-final consonants appears in the input buffer (M-level), then epenthesis takes place. This is accomplished by recording the features [+syll,+high] (represented for convenience as [i]) in P-deriv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

(7) Insertion



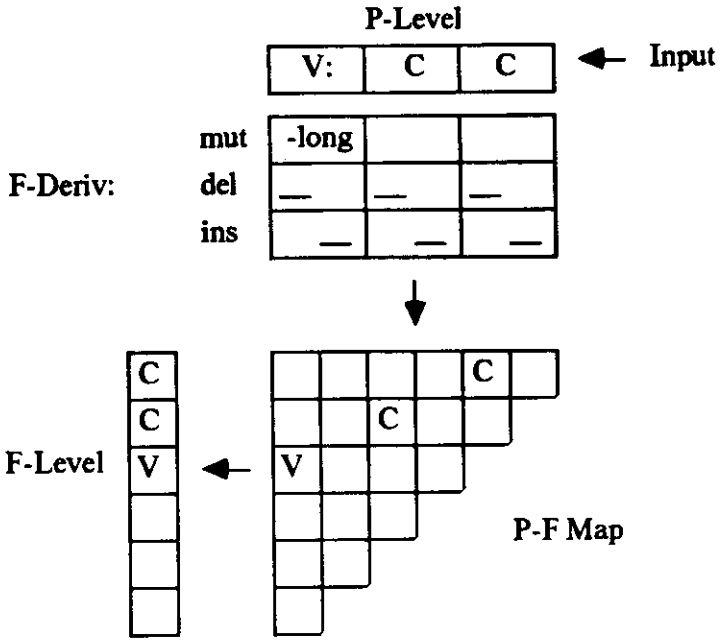
Vowel shortening in Yawelmani is an example of a mutation process. Lakoff (1989: 26) gives the following construction to account for alternations in vowel length. The general pattern is that long vowels are shortened when they occur before two consonants.

(8) Yawelmani shortening:



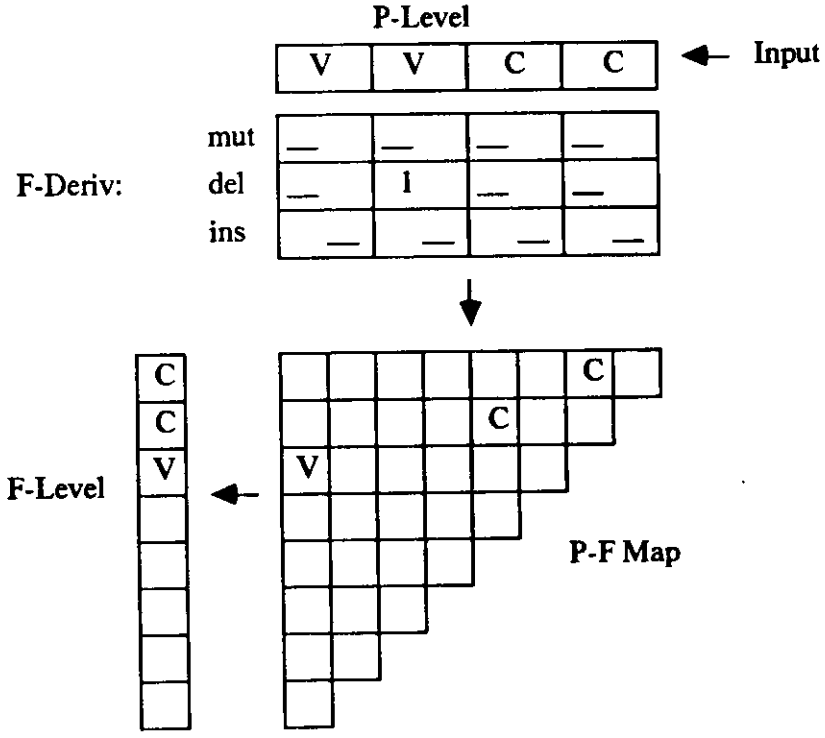
In our model, Lakoff's rule would be interpreted as a mutation process. In the P-F map, the appropriate change would be recorded in F-deriv, as illustrated below.

(9) Mutation



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:

(10) Deletion



In this case, since the deletion bit is turned on in F-deriv for the segment corresponding to the second half of the long vowel, no units are active in that column of 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.

In addition to the M-P and P-F maps, we assume a final "canonicalization" process that maps F-level representations onto well-formed phonetic segments of the language. One way to describe canonicalization is by a set of explicit rules such as (11) below. All canonicalization rules are context-free rules and merely serve the function of ensuring that the segments at the phonetic level conform to the phonetic constraints of the language. 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. Allowing this class of rules in no way compromises our position that there are no intra-level rules. These canonicalization rules 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.

As alluded to earlier, Lakoff's discussion of Yawelmani epenthesis (1989: 29) assumed the following rule of High Vowel Markedness to insure that the epenthetic vowel surfaces as [i]. Pointing out that markedness principles are required independently, this allows him to simplify the statement of the epenthesis rule (6). This is a typical canonicalization rule.

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

We have no doubt that underspecification (Archangeli 1984) 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. This is part of the function of the canonicalization rules. 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.

Although throughout this paper we refer to canonicalization as a rule-based process, this is in deference to linguistic convention, not a requirement of our model. While such rules can be implemented directly in connectionist circuitry, canonicalization might also be implemented using an associative memory, such as a Hopfield net (Hopfield 1982), in which there were no explicit rules. We are currently investigating this idea. It appears feasible because canonicalization is a strictly context-free (intra-segmental) operation. It involves no insertions or deletions, and it cannot feed or bleed other phonological processes.

III. Left-to-right vs. Simultaneous

In Lakofff'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 (12), with schematic examples in (13):

(12) [+syll, +long] --> [-long] / [+syll, +long] C_0 ___

(13)

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	[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 (1989: 12) states the following construction for Slovak:

(14) Slovak shortening

M: [+syll, +long] C_0 [+syll, +long]
|
P: [-long]

What is particularly significant about this case is that by stating the construction as in (14) 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-level condition of the construction and therefore must be short at P.

(15) 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 (16), with the environment at P. The consequence of this apparently simple difference is that only alternate vowels may shorten, as illustrated in (17).

(16) Gidabal shortening

M: [+syll, +long]
|
P: [+syll, +long] C_0 [-long]

(17) 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 1989: 13)

In fact, the analyses in (15) and (17) 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 were 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 rules 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 in (16) 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:

(18) Intra-level version of Gidabal shortening:

P: If [+syll, +long] C_0 X, then if X = [+syll] then X = [-long]

Thus, at least as it was originally formulated, Lakoff's 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. The general pattern is that vowels become round and back when following a round vowel of the same height (e.g. /-hin / surfaces as [hun] when following /u / and /-al / surfaces as [ol] when following /o /). 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 1989: 27):

(19) P: If [+syll,+rnd,αhigh] C_0 X, then if X = [+syll,αhigh], then X = [+rnd,+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 vowel harmony he gives the following 'derivation' for [do:sol] 'report (dubitative)', saying simply that: "the harmony constraint is met at P" (Lakoff 1989: 28).

(20) M: do:s+al
P: do:s+ol
F: do:s+ol

Here again, we don't know how to implement that kind of rule in parallel 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 (20) 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 to 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. First, vowel harmony must follow epenthesis because epenthesis both feeds and bleeds harmony. The epenthesis rule was given in (6), and is repeated below for convenience.

(21) Yawelmani epenthesis:

M: C C {C,#}
P: [ ] [+syll,+high] [ ]

Lakoff gives the example in (22) below to illustrate the feeding relation, with the epenthetic vowel undergoing harmony.

(22) /ʔugn+hin/ "drinks"
ʔuginhin epenthesis
ʔugunhin harmony on epenthetic vowel
ʔugunhun harmony on the final vowel
[ʔugunhun]

And the following example shows how epenthesis can block harmony, since otherwise the final [a] would be rounded to [o] (cf. (20)).

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

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 were analyzed as a P-F construction, then its interaction with the other P-F constructions, namely lowering and shortening, must be considered. Lakoff's lowering construction is stated in (24) and the shortening construction (8) is repeated below in (25) for convenience.

(24) Yawelmani lowering

P: [+syll, +long]
|
F: [-high]

(25) Yawelmani shortening

P: [+syll,+long] C C
|
F: [-long]

Lowering and shortening both have their environments at P-level, with changes at F-level, and therefore would potentially interact with vowel harmony if it is stated as a P-F construction. It is the lowering rule which is relevant in this case. In standard generative terms, harmony must precede lowering since lowering (incorrectly) bleeds harmony. Furthermore, lowering must precede shortening. The pattern is that all long vowels become [-high] by lowering. Then, shortening applies to the long vowel if it is followed by two consonants. Consider the following derivation, taken from Lakoff (1989: 26):

(26) / sudu:k'+hin /	"removes"
sudu:k'hun	harmony
sudo:k'hun	lowering
sudok'hun	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 (27) below, a P-F mapping with environment at P-level.

(27) Yawelmani harmony -- hypothetical version

P: [+syll, +rnd, αhigh] C_0 [+syll, αhigh]
|
F: [+rnd, +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.

(28) P: s u d u: k + h i n
| |
F: s u d o k h u n

There is an obvious problem with the harmony rule in (27), however, which is that with the environment stated at P-level it will not apply iteratively and so harmony cannot spread across more than one vowel. It will incorrectly predict that:

(29)

M:	?ugn+hin	
P:	?uginhin	by epenthesis
F:	?ugunhin	by harmony
	*[?ugunhin]	

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 (Touretzky 1989) the Many Maps model was 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 throughout the domain of the cluster, simultaneously recording changes in P-deriv for all elements of the cluster.

We are drawing heavily here on the insights of autosegmental phonology (Williams 1976, Goldsmith 1976, among many others), as will become clear shortly. In effect, our clustering mechanism will provide a means of identifying projections on phonological features (e.g. a vowel tier), the beginning of a cluster, and the elements within the cluster. In the following section we will discuss a case which involves clustering [-syllabic] elements. We tentatively restrict cluster types to these two projections: vocalic ([+syllabic]) and consonantal ([-syllabic]). Rules operating on clusters will have the following format, with unmarked parameters not being specified in individual rules. The rule format adopted here for processes operating on clusters is strikingly similar to that independently developed by Archangeli (1989) and Archangeli and Pulleyblank (1989). We do not believe that the differences are crucial for our purposes here. Details of the various aspects of the rule format will become clear as we discuss particular phonological processes.

(30) Cluster Operations

Cluster type:	[+syllabic] or [-syllabic]
Direction:	right-to-left or left-to-right (unmarked)
Trigger:	[feature(s)]
Element:	[feature(s)]
Range:	bounded or unbounded (unmarked)
Change:	[feature(s)]

First, the mapping architecture already described is sufficient for establishing a vowel projection in Yawelmani -- only those segments which are [+syllabic] are mapped to the output level of a vowel cluster. 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 in a cluster (but not to the trigger).

(31) Yawelmani vowel harmony — P-F mapping:

Cluster type:	[+syllabic]
Trigger:	[+rnd, αhigh]
Element:	[αhigh]
Change:	[+rnd]

The clustering mechanism has a left-to-right precedence 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 agrees with the trigger in the specification for the feature [high]. 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.

(32)

	i	u	i	i	a
trigger		+			
element			+	+	

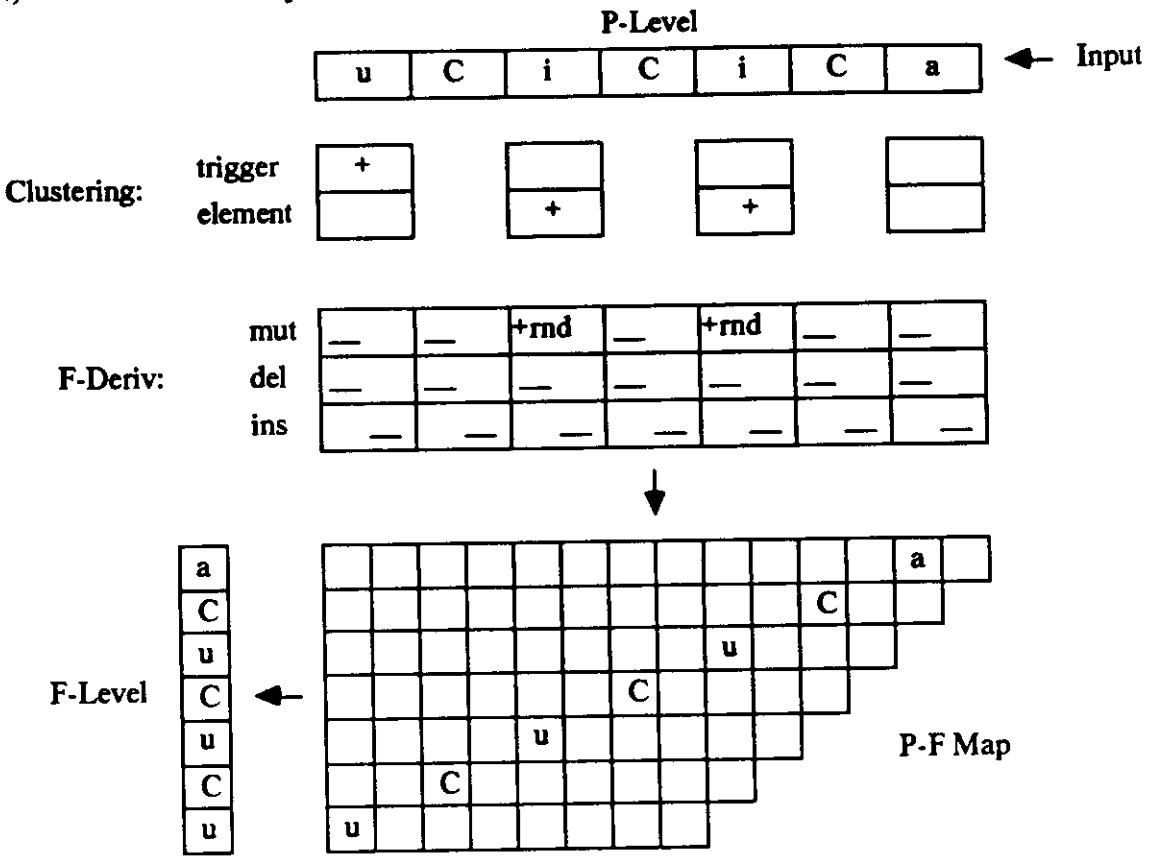
We are assuming two general (universal) conventions on clustering. First, triggers cannot also be elements, and second, there is a preference for segments to be elements rather than triggers, other things being equal. That is, in the unmarked case, clusters are unbounded and/or as large as possible, given the restrictions of the cluster. This will mean that in a string like /u i i u i/, 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 both the trigger ([o]) and [a] are [-high], because elements must be adjacent to triggers or other elements.

(33)

	o	i	a
trigger	+		
element			

Thus, vowel harmony in Yawelmani 'iterates' in cases like (32) because sequences of high vowels have been clustered. This minor 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.

(34) Yawelmani harmony



The same clustering mechanism may be used to explain the alternating vowel length patterns of Gidabal 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:

(35) Slovak shortening — M-P mapping:

Cluster type:	[+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:

(36)

	V: V: V: V:
Trigger:	+
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. It is worth noting here that this distinction between unbounded (Slovak) and bounded/alternating (Gidabal) processes arises in discussions of stress patterns as well. We have chosen to draw on the insights of research in metrical theory (Hayes 1980, Halle and Vergnaud 1987) and have characterized the Gidabal shortening process, where the element must be adjacent to the trigger, as being 'bounded'. This is the relatively marked

option. The default, as in Slovak, is where the cluster may be indefinitely large or unbounded. While we have not attempted to implement any analyses of stress patterns, we are hopeful that our clustering mechanism may be straightforwardly extended into this domain.

(37) Gidabal shortening — M-P mapping:

Cluster type:	[+syllabic]
Trigger:	[+long]
Element:	[+long]
Range:	bounded
Change:	[-long]

(38) Gidabal vowel clusters

	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 sequentality 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. In Lakoff's Gidabal solution (if one discounts the appeal to Harmony), rules must fire sequentially from left to right in order to achieve the desired outcome.

V. Consonant Clustering

Our view of clustering as a primitive operation was originally inspired by a desire to give a parallel rule analysis of voicing assimilation in Russian consonant clusters (Hayes 1984, cited in Wheeler 1988.) Word-final consonants are devoiced in Russian. Voicing assimilation causes the voicing feature of the last obstruent in a cluster to spread leftward to the remaining obstruents. However, /v/ is not treated as an obstruent trigger by this rule; it does not trigger voicing assimilation. Instead, the next rightmost obstruent controls the voicing. (/v/ is still subject to the word-final devoicing rule, though.)

To account for this pattern, Hayes suggests that /v/ be represented underlyingly as the sonorant /w/. If word-final devoicing applies, it produces a voiceless sonorant, represented here as /W/. Voicing assimilation is then triggered by the rightmost obstruent in a cluster, which excludes /w/ and /W/ since, unlike /v/ and /f/, they are not obstruents. After voicing assimilation, a rule of W Strengthening turns /w/ into [v] and /W/ into [f]. Finally, since voicing assimilation may have produced other voiceless sonorants in the cluster, such as /L/ or /M/, these must be repaired by a rule of Sonorant Revoicing.

(39) Final Devoicing

C --> [-voice] / __ #

Voicing Assimilation

In a consonant cluster, assign the voicing of the last obstruent to all consonants on its left.

W Strengthening
[C,-cons,+labial] --> [-son]

Sonorant revoicing
[+son] --> [+voice]

Our solution has much in common with Hayes' approach, including representing [v] underlyingly as /w/. We analyze Final Devoicing as an M-P construction; our Voicing Assimilation rule is a P-F construction. A /w/ in word-final position will devoice (represented here as /W/) like all other [-syllabic] segments, but will not trigger voicing assimilation since it is [+sonorant].

(40) Final Devoicing

M: [-syll, +voice] #
|
P: [-voice]

Voicing Assimilation — P-F mapping:
Cluster type: [-syllabic]
Direction: right-to-left
Trigger: [-son, αvoice]
Element: []
Change: [αvoice]

Because the cluster type is consonantal, clusters will automatically be broken by vowels, and the feature [-syllabic] may be omitted from triggers and elements. First, consider a case where Final Devoicing feeds Voicing Assimilation:

(41) M: v i / z g / `scream'
| Final Devoicing
P: v i z k
| Voicing Assimilation
F: v i [s k]

Next, a case where Final Devoicing applies, but the segment does not trigger Voicing Assimilation since /w/ is [+sonorant]. Hayes' rule of W-Strengthening may be formulated as the P-F construction in (42).

(42) Russian W-Strengthening

P: [-cons, +son, +labial]
|
F: [-son]

(43) M: t r e / z w / `sober'
| Final Devoicing
P: z W
| W-Strengthening
F: t r e [z f]

Finally, an example where two P-F level constructions apply simultaneously. First, Voicing Assimilation devoices the medial /w/ and /z/ (i.e. /w/ --> /W/ and /z/ --> /s/) and, in addition, that same /w/ segment undergoes W-Strengthening, yielding [f] at F-level.

(44)

M:	/b e z w p u s k a/	'without admission'
P:	b e z w p u s k a	Voicing Assimilation, W-Strengthening
F:	[b e s f p u s k a]	

A case where Sonorant Revoicing applies as part of the canonicalization (C) is shown below. As before, this is a context-free process, and has the effect of mapping segments at the F-level to the 'nearest' well-formed phonetic segment of the language.

(45)

M:	/i z m c e n s k a/	'from Mcensk'
P:	i z m c e n s k a	Voicing Assimilation
F:	i s M c e N s k a	Sonorant Revoicing
C:	[i s m c e n s k a]	

The significant result here is that it is possible to account for the extremely complex set of facts related to devoicing and voicing assimilation in Russian without having to introduce notions like iterative application of rules or extrinsic rule ordering. The analysis draws solely on the independently motivated notions of clustering and mappings between levels and thus poses no problems for the ultimate goal of implementing phonological processes in a connectionist model.

VI. 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 umlaut, 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-umlaut are both M-P constructions. The complex interaction is accounted for by assuming that the environment for u-umlaut may hold at either M- or P-level. The constructions are stated as follows:

(46) Syncope (Lakoff 1989: 21)

M:	V D + V	(D = [-syll, +cor, +lax])
P:	Ø	

(47) Umlaut (Lakoff 1989: 21)

M:	[+syll, +low, +back]	
P:	[-low, -back]	C_0 u

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 [ö]. 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:

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

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

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

(50) Vowel Reduction (Lakoff 1989: 23):
P: If [+syll,-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:

(51) M: fátnað+um
| |
P: fötnuðum

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, syncopated vowels will not interfere. Thus, we have a slightly different picture of the derivation of [ölnum] than Lakoff offers. In our solution, /a/ does not become /ö/ at P-level, but at F-level.

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

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 /r/. Since both rules have their environments at P-level and changes at F-level, the epenthetic [u] is invisible to u-umlaut.

(53)

M:	dag+r	
P:	dag r	
	|	u-epenthesis
F:	dagur	

The derivation of forms like [fötnuðum] in (51) 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, since clustering proceeds from right to left.

(54) Icelandic u-umlaut

Filter:	[+syllabic]
Direction	right-to-left
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 feature specification 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 canonicalization rule.

(55) If [+syll, -stress, -low, -back, +round], then [+back, +high]

This adjustment is forced by the fact that there are no instances of an unstressed [ö] in the surface phonetic representations of Icelandic. Thus, unstressed /a/ corresponding to an /ö/ 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 (55) 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, canonicalization 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 one 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.

VII. Relationship to Connectionism

Some of our colleagues have asked us what it is about this work that makes it "connectionist". Our theory is a connectionist theory because its mechanisms have simple computational instantiations. In other words, they can be implemented in a straightforward way using neuron-like threshold logic units. This is a powerful constraint, and one which may have interesting, unanticipated consequences.

One example is the restriction we derived on insertions. The P-deriv and F-deriv buffers only allow room for inserting one new segment between any two adjacent input segments. The reason for this constraint is that if two rules applying in parallel each tried to insert a segment, say A and B, between the adjacent input segments P and Q, the result could be either PABQ or PBAQ. Unless one was willing to devise some type of precedence or ordering convention for insertion processes, no doubt requiring additional special circuitry, there would be no way for the two rules to interact in a predictable way. Therefore we adopted the constraint that multiple segments may never be inserted between segments adjacent at the preceding level. (We know of morphological insertion processes that violate this contraint, but it is only intended to apply to phonology.) In the experience of the first author and several other expert phonologists present at the Berkeley workshop, no human language violates this insertion constraint. Even though the constraint appears to be universal, it was not previously recognized by phonologists. Thus we have an example of how a computational investigation of phonological processes can lead to useful insights.

Although ours is a connectionist theory, so is Lakoff's. His theory relies on a powerful computational process, 'harmony', that involves feedback and stochastic search, while our theory uses only simple feed-forward connections and deterministic units. In order to do the same job with a simpler architecture, we were forced to invent special circuitry, such as the clustering circuitry, to make up for the fact that there are no intra-level constraints. It is possible that Lakoff's theory will also require specialized circuitry to support particular phonological processes, but this is unknown, as his theory has not yet been implemented.

Some connectionist architectures have powerful learning procedures. Another reason for pursuing a connectionist approach, then, is the hope that someday these models may be able to acquire phonological rules automatically, by exposure to surface forms. To the best of our knowledge, no phonologist has seriously tackled the problem of learnability yet. Perhaps this is because the problem is so hard for generative rules. In a properly-constrained cognitive phonology system, though, the rules are simpler, and rule induction might be feasible.

VIII. Conclusion

Our goals here have been twofold. We have focused on describing our connectionist implementation of Lakoff's theory of cognitive 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 easily implementable solution. The clustering mechanism which we have proposed here offers an alternative to both iteration and harmony.

Our constraint on constructions is actually stronger than just saying that only cross-level constructions are permitted. Given Lakoff's formalism, in principle there is nothing to prevent a cross-level construction from being stated with all or part of the environment at the output level. This is in fact how Lakoff originally stated the Gidabal shortening construction (16), though as pointed out earlier this rule could also have been stated as a simple P-level rule (18). We suggest that constructions be limited to those which may be stated as cross-level constructions with environments solely at the input level.

In our implementation of phonological processes we have consistently attempted to draw on the insights of current research in theoretical phonology. It is thus our hope that our model will be of interest to linguists as well as connectionists. Our mapping and clustering mechanisms offer a means of implementing autosegmental processes in general. While we initially focused on the vowel tier, we also showed how voicing assimilation in Russian can be accounted for in this model without iterative application of rules. 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 and intend to focus on questions pertaining to underspecification, markedness, and rule application in our future research.

Acknowledgements:

This work was sponsored in part by a contract from Hughes Research Laboratories, by National Science Foundation grant EET-8716324, and by the Office of Naval Research under contract number N00014-86-K-0678. We thank George Lakoff for encouragement and many helpful explanations, and Gillette Elvgren III for his work on the simulations.

References:

Archangeli, D. (1984) "Underspecification in Yawelmani Phonology and Morphology". Doctoral dissertation, M.I.T.

Archangeli, D. (1989) "Parameters in phonological rules". Paper presented at the Workshop on Constraints vs. Rules in Phonology, University of California at Berkeley, May 1989.

Archangeli, D. and D. Pulleyblank (1989) "Flying in the face of convention: parameters of rules". Paper presented at the Conference on Parametric Theories of Phonology, University of Arizona, July 1989.

Goldsmith, J. (1976) "Autosegmental Phonology". Doctoral dissertation, M.I.T.

Halle, M. and J.-R. Vergnaud (1987) *An Essay on Stress.* Cambridge: MIT Press.

Hayes, B. (1980) "A Metrical Theory of Stress Rules". Doctoral dissertation, M.I.T.

Hayes, B. (1984) "The phonetics and phonology of Russian voicing assimilation". in M. Aronoff and R.T. Oehrle (eds.), *Language Sound Structure.* Cambridge: MIT Press. Pages 318-328.

Hopfield, J. J. (1982) "Neural networks and physical systems with emergent collective computational abilities". *Proceedings of the National Academy of Sciences, USA, 79.* Pages 2554-2558.

Lakoff, G. (1988a) "A suggestion for a linguist with connectionist foundations". In D. Touretzky, G. Hinton, and T. Sejnowski (eds), *Proceedings of the 1988 Connectionist Models Summer School,* San Mateo, CA: Morgan Kaufmann Publishers. Pages 301-314.

Lakoff, G. (1988b) "Cognitive phonology". Draft of paper presented at the LSA Annual Meeting, December 1988.

Lakoff, G. (1989) "Cognitive phonology". Draft of paper presented at the Workshop on Constraints vs. Rules in Phonology, University of California at Berkeley, May 1989.

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.

Touretzky, 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.

Wheeler, D. (1988) "Consequences of some categorially motivated phonological assumptions". In R.T. Oehrle, E. Bach and D. Wheeler (eds), *Categorial Grammars and Natural Language Structures.* Dordrecht: D. Reidel Publishing Company. Pages 467-488.

Williams, E. (1976) "Underlying tone in Margi and Igbo". *Linguistic Inquiry* 7.3. Pages 463-484.