Title

Towards a Connectionist Phonology: The "Many Maps" Approach to Sequence Manipulation

Journal

Proceedings of the Annual Meeting of the Cognitive Science Society, 11(0)

Author

Tourestzky, David S.

Publication Date

1989

Towards a Connectionist Phonology: The "Many Maps" Approach to Sequence Manipulation

David S. Touretzky

School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

Abstract: Lakoff's new theory of cognitive phonology appears to be free of the rule ordering constraints that make generative rules computationally awkward. It uses a multilevel representation for utterances, to which multiple rules may apply in parallel. This paper presents the first implementation of Lakoff's proposal, based on a novel "many maps" architecture. The architecture may also explain certain constraints on phonological rules that are not adequately accounted for by more abstract models.

Linguists established long ago the value of describing phonological processes in terms of formal symbolic rules, but they have steadfastly refrained from speculating about the nature of representations in speakers' heads. Rumelhart and McClelland (1986) argue against the neuropsychological reality of rules. Pinker and Prince (1988) offer persuasive counterarguments. The current reaction against rule-based accounts of low-level cognitive phenomena, phonology in particular, is no doubt strengthened by the computational awkwardness of classical generative phonological rules. Constraints on their order of application force the rules to act sequentially and, in some cases, cyclically. In contrast, Rumelhart and McClelland's PDP model of the phonology of English past tense formation maps input patterns to output patterns directly, in one parallel step. It is, despite its weaknesses, computationally sleek.

In 1988 George Lakoff published a new theory of "cognitive phonology" in which parallel rules apply everywhere simultaneously (Lakoff, 1988a, 1988b). Cognitive phonology is therefore free of the cycles and rule ordering constraints that mar earlier, generative theories. Lakoff described his theory as founded on connectionist principles, but did not specify how it should be implemented. The solution is non-obvious, because cognitive phonology relies on a multi-level mapping representation in which insertions, deletions, and mutations all take place at once.

This paper presents the first working implementation of Lakoff's theory. It uses a novel "many maps" architecture to manipulate sequences of phonemes at multiple levels, and to support abstractions such as the "vowel tier" required by some rules. I will begin by reviewing Lakoff's analysis of a Mohawk problem posed in (Halle & Clements, 1983), and then show how the "many maps" model implements Lakoff's solution. Finally I address the question of why one would want to have rules in a connectionist model. I will argue that the simplicity and highly constrained nature of phonology may be a consequence of humans' using a sequence manipulation architecture similar to the one described here.

Six Rules for Mohawk Speakers

Halle and Clements give six generative rules for deriving the Mohawk word /yɔ́krege?/. ("I will push it") from its underlying form /ye + ʌ̃k + hrek + ?/. (It may aid understanding to look at Lakoff's solution first; see Figure 1.) We will consider four of these rules here:

Epenthesis: ∅ --> e / C _ ? #
Stress: V --> [+stress] / _ C_0 V C_0 #
Vowel omission: V --> ∅ / _ + V
Intervocalic voicing: C --> [+voice] / V _ V

The epenthesis rule inserts /e/ between a consonant (/k/ in this example) and a word-final glottal stop /?/. The stress rule assigns stress to the penultimate vowel in a word. Notice that in the example /ʌ́/ is penultimate in underlying form but antepenultimate at the surface, due to epenthesis. Thus, the stress assignment rule must be applied prior to epenthesis in order to stress the correct vowel. The intervocalic voicing rule voices a consonant if it appears between two vowels; it changes /eke/ to /ege/. But the second /e/ was inserted by epenthesis; therefore intervocalic voicing must not be applied until after epenthesis. The vowel omission rule deletes the leftmost /e/ in the underlying form, since it precedes another vowel. Evidence from other Mohawk words shows that vowel omission applies before stress assignment. (If it didn't, the rules could perhaps assign stress to a vowel and then delete it, leaving no vowel stressed.)

In the classical account these four rules are totally ordered: vowel omission precedes stress assignment, which precedes epenthesis, which precedes intervocalic voicing. Each of these rewrite rules modifies the "current" derivation, producing a new one. When all the rules have applied, what's left is the surface form of the word.

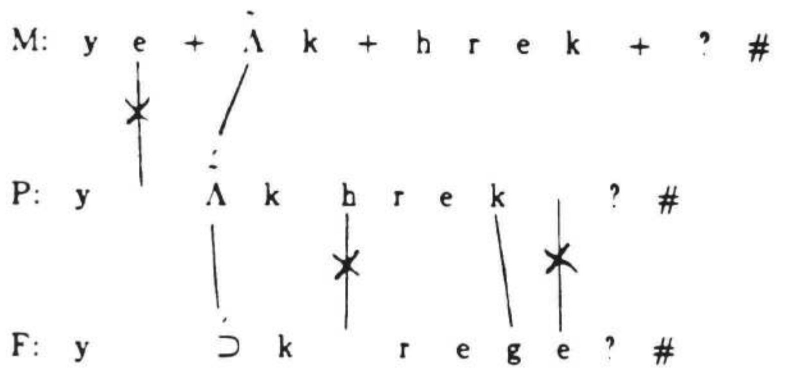

Figure 1: Lakoff's cognitive phonology derivation of the Mohawk word "I will push it."

Lakoff's analysis replaces the sequential rewrite rules with mapping constraints that all apply in parallel. There are three levels of representation: M (morphemic), P (phonemic), and F (phonetic). Sequences at M level are by default simply copied to P level. But M-P constraints can alter the mapping, causing changes in the P-level representation. Intra-level constraints may also affect the representation at P level. The combined effect of M-P and P-level constraints can be seen in the middle line of Figure 1: the first /e/ has been deleted and the penultimate vowel has been stressed. A second mapping takes P-level representations to F-level representations via a combination of P-F and F constraints. At F level we see that the epenthetic /e/ has been inserted and, consequently, the /k/ has been voiced.

Lakoff's solution elegantly answers a number of phonological questions which, unfortunately, we cannot afford to raise here. Elegance aside, though, its implementation in connectionist hardware is problematic. The major problems that arise are: how to efficiently implement insertion, deletion, and mutation operations when several occur in parallel; how associations between corresponding segments at different levels can be maintained, since levels may have varying numbers of elements; and how rules can apply everywhere at once in the input buffer. The "many maps" architecture provides solutions to these problems.

How to Build a Map

As a prelude to discussing the full "many maps" implementation I will describe the workings of a single map. Figure 2 shows the P-level map in the context of the Mohawk example. The input to this map comes from two buffers: M-level and P-deriv (P derivation). The output is the P-level representation of the utterance. The M-level buffer, which is read-only, contains the underlying form of the utterance. Segments are shifted into the buffer from the right, and discarded when they reach the left edge. M-level segments are by default mapped to identical segments at P-level. However, each M-level segment has a slot in P-deriv for describing changes that can be made to it if some rule requests. Three types of changes are supported: mutation, deletion, and insertion. Deletion of an M-level segment means blocking its appearance at P-level. Mutation maps the segment to a segment with slightly different features at P-level. Insertion causes a new segment to appear at P-level to the right of the M-level segment. (Insertion to the left could also be supported, but was omitted to simplify the simulation.) In the figure, the M-level /e/ is marked in P-deriv for deletion, and the /ʌ̃/ is to be mutated by adding stress. Thus the M-level sequence /yeʌ̃k/ appears as /yʌ̃́k/ at P-level.

The upper-diagonal matrix in the figure represents an array of connectionist mapping units. When one of these units is active (shown by a segment appearing inside it), the segment in that input column is copied to the corresponding output row. At the same time, any mutations to the segment that were requested in P-deriv are made.

The units in the mapping matrix are subject to lateral inhibition. At most one unit can be on in each row and each column. The inhibition is asymmetric, so that when choosing which row an input segment should map to, the model prefers to fill higher rows first. In addition, when choosing which segment should appear in a row, the model prefers to select the rightmost segment available. This ensures that the ordering of M-level segments is preserved at P-level, and that the P-level representation always appears right-justified in the buffer with no gaps where M-level segments are deleted, and no collisions where new segments are inserted.

Consider first the fate of the M-level /k/. The active square in the first row of the matrix shows that this segment is mapped to the rightmost position in the P-level buffer. Since this unit is fully active, no other unit can come on in that row or in that column. The /ʌ̃/ is mapped to the second row; simultaneously it is stressed, as specified in the mutation part of P-deriv. The /e/ is marked for deletion in P-deriv. Deletion is accomplished by inhibiting all the units in that column of the matrix, thereby preventing the segment from mapping to any row of P level. Thus the /y/, which is the fourth M-level segment counting from the right, appears as the third segment at P-level. The M-P mapping is computed in parallel (in fact, in constant time), independent of the number of segments in the buffer or the number of insertions and deletions to be performed.

Note that M-level segments are positioned over every other mapping column. The intervening columns are reserved for insertions. If an insertion is specified in P-deriv, the segment to be inserted will be mapped to the next available row, just as an M-level segment would be.

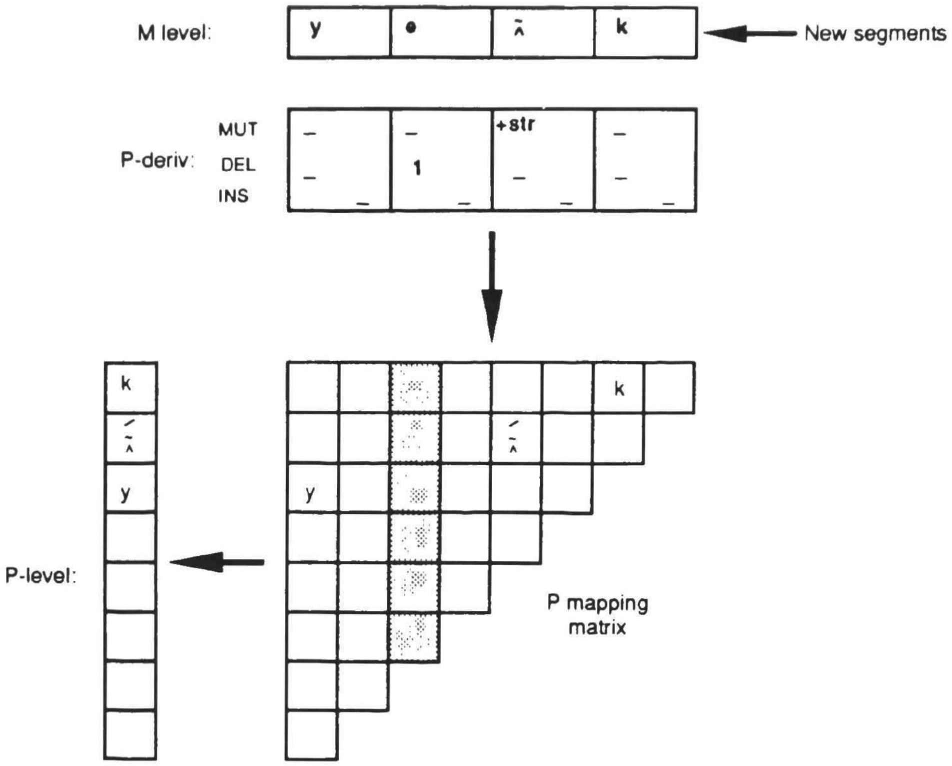

Figure 2: The map that derives P-level representations.

How Rules Work

One of the strengths of cognitive phonology is that rules may locate their environments at one level and their actions at another. Thus the application of an M-P rule does not affect the environments of other M-P rules. It does, however, affect the environments of P and P-F rules.

M-P rules are implemented by connectionist units that take their inputs from the M-level buffer and have output connections to P-deriv. After a new segment is shifted into M-level, M-P rules may cause some P-deriv units to change state, thereby recording a change the rules wish to make in the mapping. P-deriv units maintain their states indefinitely unless disturbed by rule units, thereby serving as a memory of accumulated changes. Each time P-deriv is modified by some M-P rule, the mapping matrix re-derives the P-level representation from the M-level and P-deriv buffers. When the M-level buffer is shifted left to accomodate the next incoming segment, the contents of P-deriv are also shifted left to maintain registration with the M level.

Pure P-level rules are trickier to implement than M-P rules, because they take their inputs from the *output* of the mapper. For example, suppose a purely P-level rule wanted to devoice the /y/ in Figure 2. This segment appears at position three at P level, but it is in position four at M level due to the deletion of a preceding segment. In order for P-level rules to record their changes in the correct P-deriv segment, they must invert the M-P mapping to align their changes with the M-level segments. The circuitry for this is straightforward. The state of the mapping matrix used to produce the current P-level representation provides the necessary information to invert the map.

Since P rules apply to their own outputs, they can feed each other, and there is even a possibility of long rule chains. Here is a simple example. In this implementation of Lakoff's Mohawk solution, epenthesis and intervocalic voicing are both implemented as F-level rules. Even though the rules are unordered, the former feeds the latter, so they will have to fire sequentially. The existence of rule chains appears to prevent the sort of parallel processing that cognitive phonology strives for. However, chunking can be used to automatically collapse a chain of intra-level rules into one complex rule, and thus regain the parallelism. This has been demonstrated for abstract phonological rules in (Touretzky, 1989).

Rules cannot be tied to fixed buffer positions because of feeding relationships. Suppose the intervocalic voicing rule were aligned with the right edge of the buffer, i.e., it looked at the rightmost three segments. When it saw `/eke/` it would produce `/ege/`. But if the buffer initially holds `/ek?/`, epenthesis will produce `/eke?/`, and so the first appearance of the `/eke/` fragment will not be aligned with the right edge of the buffer. It will be "downstream" of its standard position.

To make rules position-independent, we hypothesize that all rules are independently motivated and hence can be learned in standard (right-aligned) position by a primary rule module. Secondary rule modules are introduced at successive positions downstream. Their input and output connections are forced by link equality constraints to mimic the behavior of the primary module. All the rule modules operate in parallel, and their requests for changes are combined and recorded in P-deriv. In this way we achieve position-independence without having to supply examples of every rule firing in every position.

The "Many Maps" Architecture

At a minimum, cognitive phonology requires two maps: one for P-level representations and one for F-level. The F-level map is similar to the P-level discussed above, except it takes input from three buffers: M-level, P-deriv, and F-deriv. F-level representations are derived directly from M-level by merging the P-deriv and F-deriv changes at the input to the F mapping matrix. In the case of conflict, F-deriv changes are given priority. See Figure 3. This approach allows the model's multiple maps to operate independently instead of increasing the latency with each new level of representation. (This idea is due to Gillette Elvgren.)

Two types of rules influence the contents of F level. P-F rules have their environments at P and their actions at F; their actions are recorded in F-deriv by first inverting the mapping specified by the P matrix to align them properly with M-level segments. (The M-level segments and P-deriv and F-deriv changes are all kept in strict registration.) Since F-level rules have their environment at F, their actions are recorded in F-deriv by inverting the mapping specified by the F matrix. Figure 4 shows the relationships between the various rule types.

In Mohawk, the stress rule is most easily implemented by placing its environment at yet another level: a P-level vowel tier containing only vowels and word boundary markers. This allows the stress rule to look for the pattern `VV#` in the vowel tier and stress the penultimate vowel. (This solution was suggested by Deirdre Wheeler. Other evidence for an independent vowel tier is cited in (Goldsmith, 1989).) The map that extracts the P vowel tier takes inputs from the same M-level and P-deriv buffers as the regular P-level map, but only vowels appear in its output; consonants are left unmapped. The P vowel tier map operates completely in parallel with the regular P-level map.

We have successfully applied the "many maps" architecture to additional examples Lakoff chose from Slovak, Gidabal, and Lardil. Other languages will require other specialized maps. We expect, though,

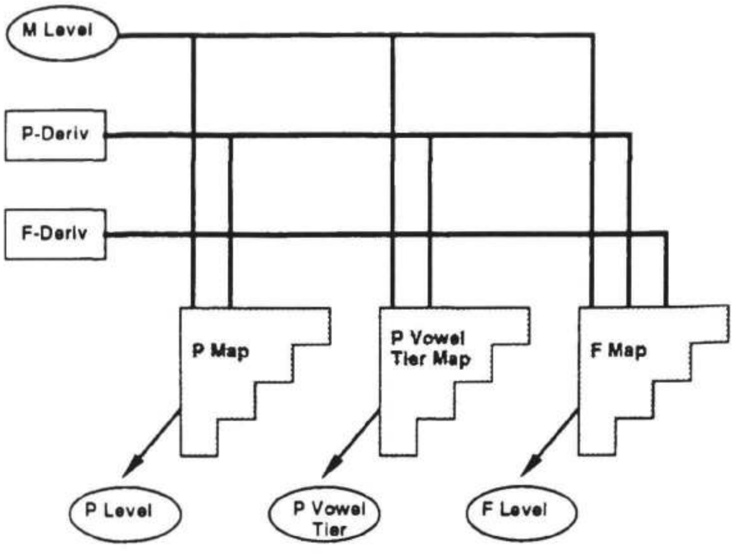

Figure 3: Inputs and outputs of the P-level, F-level, and P-vowel-tier maps. All maps operate independently, and in parallel.

that these can all be built from similar hardware. Perhaps language learners are born with a collection of such maps at their disposal, which are then trained to extract whatever features are salient in the linguistic environment.

Discussion

Phonology continues to be a rich and promising domain for connectionist investigations of language. It is simpler and less plagued by the special cases and exceptions that complicate syntax and morphology, so there is a better chance of finding a complete solution. Another advantage of phonology is its quasi-linear structure, which facilitates experimentation with parallel distributed processing techniques. The PDP approach isn't currently as well suited to manipulating hierarchical structures such as syntactic trees.[1]

The present model is not without limitations. It deals only with segmental phonology; no attempt is made to include morphology. (In contrast, the Rumelhart and McClelland verb learning model combines morphological and phonological processing in a single layer of weights.) Also, currently the model does not represent syllable structure. Certain types of phonological rules therefore cannot be expressed. This is an area where further work is in order.

The mapping architecture does not permit more than one segment to be inserted between segments adjacent at the previous level. Morphology sometimes requires multi-segment insertions, but it appears that phonology does not. If this observation holds true, it is a significant constraint on phonological machinery. The model provides an achitectural explanation for it, unlike more abstract phonological models which ignore implementation issues. Finally, the mapping matrix does not support metathesis (switching of segments) as a primitive operation. Considering the controversial and still unresolved status of metathesis in linguistic theory, we are in no rush to add it.

---

[1] However, Touretzky (1986), Hinton (1988), and Pollack (1988) offer some hope for handling hierarchical structures.

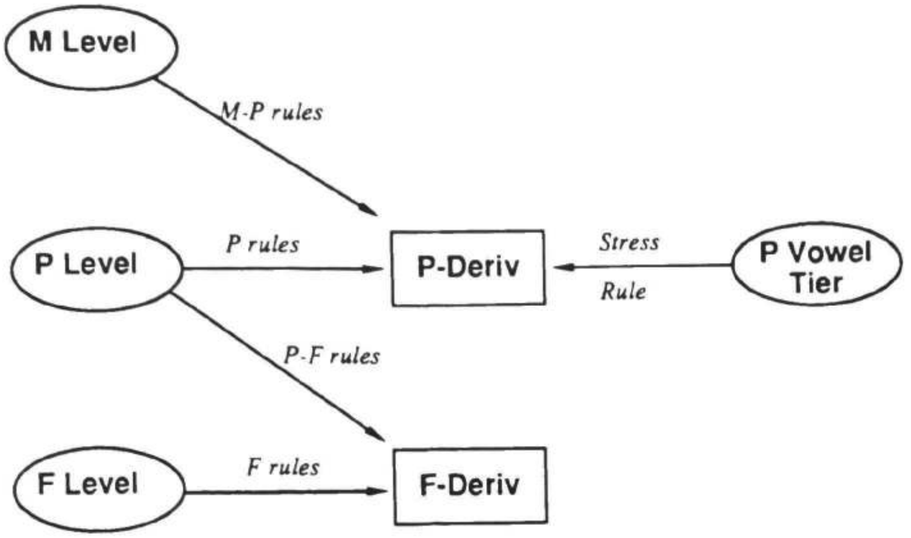

Figure 4: The different types of cognitive phonology rules that relate representations at various levels.

The "many maps" model might be improved by switching to an autosegmental representation with separate phonemic, skeletal, and tonal tiers, as in (Goldsmith, 1989). In fact, the original inspiration for the mapping matrix came from wondering how inter-tier association lines in autosegmental phonology could be represented in a connectionist network. A mapping matrix representation seems particularly appropriate for inter-tier rules, such as the tone-shift rules Goldsmith describes in various Bantu languages.

On the Reality of Rules

Why should human phonology be so regular and tightly constrained? It is amazing that this level of language can be described by classical generative rules which affect only a single segment each. This mode of description is effective, but it remains computationally inelegant. On the other hand, as Pinker and Prince point out, a connectionist architecture that directly maps input sequences to output sequences can perform outlandish transformations never seen in human language, such as reversing all the phonemes of a word.

There appear to be more modest sorts of transforms that are absent from the human repertoire. For example, no language methathesizes non-adjacent segments. Consonants are never changed to vowels, and vice versa. And harmony and assimilation phenomena always spread features from one edge of a cluster to the other, never from the interior outward. To be successful, a connectionist theory of phonology should motivate these constraints by providing computational explanations for them.

We can begin to account for constraints on phonology by adopting a universal, genetically-specified sequence manipulation machine that, like the "many maps" model, operates in parallel but can perform only a limited set of transformations. The function of linguistic rules is to operate this machine — to "press the right buttons at the right times." A speaker's linguistic knowledge does not directly modify sound sequences as in the Rumelhart and McClelland model; it modifies sequences only indirectly, by controlling this built-in machinery.

An input representation plus a discrete set of symbol manipulation primitives defines a *rule system.* If such a system underlies human phonology, then even if speakers do not have symbolic rule representations in their heads, they truly do use rules, as opposed to merely saying their behavior can be described by rules. Classical phonology concerns itself with the regularities of this rule system. Connectionist phonology attempts to ground the system in the design of the sequence manipulation machine, for it is from there that the rule system emerges.

Acknowledgements

I am most grateful to George Lakoff for sharing his work on cognitive phonology with me, and for taking the time to answer so many questions. I also thank Deirdre Wheeler for stimulating conversations, and for the linguistic guidance she has provided during our weekly meetings. Gillette Elvgren programmed the Mohawk simulation, and has made valuable refinements to the mapping architecture.

This work was sponsored 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.

References

Goldsmith, J. (1989) *Autosegmental and Metrical Phonology: A New Synthesis.* Basil Blackwell.

Halle, M., & Clements, G. N. (1983) *Problem Book in Phonology.* MIT Press.

Hinton, G. E. (1988) Representing part-whole hierarchies in connectionist networks. *Proceedings of the Tenth Annual Conference of the Cognitive Science Society,* pp. 48-54.

Lakoff, G. (1988a) A suggestion for a linguistics with connectionist foundations. In D. Touretzky, G. Hinton, and T. Sejnowski (eds.), *Proceedings of the 1988 Connectionist Models Summer School.* Morgan Kaufmann Publishers.

Lakoff, G. (1988b) Cognitive phonology. Manuscript draft; presented at the LSA annual meeting.

Pinker, S., & Prince, A. (1988) On language and connectionism: analysis of a parallel distributed processing model of language acquisition. In S. Pinker & J. Mehler (eds.), *Connections and Symbols.* MIT Press.

Pollack, J. (1988) Recursive auto-associative memory: devising compositional distributed represesnations. *Proceedings of the Tenth Annual Conference of the Cognitive Science Society,* pp. 33-39

Rumelhart, D. E., & McClelland, J. L. (1986) On learning the past tenses of English verbs. In J. L. McClelland & D. E. Rumelhart (eds.), *Parallel Distributed Processing: Explorations in the Microstructure of Cognition.* MIT Press.

Touretzky, D. S. (1986) BoltzCONS: reconciling connectionism with the recursive nature of stacks and trees. *Proceedings of the Eighth Annual Conference of the Cognitive Science Society,* pp. 522-530.

Touretzky, D. S. (1989) Chunking in a connectionist network. *Proceedings of the Eleventh Annual Conference of the Cognitive Science Society.* Lawrence Erlbaum Associates.