Cognitive Science 30 (2006) 779–801

# Harmony in Linguistic Cognition

Paul Smolensky

*Cognitive Science Department, Johns Hopkins University*

**Abstract**

In this article, I survey the integrated connectionist/symbolic (ICS) cognitive architecture in which higher cognition must be formally characterized on two levels of description. At the microlevel, parallel distributed processing (PDP) characterizes mental processing; this PDP system has special organization in virtue of which it can be characterized at the macrolevel as a kind of symbolic computational system. The symbolic system inherits certain properties from its PDP substrate; the symbolic functions computed constitute optimization of a well-formedness measure called *Harmony.* The most important outgrowth of the ICS research program is optimality theory (Prince & Smolensky, 1993/2004), an optimization-based grammatical theory that provides a formal theory of cross-linguistic typology. Linguistically, Harmony maximization corresponds to minimization of markedness or structural ill-formedness. Cognitive explanation in ICS requires the collaboration of symbolic and connectionist principles. ICS is developed in detail in Smolensky and Legendre (2006a); this article is a précis of and guide to those volumes.

*Keywords:* Connectionism; Symbolic theory; Linguistics; Optimality theory; Cognitive architecture

## 1. Introduction

The advent of parallel distributed processing (PDP)—a microtheory of cognition developed by Rumelhart, McClelland, Hinton, and others around the early 1980s (e.g., Grossberg, 1982; Hinton & Anderson, 1981; Kohonen, 1977; McClelland, Rumelhart, & PDP Research Group, 1986; Rumelhart, McClelland, & PDP Research Group, 1986)—marked a watershed of modern cognitive science. The proper treatment of these subsymbolic ideas in the context of symbolic theories of cognition remains an important open issue.

A crucial test domain for PDP theory is language. Language is a highly symbolic cognitive function. Can a subsymbolic theory such as PDP actually cope with the symbolic character of language (e.g., Christiansen & Chater, 1999; Fodor & Pylyshyn, 1988)? Existing macrotheory

Correspondence should be addressed to Paul Smolensky, Cognitive Science Department, Johns Hopkins University, Baltimore, MD 21218–2685. E-mail: smolensky@jhu.edu

in the domain of language is highly successful. Can a microtheory actually advance the science of language (e.g., Pinker & Prince, 1988; Prince & Smolensky, 1997)?

Central to the challenge of the domain of language is the extraordinary breadth and depth of the empirical phenomena to be explained, which include those listed in (1). (Those of (1a) and (1d) should be quite familiar; I explain (1b) and (1c) in Sections 3 and 4, respectively.)

(1) Explananda of the language domain
  a. *Psycholinguistics:* Comprehension processes, production processes, sensitivity to statistics, ….
  b. *Philosophical foundations:* Unbounded productivity, sensitivity to constituency, ….
  c. *Theoretical linguistics:* Cross-linguistic typology, universals, alternations, ….
  d. *Acquisition:* Statistical learning, innate structure, formal learnability, ….

To address this range of explananda, over the past two decades, PDP has been incorporated into the integrated connectionist/symbolic (or ICS) cognitive architecture, developed in detail in Smolensky and Legendre (2006a). In this article, I provide a précis of and guide to those volumes.

ICS encompasses two levels of description of higher cognition in the human mind/brain. Described at the microlevel, the system performs numerical computation—activation spreading among processing units. Described at the macrolevel, the same system can be characterized as performing symbolic computation. At the microlevel, which I call here the μ level, the activation-spreading network possesses special structure; it is this structure that gives rise to a macrolevel or *M level* that can be seen as symbol processing. In turn, the symbolic M level has special structure arising from its network microstructure.

The link between the network description of the μ level and the symbolic description of the M level is a higher level connectionist description of the M level. At the μ level, one sees individual units and individual connections. Moving up to the M level, one can group together the activation values of large numbers of units into activation patterns and treat the patterns as the basic element of description. Seen through PDP glasses, these elements are patterns of activity; seen through symbolic glasses, these same elements are symbol structures. At the M level, activation patterns and symbol structures are two different formal descriptions of the same mental representations. The symbolic description provides much power for understanding cognition at the M level; but for understanding the microstructure of cognition, the symbolic description must be transformed to the PDP M-level description consisting of activation patterns. These activation patterns can then be reduced to the μ level by decomposing them into the activation values of individual units.

In the ICS architecture, the M level can be described in terms of symbolic functions defined over symbolic representations; crucially, however, these functions are not computed by symbolic algorithms—for automatic language processing, at least; on the time scale of seconds, symbolic algorithms may well operate in domains such as problem solving (e.g., J. R. Anderson & Lebiere, 1998; Goldstone, 2005; Newell, 1990). The processes that compute these cognitive functions can only be understood by transforming this M-level description from symbols to activation patterns and then reducing these patterns to the μ level where the activation values of individual units are multiplied by the weight values of individual connections to determine the activation values of other units.

A crucial bridge between the μ level and M level is the focus of this article: Harmony. Under the PDP description, *Harmony* is a connectionist well-formedness measure; it is maximized by μ-level spreading activation processes. At the M level, in the language domain, Harmony has a symbolic interpretation as an extension of the traditional linguistic notion of markedness (Battistella, 1990; Jakobson, 1962). A linguistic structure is called *marked* when it is relatively complex or in some sense defective; marked representations have lower Harmony and are thus avoided by the language-processing system.

The split-level architecture of ICS is analogous to that found in computer science and in physics. A desktop computer is, at a macrolevel, a graphics processor, a folder manipulator, and a text rendering system; at the microlevel, it is a bit processor. At the macrolevel, a gas is a system characterized by pressure, temperature, and entropy; at the microlevel, it is a system of particles that individually possess none of these properties.

What makes the case of higher cognition in the human mind/brain particularly challenging is that at the macrolevel, the system has a discrete, symbolic description like the macrolevel of a computer; at the microlevel, the same system is continuous and numerical like the microlevel of a gas. In computer science, both macrolevels and microlevels are symbolic; in physics, both are numerical. In ICS, a qualitative shift occurs in reducing the M level to the μ level (Smolensky, 1988, p. 11): A discrete, symbolic computational system becomes a continuous, numerical one. It is not surprising that the formal tools of ICS for linking macrostructure and microstructure marry techniques from computer science with ideas from physics.

Harmony, in particular, joins the concept of (negative) energy from statistical physics (Hinton & Sejnowski, 1986; Smolensky, 1986) with the concept of markedness from linguistic theory; markedness is one conception of the notion of (negative) well-formedness, which is as central to grammar as energy is to physics. Just as physics privileges states of lowest energy, so grammar privileges states of lowest markedness. In the unification provided by ICS, the connectionist network computing states of maximal Harmony is the grammar identifying representations of maximal well-formedness.

Why might ICS be on the right track? I take as highly significant the successes of both activation-based/statistical processing theory (Jurafsky, 1996; Manning & Schütze, 1999; Seidenberg & MacDonald, 1999) and symbolic rule-/constraint-based grammatical theory. The apparent incompatibility of activation- and symbol-based theories has led many to choose one and reject the other. ICS is the result of refusing to give up the insights of either approach.

What's the potential payoff from ICS? First, explaining how, in general, symbolic macrocomputation can arise from PDP microcomputation—how it can be that both PDP and symbolic cognitive theories capture important formal properties of the mind/brain. This can be seen to address a computational formulation of aspects of the mind–body problem. Second, more specifically, ICS holds the potential to explain precisely how activation-based processing theory should be integrated with rule- or constraint-based grammatical theory. If successful, ICS would ultimately lead to explanations, based on reduction to neural computation, of the full range of explananda listed in (1).

Of these potential areas of ICS contribution, to date, some have received considerably more attention than others. Several general formal results showing how symbolic computation can be realized in PDP networks have been obtained; some of these I describe in Section 2. These results provide a new explanation, by reduction to neural computation, of key symbolic properties of higher cognition; I discuss the case of unbounded linguistic productivity (1b) in Section 3. Widespread implications for theoretical linguistics (1c) have been developed; I discuss the problem of grammatical typology—characterizing the possible types of grammars that can be found among human languages—in Section 4. There has also been significant progress in the area of language acquisition (1d), but these are beyond the scope of this article (see Davidson, Jusczyk, & Smolensky, 2006; Kager, Pater, & Zonneveld, 2004; Legendre, this issue; Stemberger & Bernhardt, 1998; Tesar, this issue; Wilson, this issue).

Thus, the focus of ICS research to date has been on the general problem of realizing symbolic computation in PDP networks and on the implications for symbolic theories of language. It remains as a central problem for future ICS research to exploit the potential of ICS to enhance the power of connectionist theories of language processing and learning by endowing PDP networks with the type of microstructure that enables them to realize symbolic grammatical knowledge (for some initial work, see Legendre, Miyata, & Smolensky, 2006; Legendre, Sorace, & Smolensky, 2006; Soderstrom, Mathis, & Smolensky, 2006).

## 2. Realizing symbolic computation in PDP networks

How can connectionist and symbolic architectures be unified into an integrated cognitive architecture? In this section, I give an introduction to the formal characterization of ICS developed in detail in Part II of Smolensky and Legendre (2006a).

I presume a high-level architecture with many cognitive modules, with complex interaction among many levels in these modules. The cognitive system as a whole is an intricate combination of serial and parallel processing, but I focus in on a small part, one parallel chunk of this overall system. Such a chunk consists of a function $f$, taking a representation $I$ as input and producing a representation $O$ as output. At the M level, $I$ and $O$ can be viewed as symbol structures $\mathtt{I}$ and $\mathtt{O}$; perhaps, for example, $\mathtt{I}$ is a parse tree encoding the syntactic structure of a sentence $S$, whereas $\mathtt{O}$ is a tree structure encoding the logical form of $S$—part of the meaning of $S$. At the μ level, $I$ and $O$ are reduced to the activation values of input and output units of a connectionist network $\mathcal{N}$. The link is the M-level PDP description in which the list of activation values over the input and output units respectively constitute the input vector **i** and the output vector **o**.

The function $f$ is computed at the μ level by the spreading activation processes of $\mathcal{N}$. The linguistic knowledge manifest in $f$ is realized at the μ level by all the individual connection weights of $\mathcal{N}$. In the PDP M-level description, the individual weights are treated as a single entity, the weight matrix **W**. The output of $\mathcal{N}$ is the output vector **o** contained within the overall pattern of activity **a**, which has maximum Harmony among all patterns that include the given input vector **i**. In this section, I flesh out this picture to reveal the formal structure of the ICS architecture.

### *2.1. Realizing symbol structures and symbolic functions*

The foundation of the ICS architecture is provided by (2) (Smolensky, 2006b). I now introduce the notions employed in this principle.

(2) Tensor product representations

a. Mental representations are defined by the activation values of connectionist units. When analyzed at the M level, these representations are distributed patterns of activity—activation vectors. For core aspects of higher cognitive domains, these vectors realize symbolic structures.

b. Such a symbolic structure **s** is defined by a collection of structural roles $\{r_i\}$, each of which may be occupied by a filler $\mathtt{f}_i$; s is a set of constituents, each a filler/role binding $\mathtt{f}_i/r_i$.

c. The connectionist realization of s is an activity vector:
$\mathbf{s} = \sum_i \mathbf{f}_i \otimes \mathbf{r}_i$ — basic; generalized: $\mathbf{s} = \mathbf{F}[\mathbb{C}[\sum_i \mathbf{f}_i \otimes \mathbf{r}_i]]$.

d. In higher cognitive domains such as language and reasoning, mental representations are recursive: The fillers or roles of **s** have themselves the same type of internal structure as **s**, and these structured fillers **f** or roles $r$ in turn have the same type of tensor product realization as **s**.

(2a) asserts the general relation between symbolic and connectionist representations in ICS. Mental representations are realized as distributed activation patterns—this is the key PDP principle. What ICS adds is this: In higher cognition—in particular, in the language faculty—these patterns can also be described as symbol structures. How this is possible is defined in the remainder of (2). Formally, the key distinction between PDP and local connectionist representation is that the crucial element in the latter is the unit, whereas in PDP, it is the activation pattern: formally, a vector—a list of the activation values of a set of units. Thus, the mathematics of vectors forms the formal foundation of the ICS architecture.

Symbol structures in ICS are formally characterized by sets of roles and fillers (2b). For example, one simple type of symbol structure is the *string:* a sequence of atomic symbols. A natural choice of roles is $\{r_1 =$ first element of the string, $r_2 =$ second element, …$\}$. The fillers are the atomic symbols. A particular string such as **s** = **AB** is then defined by the pairings or bindings $\{\mathtt{A}/r_1, \mathtt{B}/r_2\}$; each binding, or constituent, specifies which symbol fills a role. (See also Principle 4 following.)

The activation pattern realizing a structure such as **s** = **AB**, defined generally in (2c), is gotten by adding together or superimposing the patterns realizing each of the individual constituent bindings defining **s**. The pattern realizing a binding like $\mathtt{A}/r_1$ is the *tensor product* of a vector **A** that realizes the symbol **A** and a vector $\mathbf{r}_1$ that realizes the role $r_1$. The tensor product of two vectors $\mathbf{f} = (\mathrm{f}_1, \mathrm{f}_2, \ldots, \mathrm{f}_m)$ and $\mathbf{r} = (r_1, r_2, \ldots, r_n)$ is the vector $\mathbf{f} \otimes \mathbf{r} = (\mathrm{f}_1 r_1, \mathrm{f}_1 r_2, \ldots, \mathrm{f}_1 r_n, \mathrm{f}_2 r_1, \mathrm{f}_2 r_2, \ldots, \mathrm{f}_2 r_n, \ldots, \mathrm{f}_m r_1, \mathrm{f}_m r_2, \ldots, \mathrm{f}_m r_n)$ consisting of all products of elements in **f** and **r**. The vector $\mathbf{T} = \mathbf{f} \otimes \mathbf{r}$ contains $mn$ elements $\mathrm{T}_{\beta\gamma} = \mathrm{f}_\beta r_\gamma$; these elements have two subscripts, so **T** is a special type of vector called a *second-rank tensor.* (Such a tensor is often displayed as a two-dimensional numerical array in which case it equals the matrix product $\mathbf{f}\,\mathbf{r}^{\mathbf{T}}$.) For example, if the realization of **A** were the distributed pattern $\mathbf{A} \equiv (.1, .2, .3)$, and the realization of the role $r_1$ were the pattern $\mathbf{r}_1 \equiv (.1, -.1)$, then the pattern realizing the binding $\mathtt{A}/r_1$ would be $(.1, .2, .3) \otimes (.1, -.1) = (.01, -.01, .02, -.02, .03, -.03)$.

Adding together the vectors for multiple bindings does not destroy the information determining which symbols are bound to which roles provided the vectors realizing the roles are not linearly dependent; other principles of ICS impose such independence conditions.

The representations I just described are *basic* tensor product representations. *Generalized* tensor product representations (2c) are built from basic tensor product representations by collapsing the size of the representation using an operator $\mathbb{C}$ and then subjecting the activation vector to a nonlinear function **F**. (The *contraction* operation $\mathbb{C}$ is a generalization to tensors of the operation of reducing an ordinary matrix to its *trace:* The full set of values $\{M_{ij}\}$ is reduced to a single number, the sum of its diagonal elements: $\mathbb{C}[M] = M_{11} + M_{22} + \ldots$. The nonlinear operation **F** simply takes a nonlinear function F—such as the logistic sigmoid activation function $F(x) = 1/[1 + e^{-x}]$—and applies it independently to each value in the vector: $\mathbf{F}[(v_1, v_2, \ldots)] = (F(v_1), F(v_2), \ldots)$. Generalized tensor product representations reduce the size of networks needed to realize symbolic structure and inherit advantages of nonlinear activation functions, but their general properties are considerably more difficult to analyze formally.

The symbol structures essential for higher cognition, including language, are often recursive (2d); unlike a string in which each role is filled by an atomic symbol, in a recursive structure, each role can itself be filled by a composite structure. For example, binary-branching trees can be treated as having two roles, $r_0$ = left-subtree and $r_1$ = right-subtree. In the particular tree **`[A [B C]]`**, $r_0$ is filled by the atomic symbol **`A`**, but $r_1$ is filled by the tree **`[B C]`** (see also Principle 4 following). This complex filler itself has left- and right-subtree roles (filled respectively by the atomic symbols **`B`** and **`C`**). Tensors are appropriate means of realizing structures like trees because tensors can be analyzed as having a recursive structure too. In a recursive tensor product representation of binary trees, the pattern realizing the left-subtree role of the right-subtree role (the left-child of the right-child of the tree root—filled by **`B`** in **`[A [B C]]`**) is the vector $\mathbf{r_0} \otimes \mathbf{r_1}$, recursively defined in terms of the patterns $\mathbf{r_0}$ and $\mathbf{r_1}$ respectively realizing the roles $r_0$ and $r_1$. This role vector $\mathbf{r_0} \otimes \mathbf{r_1}$ is itself a second-rank tensor; when it is bound to an atomic filler such as the pattern **B** realizing the symbol **`B`**, what results is the third-rank tensor $\mathbf{B} \otimes \mathbf{r_0} \otimes \mathbf{r_1}$. (In general, the tensors realizing symbols at depth $d$ in a binary tree have rank $d + 1$.)

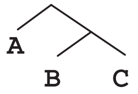
(2) is the most basic principle of the ICS architecture. Tensor product representations as such were developed in Dolan (1989) and Smolensky (1990), but it is a surprising result of ICS research that in fact, the family of tensor product representations includes a wide range of connectionist proposals for the realization of structured data; I give some examples in (3) (the asterisks mark cases of generalized tensor product representations; the other cases are basic tensor product representations).

(3) Examples of special cases of tensor product representations
   a. Recursive auto-associative memory* (Pollack, 1988, 1990).
   b. Holographic reduced representations* (Plate, 1991, 1994, 2003).
   c. Binding by synchronous neural firing (e.g., Shastri & Ajjanagadde, 1993).
   d. Spatial representations modeling parietal cortex: Pouget and Sejnowski (1997).
   e. Representations modeling propositional memory: Halford, Wilson, and Phillips (1998).
   f. Fractal representations of sequences: Tabor (2000).

The case of (3c) was presented in Tesar and Smolensky (1994); for the others, see Smolensky and Tesar (2006).

Tensor product representations provide a structure-preserving mapping—an *isomorphism*—between symbolic and PDP computation. This is spelled out in (4), which shows, for increasingly complex types of structure, the correspondences between the symbolic and connectionist formalizations of these types of structure.

(4) The symbolic-connectionist isomorphism of ICS

| Structuring | Symbolic formalization | | Connectionist formalization | |
|---|---|---|---|---|
| operation | Structures | Example | Example | Vector operation |
| Combining | Sets | {**c**$_1$, **c**$_2$} | $c_1 + c_2$ | Vector sum: + |
| Filler/role binding | Strings, frames | **AB** = {**A**/$r_1$, **B**/$r_2$} | $A \otimes r_1 + B \otimes r_2$ | Tensor product: ⊗ |
| Recursive embedding | Trees | [**A** [**B C**]] =  | $A \otimes r_0 + [B \otimes r_0 + C \otimes r_1] \otimes r_1$ | Recursive role vectors: $r_{\text{left-/right-child(x)}} = r_{0/1} \otimes r_x$ |

To what extent can tensor product representations support symbolic computation? Several formal results have been obtained (Smolensky, 2006f, Section 4.1); (5) gives one of them.

(5) Symbolic computation with tensor product representations

a. Definition: Let $\mathcal{B}$ be the set of binary trees over an alphabet *A*, and let *f* be a function from $\mathcal{B}$ to $\mathcal{B}$. *f* is called *PC* if it can be generated by a finite composition of the primitive operations

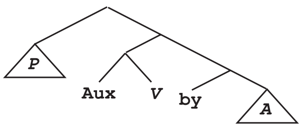
**cons**, **ex**$_0$, **ex**$_1$.

b. Theorem: Any PC function *f* can be computed by a linear associator.

The basic tree operations referred to here are **ex**$_0$, which takes a binary tree and extracts its left subtree; **ex**$_1$, its right counterpart; and **cons**, which takes two trees ***L*** and ***R*** and constructs the binary tree **[*L* *R*]**, which has and ***L*** and ***R*** as its left and right subtrees.

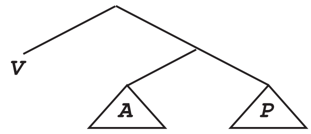
An example of a PC function is the function *g* that takes any binary tree of the form

(6) Input to *g*



(a schematic English passive-voice sentence, with patient ***P***, agent ***A***, and verb ***V***) and maps it to a binary tree encoding the corresponding logical form ***V***(***A***, ***P***):

(7) Output of *g*



According to (5b), there is a linear associator that computes *g:* That is, there is a connectionist network with the property that when the tensor product realization of a tree of the general form of (6) is placed on the input units, the output units attain an activation pattern that is the tensor product realization of the corresponding tree of the form (7). This network has no hidden units and has linear output units; the output pattern $\mathbf{a}_{\mathbf{out}}$ is simply the matrix product $\mathbf{W}_g\,\mathbf{a}_{\mathbf{in}}$ in which $\mathbf{a}_{\mathbf{in}}$ is the input vector, and $\mathbf{W}_g$ is the matrix of weights between input and output layers.

In fact, the mapping from PC functions $f$ to the weight matrix $\mathbf{W}_f$ that realizes them in a linear associator is a compositional isomorphism. This is illustrated in Principle 8 for the particular function $g$ of the previous paragraph (Smolensky, 2006b, p. 196).

(8) An example function $g$

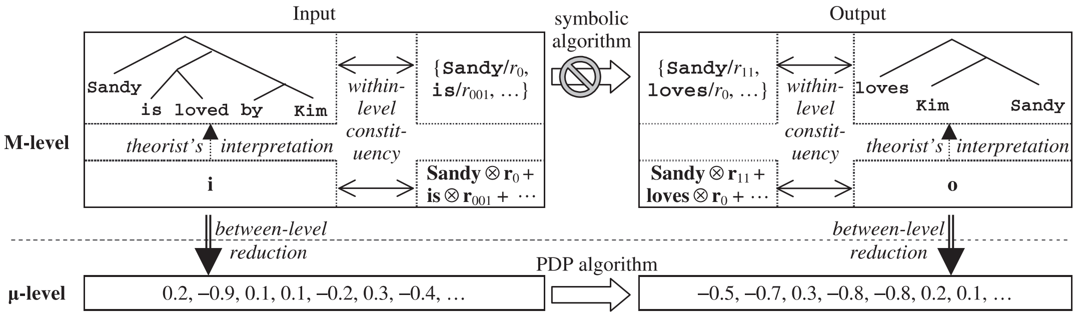
a. $\mathtt{g(s) = cons\ (\ ex_1, (ex_0, (ex_1\ (s))),}$
$\mathtt{cons\ (ex_1, (ex_1, (ex_1\ (s))), (ex_0\ (s)))}.$

b. $\mathbf{W}_g = \mathbf{W}_{\text{cons0}}[\mathbf{W}_{\text{ex1}}\mathbf{W}_{\text{ex0}}\mathbf{W}_{\text{ex1}}] + \mathbf{W}_{\text{cons1}}[\mathbf{W}_{\text{cons1}}(\mathbf{W}_{\text{ex1}}\mathbf{W}_{\text{ex1}}\mathbf{W}_{\text{ex1}}) + \mathbf{W}_{\text{cons1}}(\mathbf{W}_{\text{ex0}})]$.

(8a) shows how $g$ can be constructed by composing the primitive tree-manipulating operations of (5a). There are weight matrices $\mathbf{W}_{\text{cons0}}$, $\mathbf{W}_{\text{cons1}}$, $\mathbf{W}_{\text{ex0}}$, and $\mathbf{W}_{\text{ex1}}$ that realize the primitive symbolic functions in a linear associator; their definition requires too much preparatory work to undertake here. (8b) shows how these primitive matrices can be composed to form the weight matrix $\mathbf{W}g$ that realizes $g$ in a network; (8b) mirrors the structure of its symbolic counterpart (8a).[1]

As illustrated in (9), the symbolic-connectionist relations exhibited in (5) and (8) are within-level relations in the following sense: They relate symbolic descriptions to macrolevel connectionist quantities such as entire patterns of activity (which correspond to symbol structures) and entire weight matrices (which correspond to symbolic functions). These relations link symbolic and connectionist descriptions within the M level. These formal relations hold regardless of the particular patterns that happen to realize individual symbols and individual symbolic roles. Tensor calculus enables activation vectors and weight matrices to be manipulated in a way that corresponds closely to the way symbols and functions are manipulated.

(9) ICS levels


Input
symbolic algorithm
Output
Sandy
is loved by Kim
{Sandy/$r_0$, is/$r_{001}$, …}
within-level constituency
{Sandy/$r_{11}$, loves/$r_0$, …}
within-level constituency
loves
Kim Sandy
M-level
theorist's interpretation
theorist's interpretation
i
Sandy ⊗ $\mathbf{r}_0$ + is ⊗ $\mathbf{r}_{001}$ + ⋯
Sandy ⊗ $\mathbf{r}_{11}$ + loves ⊗ $\mathbf{r}_0$ + ⋯
o
between-level reduction
between-level reduction
PDP algorithm
µ-level
0.2, –0.9, 0.1, 0.1, –0.2, 0.3, –0.4, …
–0.5, –0.7, 0.3, –0.8, –0.8, 0.2, 0.1, …

In addition, however, there is the between-level relation linking the microlevel and macrolevel descriptions of the connectionist network itself. This is the relation between an entire activation pattern **a** and the individual activation values that it comprises and the relation between an entire weight matrix **W** and the individual connection strengths that it comprises. This between-level relation is where the particular patterns realizing symbols come into play. To understand the respective roles of symbolic and connectionist computation in ICS explanations of cognitive phenomena, which I discuss in Section 3, it is important to keep distinctly in mind (a) the within-level relation (M level to M level) relating symbols to entire patterns and (b) the between-level relation (M level to μ level) relating entire patterns to individual unit activations. I postpone until Section 3 the discussion of the algorithms referred to in (9) that actually compute outputs from inputs.

### *2.2. Harmony*

In ICS, symbolic computation arises as the macrolevel behavior of underlying neural networks. It is therefore crucial to characterize the macrolevel properties of various types of networks. For grammar, the type of network defined in Principle 10 and exemplified in (11) proves to be central (Smolensky, 2006e, p. 381).

(10) Definition: A deterministic network $\mathcal{N}$ is harmonic if it has all the following properties:

a. Updating: The units in $\mathcal{N}$ are either
   - ♦ Discrete valued and updated asynchronously in discrete time or
   - ♦ Continuous valued and updated synchronously in discrete or continuous time.

b. Activation function: The units in $\mathcal{N}$ are quasi-linear units with a monotonically increasing activation function $f$: Each unit's equilibrium activation level $a_\alpha^{eq}$ is given by $a_\alpha^{eq} = f(\Sigma_\beta W_{\alpha\beta} a_\beta)$.

c. Connectivity: The connectivity of $\mathcal{N}$ is either
   - ♦ Feed forward or
   - ♦ Symmetric.

(11) Examples: These networks are harmonic:

a. Hopfield (1982, 1987) networks.
b. Perceptrons (Minsky & Papert, 1969).
c. Linear associators (Kohonen, 1977).
d. Prototypical "backprop nets" (Rumelhart, Hinton, & Williams, 1986).
e. Brain-state-in-a-box models (J. A. Anderson, Silverstein, Ritz, & Jones, 1977).
f. Additive nets (Cohen & Grossberg, 1983).
g. The cascade model (McClelland, 1979).

The macrolevel behavior of Harmonic networks[2] is characterized in (12)[3] (Smolensky, 2006e, pp. 381–382); this generalizes results including those of Cohen and Grossberg (1983), Golden (1986, 1988), and Hopfield (1982, 1984).

(12) Theorem: Given: $\mathcal{N}$, a harmonic network with activation function $f$ and Harmony function[4]

$$H(\mathbf{a}) \equiv H_0(\mathbf{a}) + H_1(\mathbf{a}) \equiv \Sigma_{\alpha\beta}\, a_\alpha\, W_{\alpha\beta}\, a_\beta + \Sigma_\alpha\, h^f(a_\alpha).$$

Then $\mathcal{N}$ maximizes $H$ in a sense appropriate to its architecture (10c):

- ♦ $\mathcal{N}$ feed forward: The network state is a sequential $H$ maximum.
- ♦ $\mathcal{N}$ recurrent: $H$ is monotonically weakly increasing in time.

A state of a feed-forward network is a *sequential H maximum* if the activation pattern at each level is the one with maximum Harmony given the patterns at preceding levels. In recurrent harmonic networks, Harmony is monotonically weakly increasing in time: Its value increases or stays the same.

The importance of Harmony maximization (12) for the ICS theory of language emerges when it is combined with (2), which asserts that for language, mental representations have the special structure of tensor product representations; the result is (13) (Legendre, Miyata, & Smolensky, 1990; Smolensky, 2006c).

(13) Theorem: Harmonic Grammar soft constraints
Let $\mathbf{a}$ be a tensor product representation realizing a symbolic structure $\mathsf{s}$ with constituents $\{\mathsf{c}_k\}$.

a. The Harmony of this representation is

$$H(\mathbf{a}) = \textstyle\sum_{j \leq k} H(\mathsf{c}_j, \mathsf{c}_k) \equiv H(\mathsf{s}),$$

where $H(\mathsf{c}_j, \mathsf{c}_k)$—the Harmony resulting from the co-occurrence of $\mathsf{c}_j$ and $\mathsf{c}_k$ in the same structure—is a constant for all $\mathsf{s}$.

b. Equivalently, the Harmony of $\mathsf{s}$, $H(\mathsf{s})$, can be computed using the following rules:

$R_{jk}$: If $\mathsf{s}$ simultaneously contains the constituents $\mathsf{c}_j$, and $\mathsf{c}_k$, then add the numerical quantity $H(\mathsf{c}_j, \mathsf{c}_k)$ to $H$.

Each $R_{jk}$ is called a *soft rule,* and the collection of soft rules defines a harmonic grammar. To determine the Harmony of a structure $\mathsf{s}$, we simply find all rules $R_{jk}$ that apply to $\mathsf{s}$ and add up all the corresponding Harmony contributions $H(\mathsf{c}_j, \mathsf{c}_k)$.

c. Soft rules can equivalently be recast as soft constraints. If $H(\mathsf{c}_j, \mathsf{c}_k)$ is a negative value $-w_{jk}$, then $R_{jk}$ is interpreted as the following (negative) constraint in which $w_{jk}$ serves as a weight or strength:

$C_{jk}$: $\mathsf{s}$ does not simultaneously contain the constituents $\mathsf{c}_j$ and $\mathsf{c}_k$ (strength: $w_{jk}$).

If $H(\mathsf{c}_j, \mathsf{c}_k)$ is a positive value $+w_{jk}$, then $R_{jk}$ corresponds to the following (positive) constraint:

$C_{jk}$: $\mathsf{s}$ does simultaneously contain the constituents $\mathsf{c}_j$ and $\mathsf{c}_k$ (strength: $w_{jk}$).

$H(\mathsf{s})$ is computed by adding together the strengths of all the positive constraints that $\mathsf{s}$ satisfies and subtracting the strengths of all the negative constraints that it violates.

d. With respect to the connectionist description, each Harmony contribution $H(\mathsf{c}_j, \mathsf{c}_k)$ can be calculated at the μ level in terms of the weight matrix $\mathbf{W}$ according to the formula

$$H(\mathsf{c}_j, \mathsf{c}_k) = H(\mathbf{c}_j, \mathbf{c}_k) = \textstyle\sum_{\beta\alpha} [\mathbf{c}_j]_\beta \, (\mathbf{W}_{\beta\alpha} + \mathbf{W}_{\alpha\beta}) \, [\mathbf{c}_\mathrm{k}]_\alpha = \mathbf{c}_j^\mathrm{T} \times (\mathbf{W} + \mathbf{W}^\mathrm{T}) \times \mathbf{c}_k,$$

where $[\mathbf{c}_k]_\beta$ is the activation value of unit β in the vector $\mathbf{c}_k$ that realizes $\mathsf{c}_k$.

e. $H(\mathbf{c}_j, \mathbf{c}_k)$ can be interpreted as the interaction Harmony of the pair $(\mathbf{c}_j, \mathbf{c}_k)$: the amount of Harmony contributed by $\mathbf{c}_j$ and $\mathbf{c}_k$ when they are present together, beyond the sum of the Harmonies of $\mathbf{c}_j$ and $\mathbf{c}_k$ in isolation.

A harmonic grammar is a very simple computational system, as might be expected given that it is straightforwardly realized in a simple PDP network. Despite this simplicity, however, it turns out that harmonic grammar has a great deal of computational power: Any formal language, no matter its position in the Chomsky hierarchy, can be specified by a harmonic grammar (Hale & Smolensky, 2006; Smolensky, 1993). In particular, any Turing machine can be specified by a harmonic grammar because Turing machines are computationally equivalent to unrestricted rewrite-rule grammars, the top level of the Chomsky hierarchy (Hopcroft & Ullman, 1979).

The formal grammars relevant for natural language are of lower complexity (Goldstone, 2004). An example result concerning the level of context-free grammars is given in (14).

(14) Theorem: Given: $\mathscr{G}$, a harmonic grammar for a context-free language $\mathscr{L}$ over an alphabet $A$ and a recursive tensor product realization of the binary trees over $A$ (2d). Then there is a corresponding weight matrix $\mathbf{W}_{\mathscr{G}}$ for a recurrent network $\mathscr{N}$ such that the Harmony assigned by $\mathscr{G}$ to a tree $\mathbf{t}$ is the same as the Harmony of the state $\mathbf{a}_{\mathbf{t}}$ realizing t in $\mathscr{N}$.

As an extremely simple example, the grammar $\mathscr{G}_0 \equiv \{\mathtt{X} \rightarrow \mathtt{A\ B}, \mathtt{Y} \rightarrow \mathtt{B\ A}\}$ has a harmonic grammar with soft rules such as the following:

(15) $R_{\mathtt{X/A}}$: If $\mathbf{s}$ contains a constituent labeled $\mathtt{X}$, and its left subconstituent is labeled $\mathtt{A}$, then add +2 to $H$.

These soft rules can be realized in a simple harmonic network that functions as a parser for $\mathscr{G}_0$ (Smolensky & Legendre, 2006b, pp. 76–77). If the distributed representation of $\mathtt{X}$ is clamped on the units for the mother node, then the network performs top-down completion, filling in the distributed tensor product representation of the sequence $\mathtt{A\ B}$ on the units for the daughter nodes; if $\mathtt{Y}$ is clamped instead, the network completes the tree with $\mathtt{B\ A}$. If the distributed tensor product representation for $\mathtt{B\ A}$ is clamped on the daughter-node units, the network performs bottom-up completion, recognizing the string as a phrase of category $\mathtt{Y}$ by filling in the distributed representation of $\mathtt{Y}$ on the mother-node units.

An example harmonic grammar soft rule for phonology is given in (16); σ denotes "syllable," and $\mathtt{C}$ denotes "consonant."

(16) $R_{\sigma/\mathtt{C}}$: If $\mathbf{s}$ contains a constituent labeled σ and its left subconstituent is labeled $\mathtt{C}$, then add $w$ to $H$.

According to markedness theory, syllables that begin with a consonant are unmarked (relative to syllables beginning with a vowel; Jakobson, 1962; Prince & Smolensky, 2006). Therefore, for real grammars, the $w$ in (16) is positive: A structure obeying this rule has positive Harmony. Markedness theory also states that syllables ending with a consonant are marked; thus, in actual grammars, the soft rule analogous to (16) that refers to a constituent σ with $\mathtt{C}$ in its final position has a negative weight.

## 3. Explaining the productivity of higher cognition

ICS techniques are close to enabling full-scale parsing of formal languages—via harmonic grammar—through PDP Harmony maximization. Other ICS techniques already enable recursive functions over trees to be computed by feed-forward PDP networks (5). For example, *PassiveNet,* the linear, two-layer, feed-forward network with the weight matrix given in (8b) provably computes the recursive function *g* of (6) through (8a). Although it is of little interest for natural-language theory, PassiveNet is nonetheless a finitely specified PDP network that exhibits three properties of higher cognition properly identified by Fodor and Pylyshyn (1988) as particularly challenging for connectionist computation. First, this network displays *structure sensitivity:* Its output is exactly as sensitive to the structure of its input as is the symbolic recursive function *g* that it realizes. Second, the representations of this network display *systematicity*—their structure is isomorphic to the symbol system they realize (e.g., the capability to encode representations in which **A** falls in one position entails the capability to encode representations in which **A** falls in any other representable position.) Third, and most important, PassiveNet displays unbounded productivity; it correctly processes inputs of the specified form (6) no matter how complex the constituents of that input—it is a formal realization of the recursive function *g*.

How can these properties—structure sensitivity, systematicity, and unbounded productivity—be explained? Do they hold because they have been specially "wired in" to PassiveNet, or do they follow necessarily from the principles of ICS (cf. Fodor & McLaughlin, 1990)? Do they hold because the network simply implements a "classical" symbolic architecture (Fodor & Pylyshyn, 1988)?

The properties of PassiveNet, including its correctness as a means of computing the function *g* (8a) are formally provable. There can be no mystery about why they hold. They hold because the patterns of activation in the input and output layers have a particular formal structure: They have constituent structure in the precise sense defined by (2c), and the pattern of connections in the network has constituent structure in the precise sense defined by (8b) (cf. Fodor, 1997). The structure of the activation vectors and the structure of the weight matrix are mutually compatible in that they are defined relative to the same set of recursive, distributed role vectors for binary tree constituency.

That PassiveNet has the properties of structure sensitivity, systematicity, and unbounded productivity is not a result of something specially wired in to this network. The formal demonstrations of its properties I described in the previous paragraph derive from the general principles of ICS: (2) is the principle governing the structure of representations (activation patterns), and there is a corresponding principle governing the structure of processes (weight matrices); this principle essentially asserts that weight matrices have the general structure described in Note 1 (Smolensky & Legendre, 2006b, p. 71). These principles define an entire class of systems, all of which have the same crucial general properties as PassiveNet because those properties are a necessary consequence of those principles (Smolensky, 1995).

The systems defined by the general principles of ICS do not implement a symbolic architecture. If they did, there would be a complete formal account of the processing in these systems in terms of algorithms performing symbol manipulation over constituents. However, there is no such algorithm for PassiveNet nor for the general class of which it is but one example. There is of course a symbolic algorithm for computing the same function *g* that PassiveNet computes—it can be read off the symbolic expression for *g* in (8a)—but that algorithm is not a description of how PassiveNet computes. There is no symbolic algorithm operating in PassiveNet, but the fact that it computes *g* is no mystery: It follows by tensor algebra from the structure of the activation patterns and weights in the network, as I already pointed out. There is an algorithm that describes precisely and completely the computation carried out by PassiveNet, but it is a PDP algorithm specifying how weighted sums of input activations determine output activations: It is a parallel numerical algorithm that is not an implementation of a serial symbolic algorithm.

Essential to the role of the algorithms hypothesized by cognitive theories is that they must specify the computational resources required by particular input–output mappings: They must predict which inputs will take more resources and therefore lead to more errors or greater reaction time. Algorithms must specify internal computational states between input and output, predicting that one input will prime another by increasing the accessibility of intermediate states common to the two. According to ICS, there are no symbol-manipulation algorithms that do this work for the kinds of rapid, automatic, unconscious mental processes like those at work in language processing. It is PDP algorithms that do this work.

Thus, the explanation of the unbounded productivity of cognition provided by ICS is a new explanation: It is not an implementation of the classical symbolic explanation. Symbolic constituents play an explanatory role in ICS, but not the role they play in symbolic theory: Mental processes are not algorithms manipulating those constituents.

This is where the distinction I drew at the end of Section 2.1 between within-level and between-level relations becomes crucial: I now return to (9) (derived from Cummins & Schwarz, 1991). A mental representation in ICS can be decomposed in two different ways. There is a within-level constituency relation between the activation pattern for the sentence with symbolic description **`[Sandy [[is loved] [by [Kim]]]]`** and the activation pattern with symbolic description **`Sandy`**. This within-level relation is crucial for the semantic interpretation of this mental representation in ICS as well as in classical symbolic theory. However, this constituency relation plays no role in the algorithmic description of how that mental representation is processed internally; for this, what is crucial is the between-level reduction of the global pattern of activation that is the mental representation of the sentence to the individual activation values that make up the pattern. The internal computation is numerical manipulation of these activation values, not symbol manipulation of constituents. For detailed development of this argument, see Smolensky (2006a).[5]

In the preceding discussion, I have addressed the kind of productivity exhibited by feed-forward ICS networks computing recursive functions. The same type of analysis applies to recurrent ICS networks that compute representations with maximum Harmony, including those that embody harmonic grammars as I discussed previously (13). These networks display similar general properties, including unbounded productivity, and the explanation ICS thus provides for unbounded grammatical competence has the same character, depicted in (9).

## 4. Formal theory of grammatical typology

In the preceding sections, I summarized some of the progress achieved to date by pursuing the promise of ICS theory to reduce to neural computational principles the kind of general symbolic properties characteristic of higher cognition in general and of language in particular—the explananda of (1b). In this section, I turn to the explananda of (1c) and briefly summarize some surprising contributions to the theory of universal grammar.

Recall that at the M level, the ICS concept of Harmony, when applied to language, absorbs the traditional linguistic notion of markedness. Can the formal ICS characterization of grammar as a Harmony-maximizing system make contributions to theoretical linguistics that go beyond those of previous theories such as markedness theory?

According to the ICS principles, our knowledge of grammar is embodied in harmonic PDP networks; the representations created by these networks are those that maximize Harmony. The Harmony of these representations is governed by a harmonic grammar, which can be viewed as a set of soft constraints that assess positive or negative Harmony to preferred or dispreferred combinations of constituents. An example of such a constraint was given in (16); one version of this constraint assigns negative Harmony to any syllable ($\sigma$) that lacks an initial consonant (an *onset*): I call this constraint $C_{\text{Onset}}$. This is the formal expression in harmonic grammar of the markedness principle that says syllables lacking an onset are marked—dispreferred. In any particular harmonic grammar, $C_{\text{Onset}}$ has a numerical strength *w;* this is the quantity by which the Harmony of a representation is diminished by each onsetless syllable.

The principles of markedness theory are universal: Onsetless syllables are marked in all languages. In some languages, such as the Australian language Lardil, every syllable has an onset[6]: Intuitively, these are languages in which the strength of $C_{\text{Onset}}$ is high. In languages such as English, not all syllables have onsets: Intuitively, the strength of $C_{\text{Onset}}$ is lower in English, but $C_{\text{Onset}}$ still makes itself felt. One says *an apple* rather than *a apple* to provide the initial syllable of *apple* with an onset; one pronounces *below* as *be.low*, with *l* serving as the onset of the second syllable rather than *bel.ow* in which *l* is a coda of the first syllable.

The problem of linguistic typology is to identify the principles that delimit the set of possible human grammars. It is a fundamental empirical fact that human grammars share many properties; there are many conceivable linguistic systems that are never found among the world's languages—they violate the principles that all actual languages respect.[7] The universality of markedness principles suggests a possible harmonic grammar theory of typology: The grammar of every language consists of the same set of constraints (including $C_{\text{Onset}}$), but the strength of each constraint can vary from language to language.

This theory of typology was formulated by Prince and Smolensky in 1990, but they soon discovered that it predicted as possible types of languages that do not exist. Constraint interaction in human grammars is more restricted than that permitted by arbitrary numerical constraint strengths. The more restricted theory of constraint interaction that yields empirically valid typologies is *optimality theory* (OT; Prince & Smolensky, 1991, 1993/2004). Several crucial principles of OT are given in Principle 17.

(17) Principles of OT
   a. Constraints are universal.[8]
   b. Constraints are violable (forms are optimal).

c. Constraints apply in parallel.
d. Constraint strengths are encoded in strict domination hierarchies.

In a *strict domination hierarchy,* constraints are ranked from strongest to weakest, each constraint having absolute priority over all weaker constraints combined. This can be simulated approximately by numerical constraint strengths that grow exponentially up the hierarchy; it is this special type of interaction, not that arising from the arbitrary numerical constraint strengths of harmonic grammar, that characterizes human grammatical computation. According to OT, the typological space of possible human languages is exactly the space of all possible hierarchies built from a fixed set of universal constraints. Among these are constraints like ONSET, the OT counterpart of the harmonic grammar constraint $C_{\mathrm{Onset}}$; these correspond to principles of markedness theory and form the OT family of constraints called MARKEDNESS.

OT provides a precise formal answer to the fundamental problem of linguistic typology: Human grammars share a common set of constraints, which interact via strict domination. Human grammars differ only in how the universal constraints are ranked, that is, only in which constraint has priority when two universal constraints come into conflict. This is the first general, formal theory of linguistic typology and the first theory in which formal grammars are built directly from markedness principles.

At a somewhat less general level, OT makes a number of contributions to linguistic theory. I sketch a few here; many others were discussed in Smolensky, Legendre, and Tesar (2006, pp. 504–510).

In OT, grammars are built of principles (constraints) defining what structural properties are preferred in linguistic representations; rule-based grammars are built of principles (symbolic rewrite rules) defining operations that construct linguistic representations. In many cases, constraints are universal, whereas rules are not. This is what enables a theory of typology based on constraints when a typological theory based on rules has not proved possible. Stated concisely, it is often true of grammars that universals are manifest in products, not processes (McCarthy, 2002, pp. 25–26).

OT grammars entail implicational universals; although these were a major empirical motivation of markedness theory, it was typically not previously possible to formally derive them from grammatical principles. An example of an implication universal is “if a language allows complex codas (syllable-final sequences of multiple consonants, as in *apt*) then the language also allows simple codas (consisting of a single consonant, as in *at*)” (Greenberg, 1978). Such a universal motivates a markedness principle asserting that complex codas are marked relative to simple codas. The formal counterpart in OT of implicational universals is the *harmonic completeness* of inventories: Because OT grammars are Harmony-maximizing systems, if the inventory of possible elements of type $T$ in language $L$ includes $x$, then it follows formally that $L$’s inventory must also include elements $y$ of type $T$ that have higher Harmony than $x$ (Smolensky, 2006d, p. 149, inter alia).

OT grammars, unlike markedness theory, provide a formal account of the basic empirical phenomena of linguistic theory: the contexts in which various elements are found and the changes—*alternations*—elements undergo when their contexts change. A well-known example is the English past-tense inflection, which alternates between *–t*, *–d*, and *–əd*. (To minimize distractions, I do not treat the case of *–əd* here; it can be analyzed in the same fashion. I simplify the following discussion for expository purposes.)

In OT, alternations are governed by the relative ranking of constraints of two types. A MARKEDNESS constraint $M_{voice}$ requires that adjacent consonants agree in *voicing:* $M_{voice}$ is violated by [pæsd], a potential pronunciation of *passed;* [pæst] satisfies $M_{voice}$ because both [s] and [t] are voiceless. Similarly [bart], a potential pronunciation of *barred,* violates $M_{voice}$, unlike the actual pronunciation [bard]. Of course [bart] is a perfectly good English word—it just is not the past tense of *bar.* Why is [bart] the correct pronunciation of *Bart* when it violates $M_{voice}$? For that matter, why do marked elements such as the cluster [rt] ever appear in languages?

OT's answer is that grammars contain constraints other than MARKEDNESS: They also contain *faithfulness* constraints: These require that the pronounced form of a word be faithful to the context-independent form of that word stored in the lexicon, its *underlying form.* One faithfulness constraint $F_{voice}$ requires that the voicing of a consonant in the pronounced form of a word be the same as the voicing of that consonant in the underlying form. To illustrate simply the general picture provided by OT, suppose that the underlying form of the past-tense suffix is /-t/; the pronounced form of the suffix is faithful to this underlying form in verbs such as *passed, miffed, slapped, kicked,* and so on. In the context of a verb stem with a final voiced consonant like /bar/, a conflict arises that the grammar must resolve: For the past-tense underlying form /bar-t/ (*barred*), the two possible pronunciations each violate a constraint: [bard] violates $F_{voice}$, whereas [bart] violates $M_{voice}$. In English, it must be that the latter constraint is stronger because the latter pronunciation is ruled out; that is, $M_{voice}$ dominates $F_{voice}$ in the English constraint hierarchy: $M_{voice} \gg F_{voice}$. In general, when a markedness constraint dominates a conflicting faithfulness constraint, alternations occur: In contexts in which the markedness constraint demands something other than the underlying form, the pronunciation will differ from the underlying form.

It is when faithfulness dominates markedness that pronunciations will violate markedness. One sees that when the final /t/ is part of the word root, as in the name *Bart* /bart/, the pronunciation is faithful to the underlying /t/ even though a violation of $M_{voice}$ results: [bart] is the correct pronunciation. So for *Bart,* underlying /bart/ leads to pronunciation [bart], whereas for *barred,* underlying /bar-t/ leads to [bard]. This can be naturally analyzed in OT by positing a constraint $F_{voice}$(root), which is like $F_{voice}$ except that $F_{voice}$(root) applies specifically to consonants in roots, whereas $F_{voice}$ applies to all consonants (McCarthy & Prince, 1995). The final root consonant in *Bart* is pronounced faithfully because $F_{voice}(\text{root}) \gg M_{voice}$; the marked cluster [rt] is tolerated in the pronunciation because it would be a worse violation to be unfaithful to the voicing of a root consonant.

The overall English ranking of these constraints is thus $F_{voice}(\text{root}) \gg M_{voice} \gg F_{voice}$; the increased force of faithfulness to root consonants is strong enough to yield pronunciations that are marked (*Bart*). However, outside of roots, as in the past-tense suffix, faithfulness is weaker than markedness, so alternations occur to avoid marked sequences (*barred*). The OT tableaux in (18) explicitly show the competition among alternative pronunciations—candidate outputs of the grammar—for the two underlying forms—inputs to the grammar. The constraint hierarchy is displayed across the top row with more dominant constraints to the left. A "☞" identifies the optimal candidate (the actual output of the grammar for the given input). An asterisk ("*") marks the violation of a constraint by a candidate; "!" marks the fatal violation that renders a candidate *C* suboptimal: the violation of the highest ranked constraint according to which *C* is dispreferred to the optimal candidate.

(18) OT tableaux

a. *Bart*

| input: /bart/ | $F_{voice}$(root) | $M_{voice}$ | $F_{voice}$ |
|---|---|---|---|
| ☞ [bart] | | * | |
| [bard] | *! | | * |

b. *barred*

| input: /bar-t/ | $F_{voice}$(root) | $M_{voice}$ | $F_{voice}$ |
|---|---|---|---|
| [bart] | | *! | |
| ☞ [bard] | | | * |

The concept of faithfulness is a major innovation of OT; it is what enables a formal grammatical framework to be constructed from markedness principles. Markedness alone is insufficient; it is merely one half of the markedness/faithfulness battle that is grammar.

Prince and Smolensky (1993/2004) formulated OT as a general theory of grammatical structure but applied it to problems in phonology. Other seminal contributions to the founding of OT in phonology include McCarthy and Prince (1993a, 1993b, 1995) and closely related to OT, Burzio (1994). OT has been applied to the remaining components of grammar as well (Archangeli, 1997; Kager, 1999; McCarthy, 2002): syntax (Bresnan, 2000; Grimshaw, 1997; Legendre, Grimshaw, & Vikner, 2001; Legendre, Raymond, & Smolensky, 1993, 2006) and semantics/pragmatics (Blutner & Zeevat, 2003; Hendriks, de Hoop, & de Swart, 2000). For further sources on OT, see the on-line Rutgers Optimality Archive.

Ongoing work attempts to develop OT as a theory of linguistic performance as well as a theory of linguistic competence. For example, incremental OT optimization (one word at a time) with constraints from OT syntax competence theory can explain many empirical findings concerning human sentence processing (Stevenson & Smolensky, 2006). Further examples are found in other articles in this issue as I discuss below.

## 5. Conclusion

In this article, I have summarized principles and results of the ICS cognitive architecture that address the reduction of symbolic computation to PDP computation, the explanation of the unbounded productivity of higher cognition, and the contributions to linguistic theory from OT, a grammatical theory that inherits its Harmony-maximizing character from underlying PDP computational principles. Leading work at the frontiers of the ICS research program is discussed in other articles in this issue. Gafos and Benus (this issue) consider how continuous representations and nonlinear dynamics analogous to those of the PDP microlevel can be incorporated into grammatical theory. Davidson (this issue) examines representations that are intermediate between continuous dynamical systems and the discrete segment-level representations traditional in phonology. Wilson (this issue) adopts discrete segmental representations but develops a grammatical theory that is probabilistic and related to both OT and the original probabilistic harmony theory (Smolensky, 1986). This theory develops a novel approach to learning in which universal substantive principles of phonology are formally encoded as learning biases. Learning is also the focus of the other two articles (see also Misker & Anderson, 2003). Legendre (this issue) develops a theory of child language in which probability enters at a higher level: the likelihood that a range of constraint rankings will be used for child production. This work also illustrates an application of OT outside of phonology (to morphosyntax). Finally, Tesar (this issue) shows how OT can enable progress on the difficult problem facing children of simultaneously learning the phonological grammar and the underlying lexicon of their native language.[9]

## Notes

1. An important issue I cannot take up properly here concerns the unbounded nature of recursive functions. To accommodate unboundedly large inputs, the network that realizes a recursive function $f$ must itself have an unbounded number of input and output units. Crucial to the full theorem of which (5) is a part is a finite specification for this unbounded network. The theorem states that for each particular PC function $f$, there corresponds a finite weight matrix $\underline{\mathbf{W}}_f$ that is given by a compositional isomorphism identical to that illustrated in (8). To get the weight matrix for a network with $n$ input units, it suffices to take the tensor product $\mathbf{I}_n \otimes \underline{\mathbf{W}}_f$ in which $\mathbf{I}_n$ is an identity matrix of a size determined by $n$ ($\mathbf{I}_n$ has 1s on the main diagonal and 0s everywhere else). This tensor product operation effectively copies $\underline{\mathbf{W}}_f$ as many times as needed to fill out the size-$n$ network. Alternatively, in place of $\mathbf{I}_n$, one can use the infinite identity matrix providing a finite specification for a single infinite network capable of processing unboundedly large inputs.
2. Networks with asymmetric recurrent connections are not harmonic; their macrolevel behavior is different. For example, they can exhibit periodic behavior, cycling through a set of states repeatedly (e.g., Jordan, 1986).
3. A result corresponding to the theorem in (12) also holds when the notion of harmonic network is extended to include stochastic networks such as the Boltzmann machine (Hinton & Sejnowski, 1983, 1986) and harmony theory (Smolensky, 1983, 1986).
4. The first term, the *core Harmony* $H_0$, is the key term, as it evaluates all interactions in the network using the weight matrix $\mathbf{W}$ that encodes the knowledge in the network; α and β range over the units in the net. It is core Harmony that is characterized in (13). The second term, the *unit Harmony* $H_1$, is simply the sum of terms dependent on each unit's activation value separately; it depends on the nonlinear activation function$f$: $h^f(a) \propto -\int f^{-1}$.
5. The M and μ levels of this article correspond respectively to the highest and lowest sublevels of what is called the c(omputational) level in Smolensky (2006a).
6. For an optimality theoretic analysis of Lardil syllable structure phenomena, see Prince and Smolensky (1993/2004), chap. 7.
7. How can we determine that some conceivable linguistic system is not a "possible human language"? Some recent work (e.g., Wilson, 2003) has attempted to address this

experimentally under the hypothesis that linguistic systems that are not possible will be distinguished by participants' relative difficulty in learning them. In contrast, traditional linguistic typology identifies properties observed in all languages and postulates that the languages violating these properties are not possible. Of the many hundreds of studied languages, every one includes syllables that have onsets; thus, it is postulated that a syllable system that forbids onsets is not a possible human language—whereas systems that forbid codas are often observed. A theory predicting a typology that includes languages in which syllable onsets are forbidden is an empirically unacceptable theory as is a theory predicting a typology not including languages in which codas are forbidden.

8. To assert that a constraint is universal is to assert that it appears in the grammars of all languages—not that it is innate. OT is a theory of universal grammar in this sense: It is a theory of the universal computational properties of grammars, not a theory of innate knowledge. The source of knowledge of universal constraints is an open empirical question.
9. For very helpful suggestions for improving this article, I thank Lisa Davidson, Gary Dell, and an anonymous reviewer. Any remaining errors and points of imprecision, incompleteness, and unclarity are of course my own responsibility. For support (enumerated in Smolensky & Legendre, 2006a) that has helped make possible the two decades of research summarized here, I acknowledge the intellectual and financial support of NSF, the University of Colorado at Boulder, and Johns Hopkins University. Most important, I express my appreciation to all my collaborators in this work; many of these are identified in the references, but special thanks go to Alan Prince and Géraldine Legendre for all they have taught me, on many topics at many levels, and for the great pleasure I have derived from working with them.

## Acknowledgments

I thank the Robert Glushko and Pamela Samuelson for their generosity in founding the prize recognizing David Rumelhart's seminal role in the creation of the modern field of cognitive science. Dave's vision of a cognitive science with broad and deep foundations, both empirically and theoretically, has had a profound impact on me as on so many others, and his solid faith in the potential of formal research on cognition has long been a major source of encouragement. I also acknowledge advice from Natalie Smolensky that has driven the work described here: "When you get to the fork in the road, take it" (Yogi Berra). This attitude is very much in the spirit of Dave Rumelhart, who inspired those of us who worked with him to look past the negative and to seek out the positive in others' work.

## References

Anderson, J. A., Silverstein, J. W., Ritz, S. A., & Jones, R. S. (1977). Distinctive features, categorical perception, and probability learning: Some applications of a neural model. *Psychological Review, 84,* 413–451.

Anderson, J. R., & Lebiere, C. J. (1998). *The atomic components of thought.* Mahwah, NJ: Lawrence Erlbaum Associates, Inc.

Archangeli, D. (1997). Optimality theory: An introduction to linguistics in the 1990s. In D. Archangeli & D. T. Langendoen (Eds.), *Optimality theory: An overview* (pp. 1–32). Malden, MA: Blackwell.

Battistella, E. L. (1990). *Markedness: The evaluative superstructure of language.* Albany: State University of New York Press.

Blutner, R., & Zeevat, H. (Eds.). (2003). *Pragmatics in optimality theory.* London: Palgrave Macmillan.

Bresnan, J. (2000). Optimal syntax. In J. Dekkers, F. van der Leeuw, & J. van de Weijer (Eds.), *Optimality theory: Phonology, syntax and acquisition* (pp. 334–385). Oxford, England: Oxford University Press.

Burzio, L. (1994). *Principles of English stress.* Cambridge, England: Cambridge University Press.

Christiansen, M. H., & Chater, N. (Eds.). (1999). Special issue on connectionist models of human language processing [Special issue]. *Cognitive Science, 23.*

Cohen, M. A., & Grossberg, S. (1983). Absolute stability of global pattern formation and parallel memory storage by competitive neural networks. *IEEE Transactions on Systems, Man, and Cybernetics, 13,* 815–825.

Cummins, R., & Schwarz, G. (1991). Connectionism, computation, and cognition. In T. E. Horgan & J. Tienson (Eds.), *Connectionism and the philosophy of mind* (pp. 60–73). Dordrecht, The Netherlands: Kluwer.

Davidson, L., Jusczyk, P. W., & Smolensky, P. (2006). Optimality in language acquisition: I. The initial and final states of the phonological grammar. In P. Smolensky & G. Legendre (Eds.), *The harmonic mind: From neural computation to optimality-theoretic grammar* (Vol. 2, pp. 793–839). Cambridge, MA: MIT Press.

Dolan, C. P. (1989). *Tensor manipulation networks: Connectionist and symbolic approaches to comprehension, learning, and planning.* Unpublished doctoral dissertation, University of California, Los Angeles.

Fodor, J. A. (1997). Connectionism and the problem of systematicity (continued): Why Smolensky's solution still doesn't work. *Cognition, 62,* 109–119.

Fodor, J. A., & McLaughlin, B. P. (1990). Connectionism and the problem of systematicity: Why Smolensky's solution doesn't work. *Cognition, 35,* 183–204.

Fodor, J. A., & Pylyshyn, Z. W. (1988). Connectionism and cognitive architecture: A critical analysis. *Cognition, 28,* 3–71.

Golden, R. M. (1986). The "brain-state-in-a-box" neural model is a gradient descent algorithm. *Mathematical Psychology, 30–31,* 73–80.

Golden, R. M. (1988). A unified framework for connectionist systems. *Biological Cybernetics, 59,* 109–120.

Goldstone, R. L. (Ed.). (2004). 2003 Rumelhart prize special issue honoring Aravind Joshi [Special issue]. *Cognitive Science, 28*(5).

Goldstone, R. L. (Ed.). (2005). 2004 Rumelhart prize special issue honoring John R. Anderson [Special issue]. *Cognitive Science, 29*(3).

Greenberg, J. (1978). Some generalizations concerning initial and final consonant clusters. In J. Greenberg (Ed.), *Universals of human language: Vol. 2. Phonology* (pp. 243–279). Stanford, CA: Stanford University Press.

Grimshaw, J. (1997). Projection, heads, and optimality. *Linguistic Inquiry, 28,* 373–422.

Grossberg, S. (1982). *Studies of mind and brain: Neural principles of learning, perception, development, cognition, and motor control.* Boston: Reidel.

Hale, J., & Smolensky, P. (2006). Harmonic grammars and harmonic parsers for formal languages. In P. Smolensky & G. Legendre (Eds.), *The harmonic mind: From neural computation to optimality-theoretic grammar* (Vol. 1, pp. 393–415). Cambridge, MA: MIT Press.

Halford, G. S., Wilson, W. H., & Phillips, S. (1998). Processing capacity defined by relational complexity: Implications for comparative, developmental, and cognitive psychology. *Behavioral and Brain Sciences, 21,* 803–864.

Hendriks, P., de Hoop, H., & de Swart, H. (Eds.). (2000). Special issue on the optimization of interpretation [Special issue]. *Journal of Semantics, 17*(3–4).

Hinton, G. E., & Anderson, J. A. (Eds.). (1981). *Parallel models of associative memory.* Hillsdale, NJ: Lawrence Erlbaum Associates, Inc.

Hinton, G. E., & Sejnowski, T. J. (1983). Optimal perceptual inference. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (pp. 448–453). IEEE Computer Society Press.

Hinton, G. E., & Sejnowski, T. J. (1986). Learning and relearning in Boltzmann machines. In D. E. Rumelhart, J. L. McClelland, & the PDP Research Group, *Parallel distributed processing: Explorations in the microstructure of cognition* (Vol. 1, pp. 282–317). Cambridge, MA: MIT Press.

Hopcroft, J. E., & Ullman, J. D. (1979). *Introduction to automata theory, languages, and computation.* Reading, MA: Addison-Wesley.

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

Hopfield, J. J. (1984). Neurons with graded response have collective computational properties like those of two-state neurons. *Proceedings of the National Academy of Sciences USA, 81,* 3088–3092.

Hopfield, J. J. (1987). Learning algorithms and probability distributions in feed-forward and feed-back networks. *Proceedings of the National Academy of Sciences USA, 84,* 8429–8433.

Jakobson, R. (1962). *Selected writings: I. Phonological studies.* The Hague, Netherlands: Mouton.

Jordan, M. I. (1986). Attractor dynamics and parallelism in a connectionist sequential machine. In *Proceedings of the Cognitive Science Society* (Vol. 8, pp. 10–17). Hillsdale, NJ: Lawrence Erlbaum Associates, Inc.

Jurafsky, D. S. (1996). A probabilistic model of lexical and syntactic access and disambiguation. *Cognitive Science, 20,* 137–194.

Kager, R. (1999). *Optimality theory.* Cambridge, England: Cambridge University Press.

Kager, R., Pater, J., & Zonneveld, W. (Eds.). (2004). *Constraints in phonological acquisition.* Cambridge, England: Cambridge University Press.

Kohonen, T. (1977). *Associative memory: A system-theoretical approach.* New York: Springer.

Legendre, G., Grimshaw, J., & Vikner, S. (Eds.). (2001). *Optimality-theoretic syntax.* Cambridge, MA: MIT Press.

Legendre, G., Miyata, Y., & Smolensky, P. (1990). Harmonic Grammar—A formal multi-level connectionist theory of linguistic well-formedness: Theoretical foundations. In *Proceedings of the Cognitive Science Society* (Vol. 12, pp. 388–395). Lawrence Erlbaum Associates, Inc.

Legendre, G., Miyata, Y., & Smolensky, P. (2006). The interaction of syntax and semantics: A harmonic grammar account of split intransitivity. In P. Smolensky & G. Legendre, *The harmonic mind: From neural computation to optimality-theoretic grammar* (Vol. 1, pp. 417–452). Cambridge, MA: MIT Press.

Legendre, G., Raymond, W., & Smolensky, P. (1993). An optimality-theoretic typology of case and grammatical voice systems. In *Proceedings of the Berkeley Linguistics Society* (Vol. 19, pp. 464–478). Berkeley, CA: Berkeley Linguistics Society.

Legendre, G., Raymond, W., & Smolensky, P. (2006). Optimality in syntax: II. *Wh*-questions. In P. Smolensky & G. Legendre, *The harmonic mind: From neural computation to optimality-theoretic grammar* (Vol. 2, pp. 183–230). Cambridge, MA: MIT Press.

Legendre, G., Sorace, A., & Smolensky, P. (2006). The optimality theory-harmonic grammar connection. In P. Smolensky & G. Legendre, *The harmonic mind: From neural computation to optimality-theoretic grammar* (Vol. 2, pp. 339–402). Cambridge, MA: MIT Press.

Manning, C. D., & Schütze, H. (1999). *Foundations of statistical natural language processing.* Cambridge, MA: MIT Press.

McCarthy, J. J. (2002). *A thematic guide to optimality theory.* Cambridge, England: Cambridge University Press.

McCarthy, J. J., & Prince, A. (1995). Faithfulness and reduplicative identity. In J. Beckman, L. Walsh Dickey, & S. Urbanczyk (Eds.), *University of Massachusetts occasional papers in linguistics: 18. Papers in optimality theory* (pp. 249–384). Amherst: University of Massachusetts at Amherst, GLSA.

McClelland, J. L. (1979). On the time-relations of mental processes: An examination of systems of processes in cascade. *Psychological Review, 86,* 287–330.

McClelland, J. L., Rumelhart, D. E., & the PDP Research Group. (1986). *Parallel distributed processing: Explorations in the microstructure of cognition* (Vol. 2). Cambridge, MA: MIT Press.

Minsky, M., & Papert, S. (1969). *Perceptrons.* Cambridge, MA: MIT Press.

Misker, J. M. V., & Anderson, J. R. (2003). Combining optimality theory and a cognitive architecture. In *Proceedings of the International Conference on Cognitive Modeling* (Vol. 5).

Newell, A. (1990). *Unified theories of cognition.* Cambridge, MA: Harvard University Press.

Pinker, S., & Prince, A. (1988). On language and connectionism: Analysis of a parallel distributed processing model of language acquisition. *Cognition, 28,* 73–193.

Plate, T. A. (1991). Holographic reduced representations: Convolution algebra for compositional distributed representations. In J. Mylopoulos & R. Reiter (Eds.), *Proceedings of the International Joint Conference on Artificial Intelligence* (Vol. 12, pp. 30–35). San Mateo, CA: Morgan Kaufmann.

Plate, T. A. (1994). *Distributed representations and nested compositional structure.* Unpublished doctoral dissertation, University of Toronto, Toronto, Ontario, Canada.

Plate, T. A. (2003). *Holographic reduced representation: Distributed representation for cognitive structures.* Stanford, CA: CSLI Publications.

Pollack, J. (1988). Recursive auto-associative memory: Devising compositional distributed representations. In *Proceedings of the Cognitive Science Society* (Vol. 10, pp. 33–39). Hillsdale, NJ: Lawrence Erlbaum Associates, Inc.

Pollack, J. (1990). Recursive distributed representations. *Artificial Intelligence, 46,* 77–105.

Pouget, A., & Sejnowski, T. J. (1997). Spatial transformations in the parietal cortex using basis functions. *Journal of Cognitive Neuroscience, 9,* 222–237.

Prince, A., & Smolensky, P. (1991). *Notes on connectionism and harmony theory in linguistics* (Technical Rep. No. CU–CS–533–91). Boulder: Computer Science Department, University of Colorado at Boulder.

Prince, A., & Smolensky, P. (1993/2004). *Optimality theory: Constraint interaction in generative grammar.* Cambridge, MA: Technical report, Rutgers University and University of Colorado at Boulder, 1993. Rutgers Optimality Archive 537, 2002. Revised version published by Blackwell, 2004.

Prince, A., & Smolensky, P. (1997). Optimality: From neural networks to universal grammar. *Science, 275,* 1604–1610.

Prince, A., & Smolensky, P. (2006). Optimality in phonology: I. Syllable structure. In P. Smolensky & G. Legendre (Eds.), *The harmonic mind: From neural computation to optimality-theoretic grammar* (Vol. 2, pp. 3–25). Cambridge, MA: MIT Press.

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning internal representations by error propagation. In D. E. Rumelhart, J. L. McClelland, & the PDP Research Group, *Parallel distributed processing: Explorations in the microstructure of cognition* (Vol. 1, pp. 318–362). Cambridge, MA: MIT Press.

Rumelhart, D. E., McClelland, J. L., & the PDP Research Group. (1986). *Parallel distributed processing: Explorations in the microstructure of cognition* (Vol. 1). Cambridge, MA: MIT Press.

Rutgers Optimality Archive. Retrieved from http://roa.rutgers.edu

Seidenberg, M. S., & MacDonald, M. C. (1999). A probabilistic constraints approach to language acquisition and processing. *Cognitive Science, 23,* 568–588.

Shastri, L., & Ajjanagadde, V. (1993). From simple associations to systematic reasoning: A connectionist representation of rules, variables and dynamic bindings using temporal synchrony. *Behavioral and Brain Sciences, 16,* 417–494.

Smolensky, P. (1983). Schema selection and stochastic inference in modular environments. In *Proceedings of the National Conference on Artificial Intelligence* (Vol. 3, pp. 378–382). San Mateo, CA: William Kaufmann.

Smolensky, P. (1986). Information processing in dynamical systems: Foundations of harmony theory. In D. E. Rumelhart, J. L. McClelland, & the PDP Research Group (Eds.), *Parallel distributed processing: Explorations in the microstructure of cognition* (Vol. 1, pp. 194–281). Cambridge, MA: MIT Press.

Smolensky, P. (1988). On the proper treatment of connectionism. *Behavioral and Brain Sciences, 11,* 1–74.

Smolensky, P. (1990). Tensor product variable binding and the representation of symbolic structures in connectionist networks. *Artificial Intelligence, 46,* 159–216.

Smolensky, P. (1993). Harmonic grammars for formal languages. In S. Hanson, J. D. Cowan, & C. L. Giles (Eds.), *Advances in neural information processing systems* (Vol. 5, pp. 847–854). San Mateo, CA: Morgan Kaufmann.

Smolensky, P. (1995). Constituent structure and explanation in an integrated connectionist/symbolic cognitive architecture. In C. Macdonald & G. Macdonald (Eds.), *Connectionism: Debates on psychological explanation* (Vol. 2, pp. 221–290). Oxford, England: Blackwell.

Smolensky, P. (2006a). Computational levels and integrated connectionist/symbolic explanation. In P. Smolensky & G. Legendre, *The harmonic mind: From neural computation to optimality-theoretic grammar* (Vol. 2, pp. 503–592). Cambridge, MA: MIT Press.

Smolensky, P. (2006b). Formalizing the principles: I. Representation and processing in the mind/brain. In P. Smolensky & G. Legendre, *The harmonic mind: From neural computation to optimality-theoretic grammar* (Vol. 1, pp. 147–205). Cambridge, MA: MIT Press.

Smolensky, P. (2006c). Formalizing the principles: II. Optimization and grammar. In P. Smolensky & G. Legendre, *The harmonic mind: From neural computation to optimality-theoretic grammar* (Vol. 1, pp. 207–234). Cambridge, MA: MIT Press.

Smolensky, P. (2006d). Optimality in phonology: II. Harmonic completeness, local constraint conjunction, and feature domain markedness. In P. Smolensky & G. Legendre, *The harmonic mind: From neural computation to optimality-theoretic grammar* (Vol. 2, pp. 27–160). Cambridge, MA: MIT Press.

Smolensky, P. (2006e). Optimization in neural networks: Harmony maximization. In P. Smolensky & G. Legendre, *The harmonic mind: From neural computation to optimality-theoretic grammar* (Vol. 1, pp. 345–392). Cambridge, MA: MIT Press.

Smolensky, P. (2006f). Tensor product representations: Formal foundations. In P. Smolensky & G. Legendre (Eds.), *The harmonic mind: From neural computation to optimality-theoretic grammar* (Vol. 1, pp. 271–344). Cambridge, MA: MIT Press.

Smolensky, P., & Legendre, G. (2006a). *The harmonic mind: From neural computation to optimality-theoretic grammar: Vol. 1. Cognitive architecture. Vol 2: Linguistic and philosophical implications.* Cambridge, MA: MIT Press.

Smolensky, P., & Legendre, G. (2006b). Principles of the integrated connectionist/symbolic cognitive architecture. In P. Smolensky & G. Legendre, *The harmonic mind: From neural computation to optimality-theoretic grammar* (Vol. 1, pp. 63–97). Cambridge, MA: MIT Press.

Smolensky, P., Legendre, G., & Tesar, B. B. (2006). Optimality theory: The structure, use and acquisition of grammatical knowledge. In P. Smolensky & G. Legendre (Eds.), *The harmonic mind: From neural computation to optimality-theoretic grammar* (Vol. 1, pp. 453–544). Cambridge, MA: MIT Press.

Smolensky, P., & Tesar, B. B. (2006). Symbolic computation with activation patterns. In P. Smolensky & G. Legendre, *The harmonic mind: From neural computation to optimality-theoretic grammar* (Vol. 1, pp. 235–270). Cambridge, MA: MIT Press.

Soderstrom, M., Mathis, D. W., & Smolensky, P. (2006). Abstract genomic encoding of universal grammar in Optimality Theory. In P. Smolensky & G. Legendre, *The harmonic mind: From neural computation to optimality-theoretic grammar* (Vol. 2, pp. 403–471). Cambridge, MA: MIT Press.

Stemberger, J. P., & Bernhardt, B. H. (1998). *Handbook of phonological development from the perspective of constraint-based nonlinear phonology.* San Diego, CA: Academic.

Stevenson, S., & Smolensky, P. (2006). Optimality in sentence processing. In P. Smolensky & G. Legendre, *The harmonic mind: From neural computation to optimality-theoretic grammar* (Vol. 2, pp. 307–338). Cambridge, MA: MIT Press.

Tabor, W. (2000). Fractal encoding of context-free grammars in connectionist networks. *Expert Systems, 17,* 41–56.

Tesar, B. B., & Smolensky, P. (1994). Synchronous-firing variable binding is spatio-temporal tensor product representation. In *Proceedings of the Cognitive Science Society* (Vol. 16, pp. 870–875). Hillsdale, NJ: Lawrence Erlbaum Associates, Inc.

Wilson, C. (2003). Experimental investigation of phonological naturalness. In G. Garding & M. Tsujimura (Eds.), *Proceedings of the West Coast Conference on Formal Linguistics* (Vol. 22, pp. 533–546). Somerville, MA: Cascadilla.