Symbol Taxonomy in Biophonology *

Camila Matamoros & Charles Reiss

Concordia University, Montréal

Abstract

We present phonological parallels to Gallistel & King's (2009) discussion of symbols, variables and function in animal cognition in order to take up Poeppel's (2012) challenge to cognitive scientists to formulate their models in general computational terms that can potentially be translated into the kind of representational and computational behaviors that we might plausibly find in a nervous system.

1 Introduction

Generative phonological theory is formulated as part of the Computational-Representational Understanding of Mind (see Thagard (2005) for accessible

*The authors would like to thank an anonymous reviewer, as well as participants at the conference *Biolinguistic Investigations on the Language Faculty* held in January 2015 in Pavia, for helpful comments.

discussion). In order to unify such a theory with neurobiology it is necessary to map the representations and computations posited by phonologists to the entities that populate neuroscience. As David Poeppel points out, this has not proven to be a simple task for phonology (or any other branch of linguistics, or any other cognitive science) and no straightforward mappings, like '*This* neuron corresponds to *that* syllable' seem even remotely plausible. Unlike some of his neuroscience colleagues, Poeppel's reaction to this "mapping problem" is not to reject phonological theory, but rather to exhort phonologists, and cognitive scientists in general, to try to formulate their models in general computational terms where possible, in terms that can potentially be translated into the kind of representational and computational behaviors that we might plausibly find in a nervous system. In a nutshell, Poeppel favors having the phonologists (and other cognitive scientists) tell the neuroscientists what to look for.

Poeppel knows the linguistics literature, but as an example of the kind of low-level characterization he is looking for he refers to Gallistel & King (2009, henceforth G&K) whose primary examples of cognitive behavior come from insect navigation. The spirit of G&K, as reflected in their Preface (p.ix), is these lines: There is no reason to think that our understanding of the computational capacity of biological systems is nearly as explicit as our understanding of the basic math and logic of a process like, say, navigation by dead-reckoning, so let's characterize the operations and representations that we know are needed in insect navigation from the perspective of theories of computation, and provide this analysis to neuroscientists so that they can figure out how to get this representational and computational behavior out of biological nervous systems. This perspective of G&K leads them to discuss fundamental notions such as variables and functions from a computational perspective and to talk about the prospects for understanding these notions under various approaches to brain science. A large part of G&K is devoted to arguing that connectionist neural network models and other dominant models of brain function based, for example, on patterns of electrical activity in the brain, have been utter failures in their attempts to model cognition, exactly because they have not and, G&K argue, cannot model variables and functions in the open-ended productive manner required for animal cognition, even at the level of insect navigation.

One reason Poeppel discusses G&K's insect studies despite his own familiarity with linguistics is that similarly explicit work concerning computational fundamentals in linguistics is hard to find. In this paper, we hope to demonstrate that phonological theory, in a fairly standard form represented by Bale and Reiss (Forthcoming) is quite consistent with the spirit of work like G&K, although linguists tend not to be as explicit about the issues as G&K are. In other words, we can think of generative phonology as high-level theoretical neuroscience—we are just making explicit some of Poeppel's suggestions.

2 Overview of Gallistel and King (2009)

When Poeppel asks how the primitive units of analysis of the cognitive sciences map onto the primitive units of analysis of the neurosciences, he stipulates that good answers come from "theoretically well-motivated, computationally explicit, and biologically realistic characterizations of [brain] function." He exhorts linguists and other cognitive scientists to formulate theories in computational terms in order to guide neurobiologists in their search for more explanatory, not just correlative, data. G&K's book length treatment helps us to understand some of the ideas that cognitive scientists should use in their discussions, ideas that are not specific to particular cognitive domains.

The main conclusions from G&K's work are quite simple: the evidence from cognition shows there must be an addressable read/write memory mechanism in brains that

- encodes information received into symbols (writes)
- locates the information when needed (addresses)
- and transports it to computational machinery that makes productive use of the information (reads)

As G&K point out, "[s]uch a memory mechanism is indispensable in powerful engineered computing devices, and the behavioral data imply that brains are powerful organs of computation", as well. This read/write memory instantiates a system of symbols, variables and functions that are somehow implemented in neural tissue. In this paper, we explore how the symbols and variables, and the functions operating over them can be expressed in "our" phonological theory in a way that potentially contributes to explanatory neuroscience research.

2.1 Symbol taxonomy

G&K explain that for the brain to compute stimuli from experience, it must be able to generate representations of the world. This concept of representations depends on a concept of functions. Functions are operations that take variables as input; these variables get their values from symbols and the function outputs a symbol that is determined by the value of the symbol that gave the input variable its value. Therefore, a discussion of symbols is required to build up the picture of how computation works.

For G&K, symbols in computation must be physically realized entities that are manipulated by the programming and architecture of a computational system. Because the brain is physically limited in power and space, and because scientific reasoning leads us to propose the simplest possible hypothesis, a desirable model is one with a small number of primitive symbols and basic processes for combining them. By combining primitive symbols, more complex symbols can be created when new stimuli require new representations.

G&K (p.79) themselves use a linguistic example to illustrate the idea of a basic symbol inventory in both engineered systems and natural systems:

> No language in the world has a word for the message, "After the circus, I'm going to the store to get a quart of skim milk." ... Minimizing the number of atomic data is desirable in any symbol system as it reduces the complexity of the machinery required to distinguish these data. This is why computers use just two atomic data and likely why nucleotide sequences use only four.

The atomic data enter into combinations that lead to more complex symbols, according to the followng taxonomy:

(1) Gallistel & King's symbol taxonomy (p.79)

- atomic data
- data strings
- nominal symbols
- encoding symbols
- data structures

For example, atomic data can be ordered in strings, and strings can be arranged into complex structures like trees with precedence and containment relations among strings. Our modest goal in this paper is to show that standard phonological theory provides a similar model of representations. In this discussion we will focus on the combinatoric and structural aspects of phonological feature systems, although there may be interesting comparisons to be made with G&K's discussions of nominal symbols and encoding symbols.[1]

2.2 Combinatoric explosion

The effect of having a hierarchical taxonomy of symbols is that a relatively small number of lower level symbols can be combined into a very large number of higher level symbols. As we will discuss below, the combinatorial space of a simple system can lead to combinatoric explosion of the space of complex symbols. However, G&K explain that this is in fact a *desirable* trait since it removes the necessity of a look-up table for complex symbols. This idea is applicable not only to the symbols of mental representation, but to systems like the genetic code which generates the diversity of life from sequences of just four basic elements, the nucleotide 'letters' T,G,C and A. We will see that phonological symbols also give us a massive power of descriptive variety, exactly what we want in a theory of Universal Grammar.

In their chapter on symbols, G&K also address the issue of storage space. They explain that the number of possible data strings and nominal or encoding symbols is so great as to be considered infinite, for all practical purposes, but the resources required for realizing and storing these are clearly limited.

[1] In brief, symbols are 'nominal' if there is an arbitrary mapping between their form and their referent, a property that clearly holds of the mapping between phonological mental representations and their phonetic (articulatory and acoustic) correlates (see below). It is not immediately clear to us whether phonological representation has any parallel to the notion of encoding symbols that G&K introduce.

A biological plausible computational system must therefore start with a small number of simple symbols and be able to combine them and store relevant combined symbols in response to received stimuli—the system does not store all the possibilities, only those that are actually encountered.[2] The example (p.82) of the "picture space" of a digital camera illustrates this point:

> The camera constructs a symbol for an image during the interval when the shutter is open. If it is a six-megapixel camera, the symbol consists of 6 million 24-bit binary numbers. The number of different symbols of this kind (the number of different pictures that may be taken with a 6-megapixel camera) is [for practical purposes—cm and cr[3]] infinite, just as is the number of possible images. But, of course, the number of different pictures that even the most avid photographer will take is finite (...) The camera can represent an infinite range of possibilities with finite resources, because it constructs its symbols on demand. The possibilities are infinite, but the actualities are finite. (82)

We'll see how phonological systems mirror this idea of combinatoric capacity built on a simple system.

[2]The system has to parse, using working memory, whatever is encountered, but only store in long-term memory what cannot be (re)generated by the computational system. For example, encountered phonological representations like the segment string [dɔg] *dog* must be stored, but the syntactic structure of a sentence like *The cat bit the dog* does not have to be stored in long-term memory once it is parsed.

[3]It is actually $2^{24^{6,000,000}}$

2.3 Variables and functions

Symbols and data structures that encode information about the world are key components of a computational system, but algorithms that calculate over their values must also have some way of finding the relevant data. The process of locating a symbol is called *addressing*. An address for a symbol, or set of possible symbols in a category is called a *variable*. In concrete terms the implementation of a variable is a location at which a symbol can be found—the symbol found at that address is the current value of the variable. The variable and its value are different pieces of information. The variable is the address, the value is the symbol found at that address. A category variable is the address that points to the addresses of possible values within that category. G&K give many examples of how insects must use variables. In navigation, since the position of the sun in calculating their direction relative to the North-South axis is not a stable value, computation for navigation must include a variable for the sun's position. Other animal examples show the need for variables for a wide variety of dimensions including the sweetness of available nectar or the rotting rate of a cached food supply. In the following sections, we present parallels to G&K's examples of symbol taxonomy, combinatoric explosion and the role of variables from the domain of phonology.

3 Symbol taxonomy in phonology

According to Distinctive Feature Theory (e.g. Jakobson et al. (1967)) the atomic symbols of phonology are not segment or phonemes like /p/, /t/, /a/, *etc.*, but rather the members of a set of *features* and the members of a set of *values*. The features correspond to a phonetic dimension with complex acoustic and articulatory correlates. For example, the feature HIGH correlates somewhat, via complex transduction processes, with location of the first formant in a vowel spectrogram and the relative height of the tongue in the mouth. The set of values in the model we adopt has just two members, + and −. Members of the set of values and the set of features co-occur in phonological representations as complex symbols like +HIGH or −HIGH. A persistent ambiguity in terminology in the field is the use of the simple term *feature* to refer to such *valued features*, but we'll try to distinguish the two concepts consistently here. Note that a valued feature is just an ordered pair whose first member is a value and whose second member is a feature. The valued feature is thus the first step up in the phonological symbol taxonomy from the primitive values and features. The next level of organization will be to combine valued features into sets that correspond to phonological segments.

Let's look at the kind of phenomena that lead phonologists to posit discrete features that can combine with only two discrete values. We use a textbook example of Turkish from Isac and Reiss (2013, 128ff) to explore a phenomenon called vowel harmony. First examine the forms in (2) that show the eight vowels of Turkish in a list of words.

(2) The eight Turkish Vowels

a. ip 'rope'

b. kıl 'hair'

c. sap 'stalk'

d. uç 'tip'

e. son 'end'

f. öç 'revenge"

g. gül 'rose'

h. ek 'joint'

The eight different vowels in these words can be described along 3 axes as in (3): Backness, Height, and Rounding. Backness refers to how far back the tongue is placed in the mouth when the vowel is produced, therefore the sounds 'i', 'ü', 'e', and 'ö' are -BACK whereas 'ı', 'u', 'a', and 'o' are +BACK. Height refers to the height of the tongue relative to the roof of the mouth, so in the chart, those on the same line as 'i' are +HIGH while those in line with 'e' are -HIGH. Finally, the feature ROUND refers to lip rounding.

(3) Feature analysis of Turkish vowels

	-Back		+Back	
	-Round	+Round	-Round	+Round
+High	i	ü	ı	u
-High	e	ö	a	o

Vowels and consonants are segments. That is, they are the level of organization of language that we represent with phonetic alphabet symbols. However, each segment symbol is actually an abbreviation for a set of valued features. These sets are consistent, that is, abstracting away from so-called *contour segments*, they cannot contain incompatible values; for a given feature F, a segment cannot contain both -F and + F. The sets corresponding to the vowels /i/ and /u/ are given in (4), which is just another way of representing some of the information in (3).

(4) Specification of sets of valued features corresponding to the vowels /i/ and /u/

$$/\mathrm{i}/ = \begin{bmatrix} \text{-BACK} \\ \text{-ROUND} \\ \text{+HIGH} \end{bmatrix} \quad /\mathrm{u}/ = \begin{bmatrix} \text{+BACK} \\ \text{+ROUND} \\ \text{+HIGH} \end{bmatrix}$$

There are other features relevant to vowels across languages but the three features shown in (4) are sufficient for our discussion of Turkish.

To understand the usefulness of the featural analysis of segments and to understand why we think of features and values as basic symbols, as building blocks for segments, let's see how they allow us to understand vowel harmony, which we present with some simplifications for expository purposes.

In the following table, notice that the vowel of the suffix *-ler/-lar* is identical to the preceding vowel, suggesting that the vowel in the suffix copies the set of features from the vowel immediately before it.

(5) Some Turkish singular/plural noun pairs

SINGULAR	PLURAL	MEANING
dev	devler	giant
kek	kekler	cake
ters	tersler	contrary
can	canlar	soul
tarz	tarzlar	type
kap	kaplar	recipient

With more data, we see that even when the root vowel is not /e/ or /a/, the suffix vowel is still one of those two—there are only two forms for the suffix.

(6) More Turkish data

SINGULAR	PLURAL	MEANING
ip	ipler	rope
öç	öçler	vengeance
gül	güller	rose
ek	ekler	junction
kıl	kıllar	body hair
sap	sapar	stalk
uç	uçlar	edge
son	sonlar	end

If you look at the feature chart in (3) you see that the the suffix vowel is not copying or agreeing with the whole set of features from the preceding vowel, but only the information relative to the backness feature.

Phonologists say that the vowels i, e, ü, ö are -BACK to mean that these symbols stand for sets that have the ordered pair -BACK as a member. Similarly, we say that the vowels u, o, ı, a are +BACK to mean that these symbols stand for sets that have the ordered pair +BACK as a member.

(7) Front and back vowels differ with respect to the value of the feature BACK

- i, e, ü, ö are -BACK
- u, o, ı, a are +BACK

The suffix vowel is always -ROUND and always -HIGH. But the suffix vowel is either /a/ or /e/ and the choice is determined by agreement with the preceding vowel's value for BACK.

This simple example already illustrates the idea of a symbol hierarchy that parallels the discussion of G&K. Features and values combine into ordered pairs of valued features; valued features combine as members of sets to constitute segments; segments can be ordered in strings as the phonological part of roots and suffixes like /ip/; and, of course, these morphemes can combine into words like /ipler/ which ultimately are strung into complex symbols referred to as sentences. The vowel harmony process shows that the generalization about which vowels occur in the plural can be expressed as a function of the preceding vowel, understood as a set of valued features. An alternative that just listed the plural form for each singular in a look-up table would fail to capture the generalization of featural agreement, and also fail to explain the fact that Turkish speakers can generate plurals that they have never encountered in accordance with the same generalization we have shown. G&K (p.51ff) go to great lengths to demonstrate the implausibility of approaches to cognition that rely heavily on look-up tables, in part because of the problem of combinatorial explosion that we touch on in the next section.

Linguistic data structures thus are not restricted to information from a single module of grammar. Features and values and segment strings are phonological, but a morpheme is a data structure that contains several kinds of information, as in Fig. 1. The morpheme is a structure comprised of linked phonological, syntactic, and semantic representations. The lexicon of a language consists of a list of morphemes (or 'vocabulary items' in some work) perhaps with some structure among the members.

We have thus laid out a taxonomy of symbols from the atomic features and values up to the level of a lexicon. Other branches of linguistics in various frameworks provide the kind of explicit models for the syntactic (e.g. Adger, 2003; Bresnan et al., 2015) and semantic features (e.g. Jackendoff, 1990) of complex symbols that we have presented for phonology.

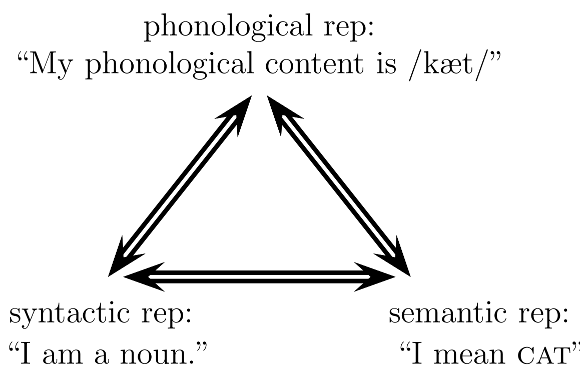

Figure 1: The morpheme of *cat*

We have suggested in this section that features are part of what Poeppel calls "the infrastructure of linguistics ... [that] provides a body of concepts that permit linguists and psychologists to make a wide range of precise generalizations about knowledge of language." (35) Features are among the crucial theoretical tools that linguists use to do the work that Poeppel hopes

Table 1: A hierarchical taxonomy in language

Level	Notation	Name
Atomic	{+, − } {HIGH, BACK, ROUND, *etc.* }	coefficients and features
Ordered pairs	<+ROUND>	(valued) features
Sets of ordered pairs	{ <+ROUND>, <-HIGH>, <+BACK>} abbreviated [k], [i], [θ], [p] *etc.*	segments
Strings of sets of ordered pairs	[kæt]	strings (phonology of morphemes)
Tripartite structures	[kæt] and the rest of Fig.1	morphemes
Lists of morphemes	(a complex structured list)	mental lexicons

neurobiologists can come to understand. In addition to their established theoretical utility, there is new evidence that features are neurobiologically plausible. Mesgarani et al. (2014) sought to identify the resolution at which speech sounds are received, deciphered, and encoded by the brain. They found that individual and small groups of neurons are sensitive to phonetic features in speech, supporting our hypothesis that features are indeed the atomic symbols that G&K tell us to look for.

Once we adopted a feature model, we then showed how features and values are organized into progressively more complex structures, providing a symbol taxonomy much like that discussed by G&K. In the next section, we will examine how a small number of coefficients and features can combine to give us the great diversity of speech sounds that constitute human languages.

4 Combinatoric explosion in phonology

Once we have a system in which basic symbols, like features and values, combine into sets that constitute segments, we have a system with the potential for combinatoric explosion. As shown by Turkish, a set of three features with two possible values for each allows for $2^3 = 8$ different vowels. For each additional feature, the number of different segments doubles, so that with, say four features, we get $2^4 = 16$ segments and with a system of just ten features, we get $2^{10} = 1024$ different segments.

For reasons discussed in Reiss (2012), many phonologists have come to believe that, although a set of valued features constituting a segment must be consistent, with no conflicting values for a given feature, it need not be *complete*. That is, a segment may be represented by a set of valued features that lacks a specification for some feature. For example, the form of the plural suffix in Turkish stored in long-term memory may contain a vowel that is neither /e/ nor /a/, but rather the intersection of these two sets, namely {-High, -Round}. The phonology of the language fills in a value for the feature Back depending on the context of occurrence (the feature of the preceding vowel), and the 'incomplete' vowel ends up being identical to either /e/ or /a/. This idea, that segments may be incomplete, is called underspecification. For our purposes, the crucial point is that the system combining features and values becomes even more explosively combinatoric, since there are now *three* possible states for each feature in a segment: combined with

"+", combined with "−", and *absent*.

This possibility of having underspecified segments has a tremendous effect on the space of possible phonological systems because we move from powers of 2 to powers of 3. Now, allowing underspecification, three features gives us $3^3 = 27$ segments instead of $2^3 = 8$; four features gives $3^4 = 81$ segments instead of 16; and ten features gives $3^{10} = 59,049$ segments instead of a mere 1024.

Phonologists typically assume that there are about twenty to thirty phonological features provided by the language faculty (but see Hale et al. (2007) for arguments that there may be a much greater number of innate features). For our discussion of combinatorics, let's assume a language faculty that provides just four features, a very modest number. As we have seen, a set of four features allows us to define $3^4 = 81$ segments. Given this universal inventory of possible segments, we can define a set of possible segment inventories. A given inventory is defined by determining whether each of the 81 possible segments is a member of that inventory. In other words, each inventory can be characterized by a unique sequence of 81 YES/NO answers depending on the presence of each segment in the given inventory. There are 2^{81} such sequences, and so, with just four features we can define $2^{81} \sim 2.4 \times 10^{24}$ possible inventories, which Wolfram Alpha estimates to be up to 10,000 times the number of grains of sand on earth. That many from just four features!

Gallistel and King tell us that this is exactly what we need:

> What is needed is an architecture that combats combinatoric explosions with combinatorics. The key to that architecture is a read/write memory. It must be possible to store sequences that actually occur in a memory capable of storing a great many [. . .] sequences, drawn from the essentially infinite number of possible such sequences, and to compare those stored sequences to whatever sequences may prove to be relevant. This architecture uses memory and combinatorics to cope with the finitude of the actual.(136)

The wrong way to look at our trivial combinatoric result is to ask "How can a child search the space of possible segment inventories to find that of the target language?" Coming back to the digital camera analogy given by G&K, learners don't have to store every possible inventory/image—they just have to be able to represent a particular one. In order to find this inventory, they only need to parse input into segments in terms of the four features and three states (+, −, *absent*) that we have endowed them with.

5 Variables and functions in phonology

The discussion by G&K of the necessity for variables and functions as primitives of cognition also finds support in phonology. G&K discuss a fundamental fact of mathematics concerning functions: functions of two or more variables cannot be built by composing functions of a single variable. They conclude, based on the kinds of functions that must be computed in insect navigation and other tasks, that basic cognition must allow for one argument and two argument functions as basic properties—nervous systems need to compute such functions. (With those, it is possible to build up functions of three or more arguments.) In this section, we will briefly illustrate how the notions of variables and functions appear to be needed for phonological cognition, too.

5.1 Functions of one variable: Reduplication

The phenomenon of reduplication allows us to illustrate that phonology makes use of functions of one variable. Consider the following singular/plural pairs from the Australian language Warlpiri discussed in detail by Isac and Reiss (2013).

SINGULAR	PLURAL	GLOSS
kurdu	kurdukurdu	*child/children*
kamina	kaminakamina	*girl/girls*
mardukuja	mardukujamardukuja	*woman/women*

It is relatively straightforward to say that 'child' is pronounced /kurdu/ in Warlpiri, but it doesn't really make sense to ask how the plural morpheme is pronounced. Clearly there is no one pronunciation, rather the plural corresponds to a variable, x, such that x has the same form as the noun with which it concatenates. We can think of the process that spells out the phonology of the Warlpiri plural as a function of a single variable, PLURAL(x), with x having as its domain the set of Warlpiri nouns. The output of this function is the concatenation of two copies of the (phonological component) of the input: $\text{PLURAL}(x) = x^\frown x$. G&K define a variable as an address in memory. The content of that address at a particular time is the current value of that variable. Given a singular noun, Warlpiri speakers can generate the corresponding plural even if they have never encountered it before, so, it appears that the phonological content of the singular can be assigned as the current value for the variable x; that is, a phonological string symbol can be copied and written into memory at address x, and the function PLURAL can be applied to x to compute an output $x^\frown x$.

Another simple example of a function of one variable is a phonological rule that nasalizes a vowel before a nasal consonant. Semi-formally, we can say that such a rule corresponds to a function NASAL(x) that maps a vowel to its nasalized counterpart. For details on how we implement such a rule, both feature-filling and feature-changing variants, see Bale et al. (2014).

5.2 Functions of two variables: Computing Identity

G&K (49) point out that the "ability to realize functions of at least two arguments is a necessary condition for realizing functions of a non-trivial nature" since, generally, "[f]unctions of one argument cannot combine to do the work of functions with two". We can now illustrate a two argument function needed for phonological computation. Reiss (2003) argues that certain phonological processes require an evaluation of whether two segments are or are not identical. There are processes that only apply if two segments are identical and there also exist processes that only apply if two segments are not identical. In order to determine whether the segments in two different positions (say, the two consonants flanking a vowel) are identical, we need a function with two arguments $\textsc{Identical}(x, y)$, where each variable corresponds to one of the segments to be compared. See Reiss (2003) for further discussion of the internal workings of such a function—in brief, the function has to quantify over the features in the segment sets and check for each feature F whether the value associated with F is the same in the two segments. This computation requires further variables corresponding to features and values below the segment level, but we can see that even abstracting to the segment level, we need functions of at least two variables in phonology, just as G&K propose we need "functions of a non-trivial nature" to explain well-understood processes in animal cognition.

6 Conclusions

In this paper, have presented some of the "infrastructure of linguistics" and we have shown that G&K's animal-inspired taxonomy of symbols and the functions of a read/write addressable memory are compatible with the formal model we apply to describing and investigating phonological processes.

Hopefully this attempt to define a 'biophonology' will be of some use and motivation to linguists and neurobiologists alike in continuing the emerging dialogue and movement toward an explanatory neurobiology.

References

Adger, David. 2003. *Core syntax*. Oxford: Oxford University Press.

Bale, Alan, Maxime Papillon, and Charles Reiss. 2014. Targeting underspecified segments: A formal analysis of feature changing and feature filling rules. *Lingua* 148:240–253.

Bale, Alan, and Charles Reiss. Forthcoming. *Phonology: A formal introduction*. MIT Press.

Bresnan, Joan, Ash Asudeh, Ida Toivonen, and Stephen Wechsler. 2015. *Lexical-functional syntax*. Wiley-Blackwell, 2nd edition.

Gallistel, C. R., and Adam Philip King. 2009. *Memory and the computational brain: why cognitive science will transform neuroscience*. Chichester, West Sussex, UK: Wiley-Blackwell.

Hale, Mark, Madelyn Kissock, and Charles Reiss. 2007. Microvariation, variation and the features of Universal Grammar. *Lingua* 117:645–665.

Isac, D., and C. Reiss. 2013. *I-language: an introduction to linguistics as cognitive science*. Oxford University Press, USA, 2nd edition.

Jackendoff, Ray. 1990. *Semantic structures*, volume 18 of *Current studies in linguistics*. Cambridge, Mass.: MIT Press.

Jakobson, R., G. Fant, and M. Halle. 1967. *Preliminaries to speech analysis*. MII Press.

Mesgarani, Nima, Connie Cheung, Keith Johnson, and Edward F Chang. 2014. Phonetic feature encoding in human superior temporal gyrus. *Science* 343:1006–1010.

Reiss, Charles. 2003. Quantification in structural descriptions: Attested and unattested patterns. *The Linguistic Review* 20:305–338.

Reiss, Charles. 2012. Towards a bottom-up approach to phonological typology. In *Towards a biolinguistic understanding of grammar: Essays on interfaces*, ed. A.M. Di Sciullo, 169–191. Amsterdam: John Benjamins Publishing Company.

Thagard, Paul. 2005. *Mind: introduction to cognitive science*. Cambridge, Mass.: MIT Press, 2nd ed edition.