The computational nature of phonological generalizations*

Jeffrey Heinz

January 22, 2015

Abstract

This chapter studies the nature of the typology of phonological markedness constraints and the nature of the typology of the transformation from underlying to surface forms from a computational perspective. It argues that there are strong computational laws that constrain the form of these constraints and transformations. These laws are currently stated most clearly in terms of the so-called subregular hierarchies, which have been established for stringsets (for modeling constraints) and are currently being established for string-to-string maps (for modeling the transformations). It is anticipated that future research will reveal equally powerful laws applicable to non-string-based representations. Finally, this chapter argues that these laws arise as a natural consequence of how humans generalize from data.

Contents

*I am in indebted to Jane Chandlee, Rémi Eyraud, Bill Idsardi, Adam Jardine, and Jim Rogers for invaluable discussion. I thank Jane Chandlee for helpful comments on an earlier draft. I also thank Regine Lai, Huan Luo, and Amanda Payne. Of course I assume full responsibility for flaws present in this chapter.

1 Illuminating the phonological component of grammar

Wilhelm von Humboldt's phrase "language makes infinite use of finite means" (1836/1999) is oft-cited by Chomsky since it not only encapsulates an important characteristic natural language, but also it highlights why *generative* grammars play an important role in understanding this aspect of language. In brief, the generative grammars are the finite means, but the linguistic knowledge they represent can be applied to unboundedly many linguistic forms. The psychological reality of generative grammars is the powerful scientific hypothesis which underlies all work in generative linguistics.

In this chapter, we will study generative grammars from both a typological and computational perspective. The same Wilhelm von Humboldt is reported (Frans Planck, p.c.) to have written that in order to do linguistic typology, two encyclopedias are necessary. The first is "an encyclopedia of categories" and the second is "an encyclopedia of types." The encyclopedia of categories provides an ontology with which the encyclopedia of types—the linguistic generalizations—can be studied. The object of inquiry is natural language and the linguistic generalizations. But the light we shine on them comes from the encyclopedia of categories.

This chapter will argue that the theory of computation provides a meaningful and insightful encyclopedia of categories, with which linguistic generalizations ought to be studied. (My discussion is limited to phonological generalizations, though much important work in a similar vein exists for other kinds of linguistic generalizations (Chomsky, 1956; Gazdar and Pullum, 1982; Shieber, 1985; Rogers, 1994; Kobele, 2006; Graf, 2013).) I will endeavor to explain that when phonological generalizations are studied in this light, there are computational laws which govern important aspects of their nature. I will also argue that current phonological theory does not account for these laws, and I will make suggestions as to how phonological theory might be modified to do so.

In this way, the goals of this chapter are similar to the goals of Charles Kisseberth in his 1970 paper "On the Functional Unity of Phonological Rules." There he writes

> I will show ... that a rather rich set of diverse phenomenon is related in a complex, but quite coherent way. The theory of phonology has hitherto been blind to phenomena of this sort... and I will attempt to make some suggestions about the kind of apparatus the facts... seem to require that a theory of phonology contain. I am not, however, principally interested in proposing detailed formalism; instead I would like to encourage phonologists to *look* at the phonological component of a grammar in a particular way. [emphasis in original] (Kisseberth, 1970a, p. 293)

Kisseberth argued that important generalizations in languages were missed by not paying attention to the functional unity of phonological rules. The introduction of surface constraints into phonological theory followed, and later became one of the cornerstones of Optimality Theory (Prince and Smolensky, 1993, 2004).

Similarly, I am arguing that important generalizations in languages are being missed by not paying attention to the computational nature of phonological generalizations. Yes, I am talking about computational generalizations of phonological generalizations. I argue these meta-generalizations are important because they too suggest a conspiracy of sorts: phonological generalizations across languages are distinct, but they exhibit a very strong tendency to exhibit particular computational properties.

I will argue that, when phonological generalizations are studied under this light, the hypothesized computational laws are *sufficiently expressive* to account for the impressive range of cross-linguistic variation, and are simultaneously *very restrictive* in the sense that strong predictions are made about which logically possible phonological generalizations are not humanly possible ones. I will argue that in this respect these computational laws better match the attested typology than what is predicted by classical Optimality Theory (and many of its variants), which, I will argue, is *neither* sufficiently expressive nor restrictive. I will also argue that the restrictive nature of the computational laws help answer questions about how such phonological generalizations can be learned.

The arguments that I am making in this chapter are not novel. They have previously been published in articles and conference proceedings. This chapter thus provides a road-map for

phonologists of this literature, and attempts to present a unified, overarching perspective on both the importance of this computational encyclopedia of categories, and its implications for a theory of phonology.[1]

2 What is phonology?

The fundamental insight in the 20th century which shaped the development of generative phonology is that the best explanation of the systematic variation in the pronunciation of morphemes is to posit a single underlying mental representation of the phonetic form of each morpheme and to derive its pronounced variants with context-sensitive transformations. This development, present in Chomsky (1951); Halle (1959), was perhaps most stated fully and completely with Chomsky and Halle (1968), and persists in Optimality Theory (Prince and Smolensky, 2004) today.

Thus there is a point of agreement between different theories of phonology, which is stated in (1).

(1) There exist underlying representations of morphemes which are transformed to surface representations.

As a result of this fundamental insight, every particular theory of phonology grapples with three fundamental questions, shown in (2).

(2) a. What is the nature of the abstract, underlying, lexical representations?
b. What is those nature of the concrete, surface representations?
c. What is the nature of the transformation from underlying forms to surface forms?

I would like to give some examples of how phonological theories aim to answer these questions. It is not possible in this chapter to comprehensively survey the range of answers that have been offered. Therefore, I only highlight some answers (and only in very broad strokes).

Rule-based theories, as exemplified by Chomsky and Halle (1968), for example, have argued that the abstract underlying representations are subject to language-specific morpheme structure constraints (MSCs). The transformation from underlying forms to surface forms are due to language-specific rules, which are applied in a language-specific order. Constraints on surface representations were, generally speaking, not part of the ontology of these theories, and therefore were not posited to have any psychological reality. Such generalizations—the phonotactic generalizations—were derivable from the interaction of the MSCs and the rules.

On the other hand, in classic Optimality Theory (Prince and Smolensky, 1993, 2004), there are no constraints on underlying representations (richness of the base), but there are psychologically real, universal constraints on surface forms (markedness constraints). The transformation from underlying forms to surface forms is formulated as an *optimization* over these markedness constraints, in addition to constraints which penalize differences between surface and underlying forms (so-called faithfulness constraints). While both the markedness and faithfulness constraints are universal, their relative importance is language-specific. So in ev-

[1] The roadmap is not exhaustive. Notable earlier research which examine the nature of phonological generalizations from a computational perspective but which will not receive as much discussion as it should includes Potts and Pullum (2002) and Graf (2010b).

ery language the surface pronunciation of an underlying representation is predicted to be the optimal form (the one that violates the most important constraints the least), though what is optimal can vary across languages because the relative importance of the constraints can vary across languages.

These two theories are radically different in what they take to be psychologically real. The ontologies of the theories are very different. Perhaps this is most clear with respect to the concept of the phoneme. Phonemes exist as a consequence of the ontology of rule-based theories, but they do not as a consequence of the ontology of OT. This is simply because phonemes are a kind of MSC; underlying representations of morphemes must be constructed out of them, and nothing else. In OT, there are no MSCs and hence there are no phonemes. Consequently, generalizations regarding complementary distribution are explained in a very different manner in the two theories, and they promote different views of the notion of *contrast*. Despite these differences however, there is an important point of agreement: In both theories, complementary distribution of speech sounds in surface forms is the outcome of a transformation of underlying forms to surface forms.

This is the point I wish to emphasize: neither theory abandons the fundamental insight stated in (1).[2] The theories offer radical different answers to the questions in (2), but *they agree on the questions being asked.*

Like earlier research in generative grammar, research in computational phonology agrees with the insight in (1) and the questions being asked in (2). In this chapter we ask three derivative questions. In theories like SPE, which posit morpheme structure constraints, what does the theory of computation bring to the nature of these generalizations regarding underlying representations? In theories like OT, which posit markedness constraints, what does it bring to the nature of these phonotactic generalizations? And for all theories of phonology, What does it bring to the study of the nature of the transformations?

The theory of computation provides a way to answer these questions. The encyclopedia of categories it provides allows these different generalizations to be classified according to computational criteria. What makes this approach valuable is that it is about as atheoretical as one can get. This is because it explicitly separates the intensional descriptions of the generalizations from their extensions. The intensional description of the generalization is the one given by a phonologist in their grammatical description of the generalization. It is the 'finite means' in von Humboldt's sense. The extension of this intensional description is one that typically describes an infinite-sized object. It is the 'infinite use' in von Humboldt's sense.

Mathematically, this infinitely-sized object exists. It is like a perfect circle, a set of infinitely many points each exactly the same distance from a center. But we can never see the object in its entirety. We cannot see an infinity of points, even if we know they are there. The situation with linguistic generalizations is similar. The extension is there, but they cannot be written down in their entirety since they are not finite. But we can write down a grammar which can be understood as generating the infinite set, in the same way that a perfect circle can be generated by specifying a center point and a distance, the radius.

The same perfect circle can be described in other ways as well. If we employ the Cartesian plane, we could generate a circle with an equation of the form $(x-a)^2 + (y-b)^2 = r^2$ where the r is the radius of the circle and (a, b) is its center. The equation is interpreted as follows: all and only points (x, y) which satisfy the equation belong to the circle. The equation is an

[2]It is true that periodically some work is published in that direction, for example the work on output-to-output correspondence [and others](Benua, 1995, 1997).

intensional description and the set of points, the circle, is its extension.

We can also describe a circle on a plane with polar coordinates instead of Cartesian ones. Recall that polar coordinates are of the form (r, θ) where r is the radius and θ is an angle. The equation $r = 2a\cos(\theta) + 2b\sin(\theta)$ provides the general form of the circle with the radius given by $\sqrt{a^2 + b^2}$ and the center by (a, b) (in Cartesian coordinates). The polar equation is interpreted like the Cartesian one: all and only points (r, θ) which satisfy the equation belong to the circle.

There are some interesting differences between these two coordinate systems. Each point in the Cartesian system has a unique representation, but each point in the polar system has infinitely many representations (since the same angle can be described in infinitely many ways, e.g. $0° = 360° = 720° = \ldots$). If the center of the circle is the origin, the polar equation simplifies to $r = a$ whereas the Cartesian equation remains more complicated $x^2 + y^2 = r^2$. Thus, the polar equation $r = 4$ and the Cartesian equation $x^2 + y^2 = 16$ are different equations with different interpretations, but they describe the same unique circle: one of radius four centered around the origin. The two equations differ intensionally, but their extension is the same.

It seems strange to ask which of these two descriptions is the 'right' description of a circle. They are different descriptions of the same thing. Some descriptions might be more useful than others for some purposes. It also interesting to ask what properties the circles have irrespective of a particular description. For instance the length of the perimeter and the area of a circle are certainly relatable to these descriptions, but they are also in a sense independent of the particulars. The perimeter and area depend on the radius but not the center, though both appear in the equations. This suggests that the radius is a more fundamental structure to a circle than its center, though both certainly matter.

The analogy I wish to draw is that rule-based and OT-theoretic formalisms are like the Cartesian and polar systems. The analogy is far from perfect, but it is instructive. Both rule-based and OT analyses provide descriptions of platonic, infinitely sized objects. In many cases, but not all, the two formalisms describe the same object, insofar as the empirical evidence allows.

What is this object? The transformations from underlying forms to surface forms can be thought of as a *function*, in the mathematical sense of the word. Another word for function becoming prevalent in the phonological literature is *map* (Tesar, 2014). There are three parts to a function. One, there is its domain, which is the set of objects the function applies to. Two, there is its co-domain, which is the set of objects to which the elements of the domain are mapped. Three, there is the map itself, which says which domain elements are transformed to which co-domain elements. Thus to specify a function, one needs to provide a description of its domain, its co-domain, and a description of which domain elements become which co-domain elements.

This lines up nearly perfectly with the fundamental questions of phonological theory. The underlying representations correspond to the domain. The surface representations are the co-domain. And the transformation from underlying to surface forms is the map from domain elements to co-domain elements. From this perspective, describing the phonology of a language requires describing aspects of this function, regardless of whether the function is described intensionally with SPE-style or OT grammars.

Further, in linguistic typology we are actually interested in the *class* of such functions that correspond to *possible* human phonologies. If the phonologies of languages are circles we would

be interested in the universal properties of circles and the extent of their variation. Circles are pretty simple, so the answers are straightforward. All circles have a center and a radius, but their centers can be different points and their radii can have different lengths. What universal properties do phonological functions share? What kind of variation does the human animal permit in this function?

This is why computational approaches to language have much to offer. In a paper being revised for publication, Chandlee and Heinz (2014) summarize the benefits this way:

> The approach is to identify the formal properties of these... [infinite objects], which will be meaningful for either... [rule-based or constraint-based theories] since they speak directly to the nature of the object that ... [such theories] describe. In particular, they can shed light on the **kinds** of rules, constraints, and constraint rankings that ought to be admissible in SPE and OT. [emphasis added]

In other words, studying the extensions of constraints and transformations through the lens of a computationally-grounded encyclopedia of categories helps us better understand the nature of phonological component of grammar.

3 Representing constraints and transformations

Ultimately, phonological grammars represent the functions mentioned earlier. However unlike circles, phonologies are not described with single equations; instead phonological grammars contain multiple, interacting parts. In OT grammars those parts are constraints. In rule- based grammars those parts are rules. In this section, we put these intensions aside and examine the extensions of phonotactic constraints and the extensions of phonological transformations. Then in the next sections we examine the computational nature of those extensions.

3.1 Phonotactic knowledge and markedness constraints

Halle (1978) gives phonotactic knowledge as an example of knowledge that is learned but not taught. He provides an experiment demonstrating this knowledge, whose results are shown in Table 1. I have informally conducted this experiment myself on dozens, if not hundreds of young adults, who are native speakers of English. When presented visually with orthographic representations of the words in Table 1 (but ungrouped), student reliably and uniformly identify *thole, plast* and *flitch* as the English words.[3] Just as circles are fruitfully thought of as an infinite set of points, phonotactic knowledge can likewise be thought of as an infinite set of strings. All possible English words are in the set; all logically possible, impossible words are out of the set. This is but one concrete way to see that "language makes infinite use of finite means"; generative grammars allow us to distinguish among infinitely many logically possible forms. One question linguists address is What is the nature of this infinite set?

Markedness constraints in Optimality Theory (Prince and Smolensky, 1993, 2004) express phonotactic knowledge. Markedness constraints are said to prohibit marked structures so they distinguish well-formed structures from ill-formed ones. We will consider their extensions as

[3] A small minority of students suggest that *vlas* and *sram* might be English words but they agree they are less sure about these than the others. For more on gradient versus categorical distinctions in phonotactics, see Hayes and Wilson (2008) and Gorman (2013). In this chapter, we assume a categorical distinction for expositional purposes, but as discussed in Heinz (2010a) nothing really hinges on this.

possible English words	impossible English words
thole	ptak
plast	hlad
flitch	sram
	mgla
	vlas
	dnom
	rtut

Table 1: Words from Halle (1978).

follows: all surface forms with zero violations are in the set; all surface forms with nonzero violations are out of the set (cf. McCarthy (2003)). Therefore, the extensions of these constraints can be interpreted as all and only those strings which are well-formed according to the constraint; they are those structures which *do not contain* the marked structure as a sub-structure.

For example, consider the constraint *NC̥ (Pater, 2001), which states that nasals followed by voiceless consonants are marked sequences. The extension of this constraint can be conceived as the set of strings not containing marked structure, some of which are explicitly shown in (3).

(3) {a, b, aba, anda, anba, ...}.

In fact every logically possible string which does not contain this marked sub-structure is in the extension of the *NC̥ constraint. As will be discussed in greater detail in §5.1, substrings like NC̥ are sub-structures of strings.

Another example comes from syllable structure. It is widely held that codas are marked. Words with codas are said to violate the constraint NoCoda. Thus the well-formed structures picked out by this constraint are all and only those strings which do not contain codas as indicated by(4).

(4) {a, a.ba, pa.pa, ...}

The representations in (4) differ from those in (3) because they include a symbol for the syllable boundary. The available symbols, and the choice of representation more generally is an important issue, to which I will return at the end of this chapter.

3.2 Transformations

Extensions of transformations can also be described as infinite sets. In this case the elements of the set are *pairs*: the first element of the pair represents the *input* and the second element the *output*. Such extensions have been called *maps* by Bruce Tesar (2014) and others.

As an example, consider the SPE-style rule shown in (5), which epenthesizes [ɨ] between stridents.

(5) $\varnothing \rightarrow$ ɨ / [+strident] ___ [+strident]

The extension of this rule can interpreted as every pair of strings (i, o) such that if i is the input to the rule o would be the output. The extension of (5) is shown in (6).

(6) { (wɪʃz, wɪʃɨz), (d͡ʒʌd͡ʒz, d͡ʒʌd͡ʒɨz), (dɑgz, dɑgz), . . . }

Here is another example. Consider the rule in (7), which devoices word-final obstruents.

(7) [-sonorant] → [-voice] / ___ #

This rule describes the infinite set of pairs, indicated in (8).

(8) { (rat, rat), (sap, sap), (rad, rat), (sab, sap), (sag, sat), (flugenrat, flugenrat), (flugenrad, flugenrat), . . . }

OT descriptions of [ɨ]-epenthesis and word-final obstruent devoicing describe the *same* extensions. Baković (2013, chapter 4) shows how to translate any rule of the form A ⟶ B / C ___ D into a core ranking where a markedness constraint like *CAD outranks those faithfulness constraints violated by A→B. As he explains, this ranking "is assumed to be embedded within a constraint hierarchy" whose other constraints must also be ranked a certain way.

In other words, the ranking in (9) is the core ranking necessary to describe the pairs of strings in (6).

(9) *[+STRIDENT][+STRIDENT] >> DEP(ɨ)

Obviously, if there is a candidate which does not violate *[+STRIDENT][+STRIDENT] but violates some other faithfulness constraint F, then F must outrank DEP(ɨ). These and other constraints are part of the presumed constraint hierarchy to which Baković refers.

Similarly, in order to describe the set of pairs in (8), the core ranking in (10) must be embedded in the constraint hierarchy.

(10) *[+VOICE,-SONORANT]# >> ID(VOICE)

Both OT grammars and rule-based grammars can be used to describe the same sets of pairs. In cases where they define the same extension, they are like the polar and Cartesian systems which can describe the same circles with different equations, which are interpreted differently according to the system they inhabit.

Here we have focused on simple transformations—ones that in rule-based theories could be described by a single rule. But phonologies in the world's languages are more complex than that. There are multiple, interacting factors. Still, both OT grammars and rule-based grammars ultimately generate pairs of strings like the ones in (6) and (8). They do this in different ways, but they do it nonetheless. Furthermore, the way phonology is taught, practiced and studied—both rule-based and constraint-based theories—is exactly by examining fragments of grammars and building up to larger and larger analyses.[4] The approach here is

[4]In fact, both rule-based and OT grammars predict there to be complete phonological grammars which only instantiate the process of inter-strident epenthesis or word-final devoicing. The fact no known phonology only contains a map which would correspond to a single traditional phonological rule has never been taken as a problem for either rule-based theories or OT.

no different. Thus, the object of interest in both cases is these sets of pairs, which are the transformations from underlying to surface forms. As discussed earlier in (1), this is the basis for modern generative phonology. What is the nature of these maps?

4 Expressivity and restrictiveness

Before continuing, I would like to emphasize a common typological goal of every theory of phonology. A good theory must be both *sufficiently expressive* to accurately describe the actual phonologies in the world's languages and *maximally restrictive*. It is clear enough why expressive adequacy matters. To be clear, by "sufficiently expressive," I am referring to theories that, for every natural language phonology P, provide a descriptively adequate grammar for P. It may be less clear why maximal restrictiveness matters, but it does and no less so.

4.1 Why restrictiveness matters

There are three reasons why restrictiveness matters. First, it helps address the problem of how children quickly learn the phonology of their language (so it helps us reach an explanatorily adequate theory of phonology; cf. Chomsky (1965, chapter 1)).

Second, scientific hypotheses are stronger when they are more restrictive. The hypothesis that outlaws the most logically possible phonologies as humanly impossible can be said to be the strongest because it is the most readily falsifiable (Popper, 1959). For a restrictive theory, it is possible to identify logically possible patterns, which would serve as a counterexample to the theory, if in fact it were found in the phonology of some language.

Third, it is easy to find a sufficiently expressive theory of phonology which is not restrictive. The widely held Church-Turing thesis states that anything that can be calculated or computed can be computed by Turing machines (and equivalently Church's lambda calculus). If phonologists believe their theories and models of phonology are computable (no matter how complex or intricate the computations) then there is already a sufficiently expressive theory of phonology available. The problem with this theory is that it is unrestrictive because it says everything that is computable is possible. The mere existence of a phonology will never be sufficient grounds for dismissing the Church-Turing Theory of Phonology.

All of this is a way of saying that a theory not only needs to explain *what there is*, but also *what there is not*. Here is an analogy. A person is happy. Why? Well, one might say: they have all the things they want in life. That explanation is only partial. A truly happy person probably also *does not* have the things *they don't want* in life (like terminal cancer, poverty, etc.)

The larger point is that expressiveness needs to be balanced against restrictiveness. Failure to be sufficiently expressive does not automatically disqualify a theory. It is enough for theories to be "nearly sufficiently expressive" to be viable. I say this because sometimes theories, especially when newly posited, are not sufficiently expressive (though they are very restrictive). For example, the Copernican theory of the solar system was originally not sufficiently expressive to predict the retrograde motion of the planets, at least as compared to the best Ptolemaic models at the time. Still the predictions were close. The Copernican theory was more restrictive however (since fewer types of retrograde motions were possible with the sun as the center of the system.) Closer to home, classic OT was not abandoned simply because

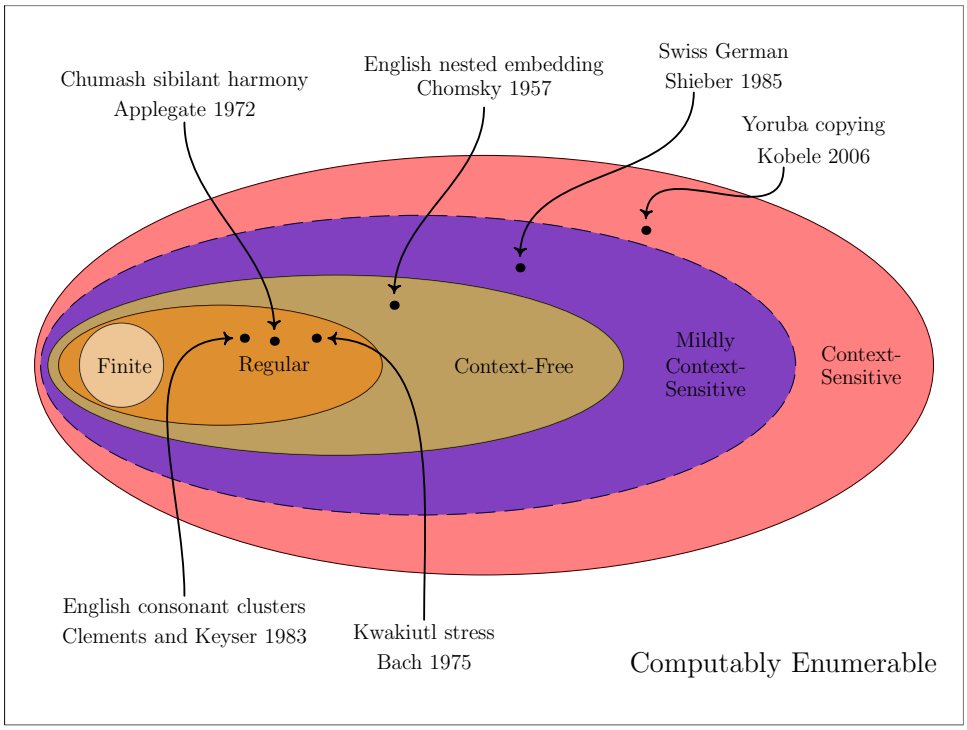


Figure 1: Natural language patterns in the Chomsky hierarchy.

of its inherent inability to represent opaque maps. Instead, subsequent research focused on trying to modify the theory to make it more expressive.

There are degrees of expressivity and degrees of restrictiveness. Once we recognize the extensions of the constraints and transformations posited in phonological theories are infinite sets and functions, then we will see that the theory of computation naturally provides an encyclopedia of categories which measures these degrees of expressivity and restrictiveness. Furthermore, in this regard, the theory of computation is without peer.

4.2 The Chomsky Hierarchy

In this section, I will provide an overview of why the theory of computation provides a valuable way to examine the expressivity and restrictiveness of linguistic theories. Figure 1 shows the Chomsky Hierarchy which classifies stringsets according to the kind of grammars that generates them. Points in the space represent stringsets. The larger regions properly include the smaller ones so for instance all regular stringsets are context-free but not vice versa. As shown in the figure, linguistic generalizations (modeled as stringsets) have been argued to belong to certain regions and not others within the hierarchy.

(The issue of representation—whether we want to model linguistic forms with string structures, tree structures, autosegmental structures, or other kinds of graph structures is taken up in the discussion section 8. There, I will argue that even if string structures are left behind, computational theory still provides an unmatched encyclopedia of categories for these other structures, analogous to the ones I discuss here for strings.)

At the top of the hierarchy is the "computably enumerable" region which includes every-

thing. These are essentially the stringsets whose elements can are computable.[5] This is the most expressive, but least restrictive class.

At the bottom of the hierarchy are the "finite stringsets." These stringsets are of finite cardinality. Unlike infinite sets, which require a generative grammar to generate or recognize them, elements of finite sets can be listed. In introduction to linguistics courses, we learn that linguistic generalizations cannot be modeled with finite sets because there is no principled upper bound on the length of possible words or sentences. The finite languages are the most restrictive, but least expressive class. In between the computably enumerable and finite classes are the regular, context-free and context-sensitive regions.

An important aspect of the hierarchy is that several regions have independently motivated, equivalent descriptions. Regular stringsets for instance can be defined with monadic second order logical formulae, finite-state acceptors, or regular expressions. Computer scientists Engelfreit and Hoogeboom explain: "It is always a pleasant surprise when two formalisms, introduced with different motivations, turn out to be equally powerful, as this indicates that the underlying concept is a natural one. Additionally, this means that notions and tools from one formalism can be made use of within the other, leading to a better understanding of the formalisms under consideration" (Engelfriet and Hoogeboom, 2001, p. 216). At a high level of abstraction, the different characterizations can be thought of as different views on the same underlying object roughly in the same way different equations in different coordinate systems can describe the same circle. The more views we have, the better we can understand what it is we are looking at.

Also, each region X describes a linguistic hypothesis: Linguistic generalizations must belong to X. Early work in generative grammar was interested in establishing evidence for or against such hypotheses in order to establish upper bounds on the nature of linguistic generalizations. The weakest scientific hypothesis is that they are computably enumerable, which is what I called the Church-Turing Theory of Phonology. As X moves down the hierarchy, the hypotheses become stronger, so the claim that the weak generative capacity of human syntax is a regular stringset is a strong scientific hypothesis. However, it is generally considered to be false (Chomsky, 1956; Shieber, 1985).

4.3 Phonology is regular

The Chomsky Hierarchy is the most well-known hierarchy in formal language theory, but it is not the only one. In fact there are several other hierarchies, some which only became well understood in recent decades, and others which are still being formed. Also, the Chomsky Hierarchy in Figure 1 classifies stringsets, but hierarchies exist (and are being developed) for sets of pairs of strings (relations/maps/functions) as well. It is important to distinguish hierarchies for one kind of set (e.g. stringsets) from another (e.g. sets of pairs of strings).

An important region in a hierarchy for relations (sets of pairs of strings) is also called Regular. It is called this because it shares much in common with the regular class of stringsets. For instance, one way to define the regular class of stringsets is with non-deterministic finite-state acceptors and one way to define the regular class of string-to-string maps is with non-deterministic finite-state *transducers*. Readers are referred to other texts for more information

[5] More formally, it is decidable whether or a not any particular string belongs to the set. Interestingly, most logically possible sets of strings are *not* computably enumerable (Turing, 1937).

about finite-state grammars (Sipser, 1997; Beesley and Kartunnen, 2003; Roark and Sproat, 2007; Jurafsky and Martin, 2008; Hulden, 2009).

The primary result in computational phonology to date is that the transformations from underlying to surface forms—these phonological maps—are in fact *regular*. The argument goes something like this: Optional, left-to-right, right-to-left, and simultaneous application of SPE-style rules A→B/C__D (where A,B,C,D are regular stringsets) *describe regular relations*, provided the rule cannot reapply to the locus of its structural change (Johnson, 1972; Koskenniemi, 1983; Kaplan and Kay, 1994). Rule ordering is functional composition (finite-state transducer composition). Regular relations are closed under composition (so the composition of two regular relations is also a regular relation). Rule-based grammars (finitely many ordered rewrite rules of the above type) can describe virtually all attested phonological patterns. This does not mean these grammars do so elegantly, that the rules correspond to psychologically real constructs, or that they have any other desirable trait. It just means that the input/output map is describable with a such a grammar.

The above argument constitutes significant evidence for the following statement.

(11) (Regular Hypothesis) Phonological maps are regular relations.

If this is true, then it is true *regardless* of whether they are described with SPE, OT, or other grammar formalisms! Here are some other ways of saying the same thing.

- There are no non-regular phonological maps.
- A *universal* property of phonological maps is that they are regular.

Again, the fact that every rule-based grammar describes a regular relation, in addition to the fact that there is no counterexample to the hypothesis that phonological maps are regular, is strong evidence that the hypothesis in (11) is correct.

One consequence of this result is that finite-state grammars become a lingua franca for different phonological theories describing some aspect of the phonology of a language. Hence in addition to the work mentioned above which translates rule-based grammars into finite-state machines, there exists much work which shows how to translate OT grammars into finite-state machines (Frank and Satta, 1998; Karttunen, 1998; Gerdemann and van Noord, 2000; Riggle, 2004). Thus, for attested phonological patterns—just as with circles—there at several ways we can describe them. Those stringsets and maps can be described with rule-based grammars, OT grammars, finite-state machines, and other tools (e.g. logical formulae).

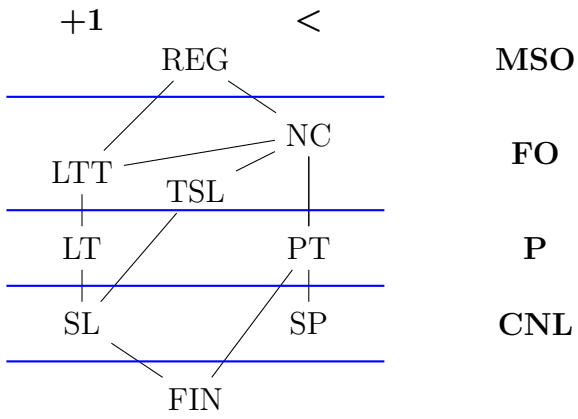
Another consequence of (11) follows from a theorem by Scott and Rabin (1959). This theorem establishes that the domain and image of regular relations are regular sets of strings. This means the set of possible underlying representations and the set of possible surface representations are also regular. In other words, phonotactic knowledge and markedness constraints describe regular stringsets. Or equivalently, every stringset defined by a markedness constraint has the property of "being regular."

4.4 The Subregular Hypothesis

"Being regular" is therefore plausibly a universal property of phonological patterns (both stringsets and maps). Furthermore, it is restrictive: there are many logically possible, *non-regular* patterns.

However, while "being regular" may be a necessary property, it is not restrictive enough. There are many logically possible, regular patterns that are still bizarre from a phonological perspective. We will encounter some of these strange creatures shortly.

There are many interesting *subregular* classes of stringsets, as shown in Figure 2. Figure 2 shows a "close-up" view of the regular region shown in Figure 1. This will constitute the "encyclopedia of categories" and it will be explored in more detail in section 5.2, though an overview will be given here.



Names of the classes of stringsets			
REG	Regular	FIN	Finite
LTT	Locally Threshold Testable	NC	Non-Counting
LT	Locally Testable	PT	Piecewise Testable
SL	Strictly Local	SP	Strictly Piecewise
TSL	Tier-based Strictly Local		
Names of the Parameters			
+1	Successor	MSO	Monadic Second Order
<	Precedence	FO	First Order
		P	Propositional
		CNL	Conjunction of Negative Literals

Figure 2: Subregular hierarchies of stringsets.

The subregular hierarchies are bounded by the Regular region at the top and Finite region at the bottom. A region higher up in the diagram which is connected by a line to a region lower down in the diagram indicates the lower region is a subset of the higher region. So every generalization in the lower region is expressible in the higher one, but not vice versa.

There are two main branches in these hierarchies, the successor branch (+1) and the precedence branch (<) (for now the Tier-Based Strictly Local can be ignored). The successor branch is also known as the Local branch, and the precedence branch is also known as the Piecewise branch. Along each branch, the regions are defined in logical terms. The Monadic Second Order regions are the most expressive and least restrictive. This is followed by (in order

of decreasing expressivity/increasing restrictiveness) the First Order, Propositional regions and then the least expressive and most restrictive regions, the Conjunction of Negative Literals. While the hierarchy is presented here in logical terms these regions can also be defined in other ways and have multiple characterizations just like the regions in the Chomsky Hierarchy.

The Subregular Hypothesis refers to the idea that phonological patterns belong to small, well-defined regions of regular stringsets and maps. Thus the term "Subregular Hypothesis" on its own does not say much because it itself does not say *which* subregular regions are at stake. To anticipate the remainder of this chapter, we distinguish between a strong and weak subregular hypothesis for constraints (a similar hypothesis will be put forward for maps). The "strong" subregular hypothesis is that phonological markedness constraints are Strictly Local and Strictly Piecewise (at the bottom of the hierarchy). The "weak" subregular hypothesis is that they are Tier-Based Strictly Local, which is a particular generalization of the Strictly Local class (inspired by phonological tiers). Both of these hypotheses are discussed explicitly in sections 5.2.1 and 5.2.3.

5 Constraints

This section is devoted to markedness constraints. The primary purpose is to describe the Subregular Hierarchies in Figure 2 on page 14, which constitutes the encyclopedia of categories, that I am arguing is important for understanding the nature of markedness constraints in phonology. To help motivate the discussion, and help make it more accessible, I will begin by discussing part of an encyclopedia of types (the actual constraints found in natural language).

5.1 The encyclopedia of types: stringsets

In this section, I present some constraints known to be attested in the world's languages. These will be contrasted with constraints that are unattested. This is not intended to be an exhaustive or comprehensive encyclopedia of types. Only four types of markedness constraints are presented. This is intended to be sufficient to motivate the encyclopedia of categories, presented afterwards.

5.1.1 Four types of constraints

The first type of markedness constraint we encountered penalizes certain contiguous sequences of sounds (substrings). The impossible English words in Halle's (1978) example all begin with illicit consonant clusters. We saw that *N̥Ç also penalizes substrings. This constitutes one kind of markedness constraint.

There are other kinds of constraints employed by phonologists which identify structures other than substrings as marked. For instance, if we were to ask native speakers of Samala (Applegate, 1972, 2007) about the words in Table 2, they would reliably and uniformly distinguish them as shown in the table. How do Samala speakers know which of these words belong to different columns?[6] Well, it appears that Samala speakers know that words can't contain both [+anterior] sounds like [s] and [−anterior] sounds like [ʃ].[7] Applegate (1972)

[6] By the way, *ʃtoyonowonowaʃ* means 'it stood upright' (Applegate 1972).

[7] The relevant feature could also be [distributed].

possible Samala words	impossible Samala words
ʃtojonowonowaʃ	stojonowonowaʃ
stojonowonowas	ʃtojonowonowas
pisotonosikiwat	pisotonoʃikiwat
nasipisotonosikiwa	naʃipisotonoʃikiwa

Table 2: Phonotactic knowledge in Samala

modeled this knowledge as the result of a productive, regressive sibilant harmony process. In Optimality Theory, this knowledge would be a consequence of a high ranking constraint of the form *[+strident,α anterior]...[+strident,$-\alpha$ anterior] (Hansson, 2001; Rose and Walker, 2004; Heinz, 2010a). This constitutes a second type of attested markedness constraint in the world's languages.

Another logically possible type of constraint is illustrated with by speakers of a language I will call "Language X" (its true identity will be revealed momentarily), shown in Table 3. What constraint are the speakers of this language utilizing to reliably distinguish the logically possible words shown there? In this case, we observe speakers of Language X reject words that

possible words of Language X	impossible words of Language X
ʃotkoʃ	sotkoʃ
ʃoʃkoʃ	ʃotkos
ʃosokoʃ	ʃoʃkos
soʃokos	soskoʃ
sokosos	
pitkol	
pisol	
piʃol	

Table 3: Phonotactic knowledge in Language X

begin and end in sibilants that disagree in the feature [anterior]. Unlike the Samala example, sibilants interior to the word may disagree with edge-bound sibilants as evidenced by possible words like [soʃokos]. But if there are sibilants at word edges, they must agree in order for the word to be a possible word of Language X. This type of constraint is distinct type of constraint from the previous two types mentioned.

Next we consider Language Y. Table 4 shows how speakers of this language discriminate logically possible words. How do they do it? The two columns in Table 4 are distinguished as follows. Possible words have an *even number* of sibilant sounds, but impossible words have an *odd number* of sibilant sounds. So speakers of Language Y are sensitive to the even/odd parity of the number of sibilant sounds. This constitutes a fourth type of constraint, distinct from the ones mentioned earlier.

So far we have considered four logically possible kinds of constraints. What is the actual typology, the encyclopedia of types?

The actual typology of course looks like this. Attested phonotactic patterns include those

possible words of Language Y	impossible words of Language Y
ʃotkoʃ	ʃoʃkoʃ
sotkoʃ	ʃoskoʃ
ʃotkos	soʃkos
pitkol	ʃoʃkos
soʃkostoʃ	soskoʃ
	soksos
	piskol
	piʃkol

Table 4: Phonotactic knowledge in Language Y

which forbid substrings of words (such as *NC̥). They also include ones where words don't contain both sounds like ʃ and *s* (as in Samala). However, the logically possible phonotactic patterns represented by languages X and Y are unattested. There are no known phonotactic patterns where the last sound in a word depends in some fashion on its first sound (as in Language X). And there are no known phonotactic patterns where the right generalization is that words must contain an even number of members of a particular natural class (as in Language Y).

5.1.2 Explaining the typology

We would like to have an explanation for this fact. We would like our theory of markedness to explain why constraints like those found in English and Samala are possible, but the ones found in Language X and Language Y are not.

So what's the explanation? In Optimality Theory, constraints like *#mgl and *[+strident,α anterior]...[+strident,$-\alpha$ anterior] structures would be part of CON. But constraints like *ODD-Sibilants or *#[+strident,α anterior]...[+strident,$-\alpha$ anterior]# would not be. The explanation in OT largely comes down to constraints that are present in CON and those that are absent in CON. (Whether complex markedness constraints can be derived via constraint interaction is a matter I take up later in section 7.).

This is not controversial. The basic syllable typology is derived in OT by including the constraints NoCoda,Onset and excluding the constraints NoOnset,Coda. If the constraints NoOnset,Coda were included in CON, then it would not be possible to derive a typology where onsets may be required (but are never forbidden) and codas may be forbidden (but are never required).

In phonetically-based phonology (Hayes *et al.*, 2004), the explanation would be that there are perceptual and/or articulatory reasons for constraints like *#mgl and *[+strident,α anterior]...[+strident,$-\alpha$ anterior]. But there would be no such reasons for constraints like *ODD-Sibilants or #[+strident,α anterior]...[+strident,$-\alpha$ anterior]#. More generally, this research program hypothesizes that constraints in CON are based on phonetic principles, and that certain rankings of these constraints are also fixed according to these principles. This is a significant improvement over stipulating certain constraints as belonging to CON to the exclusion of others.

However, we should carefully examine the proposed perceptual and/or articulatory rea-

sons. Consider the case of the pattern in Language X. Following Lai (2012, 2014), let us now refer to this pattern as First/Last Harmony. We know long-distance assimilation is well-attested (Hansson, 2001; Rose and Walker, 2004) and arguments have been made for its perceptual basis (Gallagher, 2010). We also know word edges in phonology are privileged positions (Fougeron and Keating, 1997; Beckman, 1998; Endress *et al.*, 2009). So what theory of perception or articulation prevents there from being harmony only in privileged positions?[8] This is a case where the phonetic principles seem to overpredict the attested typology.

We may also wonder to what extent memory requirements could explain the difference between the attested pattern in Samala and First/Last Harmony. In fact, however, it comes down to the pattern type, or template. This is because both types can be described simply by marking which pairs of sounds are permitted or forbidden in a given template as shown in Figure 3. The 2x2 cells for are identical—it is only the templates that differ.

	[s]	[ʃ]
[s]	✓	✗
[ʃ]	✗	✓

$[\ldots \_\_ \ldots \_\_ \ldots]$

	[s]	[ʃ]
[s]	✓	✗
[ʃ]	✗	✓

$[\#\_\_ \ldots \_\_ \#]$

Figure 3: Pattern templates for Sibilant Harmony (left) and First/Last Harmony (right).

As for the pattern in Language Y, it is plausible that perception or articulation should be able to explain the absence of even/odd parity constraints (or more generally constraints which count mod n) in phonology, but I haven't seen any explicit connection. Whatever the explanation may be, it *should* connect to the computational properties discussed here. More generally, if phonology is truly reducible *entirely* to phonetic principles then there ought to be research showing how the computational laws being posited in this chapter can be clearly derived from such phonetic principles.

The computational explanation offered in this chapter is simply this. The extensions of constraints on substrings (like *NC̥) and constraints on subsequences (like [+strident, α anterior]...[+strident, $-\alpha$ anterior] in Samala) are Strictly Local and Strictly Piecewise stringsets respectively. With the exception of the finite languages, these are the most restrictive, least expressive regions in the Subregular Hierarchies shown Figure Table 2 on page 14. On the other hand, First/Last Harmony and *ODD-Sibilants belong to the Locally Testable and Regular regions, respectively. In other words, the widely-attested constraints are the formally simple ones, where the measure of complexity is determined according to these hierarchies.

5.2 The encyclopedia of categories

In the last section, extensions of phonological constraints and transformations were introduced and it was argued that conceiving them as stringsets (formal languages), and string-to-string maps was reasonable and potentially insightful. In this section, we explain what the theory

[8] Alan Yu also points out that there is a perceptually-motivated diachronic path to arriving at this language (pace Ohala (1981) and Blevins (2004)). A language like Samala would be the precursor language to one with First/Last Harmony. Since interior sibilants in the precursor language are not perceived as accurately as ones at word edges, some of them may change over time to disagreeing sibilants. This would result in a language whose words obey First/Last Harmony.

of computation has to say about this cast of characters. To do so requires some technical details. I will try to keep them light and only provide a sketch. Readers interested in the all the details are referred to Rogers *et al.* (2013), which also provides cognitive interpretations of the different regions.

We are going to explore *language-theoretic* and *logical* descriptions of stringsets and string-to-string maps from a generative perspective. The word "language" in "language-theoretic" refers to a formal language, what we have been calling a stringset. We may as well call it "stringset-theoretic". The idea behind language-theoretic descriptions is that they are completely independent of any grammatical description. In other words, these descriptions are statements that are simply true of the stringset itself (and not the grammar that generates it). They can therefore be thought of as essential properties of the stringsets. Examples will be provided shortly.

Logical descriptions are not agrammatical. Logical formulae are grammars in the sense that they generate stringsets as extensions. However, they are useful here because the expressive power of different logics is well understood. In the encyclopedia of categories—the Subregular Hierarchies—presented in Figure 2 on page 14, there were four types of logic. As mentioned earlier, in order of strictly increasing expressive power, they are: conjunctions of negative literals (CNL), propositional logic (P), first order logic (FO), and monadic second order logic (MSO). The type of logic forms one parameter that is used to define the regions in the Subregular Hierarchies (Rogers and Pullum, 2011).

There is one other parameter used to define the regions. This parameter specifies the kind of structures used to model strings. The parameter specifies the kind of relation used to handle the order of elements in the string. The relations that have been studied are *successor* ($+1$) and *precedence* ($<$).

If strings are modeled so that the order of elements is handled by the successor relation, then *substrings* will be sub-structures of strings. On the other hand, if strings are modeled so that the order of elements is handled by the precedence relation, then *subsequences* will be sub-structures of strings. As an example, with the successor relation, **bcc** is a sub-structure of *a***bcc***ab*, but with the precedence relation, **aab** is a sub-structure of **a***bcc***ab**.

We will now examine the consequences of the ordering relation in terms of the logical power.

5.2.1 Conjunctions of Negative Literals

Stating constraints as a "negative literal" is logical talk for what phonologists simply call marked structure. When conjunctions of such constraints are considered, we define stringsets which do not contain any of the marked structures. The order parameter (successor or precedence) tell us whether to interpret literals (the string structures) as substrings or subsequences.

Let us begin with successor, so the literals are interpreted as substrings. The negative literal $\neg aa$ is thus interpreted to mean the substring aa is a marked structure. So any string containing this marked structure violates the constraint and is not in the extension of the constraint.

Here is an example. I will use φ to stand for logical formulae, and $\mathbb{L}(\varphi)$ to stand for the stringset extension of φ. Following Rogers *et al.* (2013), I also will use $\rtimes$ and $\ltimes$ for the left and right word boundaries. The the formula below can be read as "Strings which do not begin with a b, do not contain aa as a substring, do not contain bb as a substring, and do not end

with a a, are well-formed."

$$\varphi = (\neg \rtimes b) \wedge (\neg aa) \wedge (\neg bb) \wedge (\neg a \ltimes)$$

The extension $\mathbb{L}(\varphi)$ is easy to write since conjunction is interpreted as set intersection. A word about notation: Σ is a finite set of symbols (the alphabet); Σ^* means all logically possible strings one can write with this slphabet; and $\overline{S}$ means the complement of stringset S with respect to Σ^*. Thus, a term by term translation of φ above into its extension is shown below.

$$\mathbb{L}(\varphi) = \overline{b\Sigma^*} \cap \overline{\Sigma^* aa \Sigma^*} \cap \overline{\Sigma^* bb \Sigma^*} \cap \overline{\Sigma^* a}$$

It is not difficult to see that this is the same as the infinite set $\{ab, abab, ababab, \ldots\}$.

So now we can provide one definition of the Strictly Local stringsets. A Strictly k-Local (SL_k) stringset is one which can be defined as the conjunction of negative literals, where the literals are interpreted as substrings, and whose longest forbidden literal (substring) is of length k. The Strictly Local stringsets are those that are SL_k for some k.

If the order relation is precedence, then the literals are interpreted as subsequences. The negative literal $\neg aa$ is thus interpreted to mean the subsequence aa is a marked structure. So any string containing this marked structure violates the constraint and is not in the extension of the constraint.

Here is an example. The formula below can be read as "Strings which do not contain an a followed by an a nor a b followed by a c are well-formed." So here the literals aa and bc are interpreted as subsequences, and not as substrings.

$$\varphi = (\neg aa) \wedge (\neg bc)$$

A term by term translation of φ above into its extension is shown below.

$$\mathbb{L}(\varphi) = \overline{\Sigma^* a \Sigma^* a \Sigma^*} \cap \overline{\Sigma^* b \Sigma^* c \Sigma^*}$$

Strictly Piecewise stringsets are defined analogously to Strictly Local stringsets. A Strictly k-Piecewise (SL_k) stringset is one which can be defined as the conjunction of negative literals, where the literals are interpreted as subsequences, and whose longest forbidden literal (subsequence) is of length k. The Strictly Piecewise stringsets are those that are SP_k for some k.

That many attested markedness constraints are SL and SP stringsets is not in dispute. Clearly, constraints like *NÇ are SL and constraints like [+strident, α anterior]...[+strident, $-\alpha$ anterior] are SP. The strong subregular hypothesis states that *all* markedness constraints are either SL or SP (Heinz, 2010a).

(12) (Strong Subregular Hypothesis) Markedness constraints are SL or SP.

The one notable outstanding case Heinz (2010a) discusses is the set of surface forms derived from long-distance dissimilation. These appear to be properly Non-Counting (Heinz *et al.*, 2011) and are discussed further below.

Whether constraints like Onset are SL or not depends on the choice of representation. If syllable boundaries are included in string representations, which is a common practice, then constraints like Onset are SL since they can be represented this way: $(\neg\ .\mathrm{V})$. The importance of representations will be further discussed in section 8.

I would like to conclude the discussion of the "Strict" classes by providing their language-theoretic characterizations. This characterization for SL stringsets is provided in (13), which Rogers and Pullum (2011) name Suffix Substitution Closure.

(13) (Suffix Substitution Closure) A stringset L is SL if there is a k such that for all strings u_1, v_1, u_2, v_2, x with the length x equal to $k-1$, it is the case that if u_1xv_1 and u_2xv_2 belong to L then u_1xv_2 belongs to L as well.

Suffix Substitution Closure is ultimately a Markovian principle: the well-formedness of the next symbol in the string depends only on the previous $k-1$ symbols (given as x above). So if both v_1 and v_2 can follow x, what comes before x (u_1 and u_2) does not matter. This Markovian notion is important in generalizing SL stringsets to SL functions discussed in section 6.

On the other hand, subsequence closure characterizes the SP class (Rogers *et al.*, 2010).

(14) (Subsequence Closure) A stringset L is SP if for all u belonging to L, every subsequence of u also belongs to L.

What is remarkable about the language-theoretic characterizations above is not only that stringsets which have these properties are exactly the ones that can be defined as the conjunction of negative literals, but also that these characterizations do not mention any sort of grammar at all. In this way, they are more like definitions of circles which do not refer to either the Cartesian or polar coordinate system. They are characterizations which speak directly to the nature of the stringsets without getting bogged down in any particular grammatical formalism.

Another remarkable fact about these characterizations is that they immediately suggest *inference* procedures. If one is observing words from a language L *that is a priori known* to be SL_k and one observes u_1xv_2, u_2xv_2, one can immediately deduce that u_1xv_2 also belongs to L. Similarly, if one is observing words from a language L *that is a priori known* to be SP, and one observes the word u, one can immediately determine all subsequences of u also belong to L.

These facts lay at the basis of learning algorithms developed for the SL_k and SP_k classes. Garcia *et al.* (1990) first proved that the SL_k stringsets are identifiable in the limit from positive data. I have argued elsewhere that this is a rigorous and insightful learning paradigm (Heinz, 2014a), and I will not review the arguments here. Heinz (2010a) shows how a similar algorithm provably identifies SP_k languages in the limit from positive data, and Heinz *et al.* (2011) generalizes these ideas to a family of learning algorithms, a result which was generalized even further by Heinz *et al.* (2012).

If the strong subregular hypothesis is correct, these learning results provide a deep explanation of it. Constraints on phonological well-formedness are SL and SP because people learn phonology in the way suggested by these algorithms. More specifically, people generalize in accordance to inference procedures suggested by the closure properties in (13) and (14). But it is the inference procedures themselves that are basic which structure the SL_k and SP_k classes; the inference procedures are not auxiliaries to the classes.

5.2.2 Propositional Logic

Next we move up one level to the next kind of logic: propositional. Unlike the conjunction of negative literals, where all formulae had the form $(\neg\ell_1) \wedge (\neg\ell_2) \wedge \ldots \wedge (\neg\ell_n)$ for n literals (ℓ_i),

propositional logic allows any well-formed propositional formulae to generate a stringset. Not only is any combination/ordering of negation and conjunction now permitted, but disjunction (*vee*) is also allowed. As a consequence, mainly familiar propositional connectives are also allowed, such as implication ($\rightarrow$) and the biconditional ($\leftrightarrow$). Propositional logic is therefore more expressive (and less restrictive) than the conjunction of negative literals.

For example, the following formula is a well-defined formula in propositional logic.

$$\varphi = b \vee (ab \Rightarrow bc)$$

If these literals are interpreted with respect to the successor model of strings, then this formula translates to the following English: "Words are well-formed if they contain the substring b or if it is the case that if they contain the substring ab they also contain the substring bc." Below I provide the extension of φ under the successor interpretation of the literals.

$$\mathbb{L}(\varphi) = \Sigma^* b \Sigma^* \cup (\Sigma^* ab \Sigma^* ac \Sigma^* \cup \Sigma^* ac \Sigma^* ab \Sigma^*)$$

If these literals are interpreted with respect to the precedence model of strings, then this formulae translates to the following English: "Words are well-formed if they contain the subsequence b or if it is the case that if they contain the subsequence ab they also contain the subsequence bc." Here is the extension of φ under the precedence interpretation of the literals.

$$\mathbb{L}(\varphi) = \Sigma^* b \Sigma^* \cup (\Sigma^* a \Sigma^* b \Sigma^* c \Sigma^* \cup \Sigma^* a \Sigma^* c \Sigma^* b \Sigma^*)$$

I submit that both of these logically possible constraints seem more odd from a phonological perspective than the SL or SP constraints. At first glance, it seems strange to have a markedness constraint which requires that if one sub-structure is present another one must be present as well.

This is perhaps the most notable difference between the kinds of constraints permitted using propositional logic. Such constraints can *require* sub-structures to be present in well-formed words (Rogers and Pullum, 2011). The interpretation of the simple formulae $\varphi = b$ is that well-formed words must contain the sub-structure b. Such examples exist in the phonological literature. for instance, it is true that the constraint Onset has this flavor. As we have mentioned with Onset, however, the choice of representation matters: this can be construed as SL provided syllable boundaries are introduced as symbols in strings. Another constraint like this is what Hyman (2009) calls Obligatoriness, the requirement that all well-formed words bear an accent (or stress). Unlike Onset, there is no straightforward representational "fix" for this constraint. I return to this issue in section 5.3.

Now we can provide one definition of the Locally Testable stringsets. A Locally k-Testable (LT_k) stringset is one which can be defined with a formula in propositional logic, where the literals are interpreted as substrings, and whose longest literal (substring) is of length k. The Locally Testable stringsets are those that are LT_k for some k.

Similarly, a definition of the Piecewise Testable stringsets can be given. A Piecewise k-Testable (PT_k) stringset is one which can be defined with a formulae in propositional logic, where the literals are interpreted as subsequences, and whose longest literal (subsequence) is of length k. The Piecewise Testable stringsets are those that are PT_k for some k.

There are language-theoretic characterizations of these classes too. This characterization is given in (15) for the Locally Testable class.

(15) (Substring Equivalence) A stringset L is LT if there is a k such that for all strings u and v, if u and v have the same set of substrings of length k then either both u and v belong to L or both u and v do not belong to L.

In other words, Substring Equivalence means that membership in a LT stringset L only depends on the set of substrings of some length k. If two distinct strings have the same substrings up to some length k then no LT_k stringset is able to distinguish them.

A similar characterization is given in (16) for the Piecewise Testable class.

(16) (Subsequence Equivalence) A stringset L is PT if there is a k such that for all strings u and v, if u and v have the same set of subsequences of length k then either both u and v belong to L or both u and v do not belong to L.

Subsequence Equivalence means that membership in a PT stringset L only depends on the set of subsequences up to some length k. If two distinct strings have the same subsequences of some length k then no PT_k stringset is able to distinguish them.

Like the characterizations for the "Strict" classes, these characterizations naturally suggest inference procedures. If one is observing words from a language L *that is a priori known* to be LT_k and one observes u, one can immediately deduce that all words with the exactly the same substrings up to length k also belong to L. Similarly, if one is observing words from a language L *that is a priori known* to be PT_k, and one observes the word u, one can immediately determine all words with the exactly the same subsequences up to length k also belong to L.

That the PT_k and LT_k stringsets are identifiable in the limit from positive data was established by García and Ruiz (2004). Heinz *et al.* (2011) and Heinz *et al.* (2012) show there are learning algorithms for these classes which have much in common with the ones for the Strict classes. An interesting difference, however, between these algorithms and the ones for the Strict classes has to do with time-complexity: there is a clear computational sense in which learning these more expressive classes takes significantly longer than learning the Strict classes.

5.2.3 First Order Logic

The next rung up the logical hierarchy brings us to First Order (FO) Logic. The main differences between first order logic and propositional logic is that literals disappear and variables appear. It is not necessary in this chapter to provide the technical details regarding FO models of strings. For this, readers are referred to Rogers *et al.* (2013).

There are only three important items readers need to to understand. First, FO logic is strictly more powerful logic than Propositional logic. Second, as usual, whether the ordering relation is given as the successor relation or the precedence relation will determine the kinds of stringsets expressible with FO formulae. FO logic with the successor relation yields the class called the Locally Threshold Testable (LTT) class, and FO logic with the precedence relation yields the class called Non-Counting (NC). Third, successor is FO-definable from precedence, but not vice versa so the Non-Counting class properly includes the LTT class.

I will go straight to the language-theoretic properties. If the ordering relation is the successor, then the class of stringsets that is FO-definable is called the Locally Threshold Testable (LTT) class, and it properly includes the LT class.

One important difference between the FO-definable classes and the Propositional-definable classes is that the FO-definable classes are able to distinguish the presence of otherwise identical

sub-structures. In this way, FO-definable classes can count the number of sub-structures up to some threshold. On the other hand, the Propositional classes can only detect the presence or absence of sub-structures. So for a given sub-structure, Propositional logic can distinguish zero of them from one of them. FO logic, however, can detect up to some number n of sub-structures. So a limited ability to count is present at the FO-level. There is always some finite number n after which the number of sub-structures cannot be distinguished. FO-definable classes are not sufficiently expressive to be able to count indefinitely. Thus the difference between LTT and LT is that in the LTT class, the *number* of substrings can be counted, but only up to some threshold t (Thomas, 1997).

(17) (Substring Threshold Equivalence) A stringset L is LTT if there is a k and a t such that for all strings u and v, if u and v have the same number, up to some threshold t, of substrings of length k then either both u and v belong to L or both u and v do not belong to L.

By now the reader may expect that language-theoretic characterization of FO-definable classes with the precedence relation is similar except that the number of subsequences up to some threshold is distinguishable. It is, however, in fact much simpler than that.

(18) (Non-Counting) A stringset L is NC if there is a k such that for all strings u, x, v if ux^kv belongs to L then so does $ux^{k+1}v$.

The reason for this is that the Non-Counting class can do much more than count subsequences. This is partly because the successor ordering relation is FO-definable from precedence, but not vice versa.[9] Consequently, every stringset in the LTT region is also in the Non-Counting region, but not vice versa. NC is strictly more expressive than LTT. Thus, Figure 2 on page 14 shows that the the NC class properly includes the LTT class. McNaughton and Papert (1971) comprehensively establish several other important characterizations of the Non-Counting class.

Are there markedness constraints that count up to some threshold? An example of such a constraint would be something like *3NC̥ where words with zero, one or two NC̥ substrings are considered well-formed, but words with three or more are ill-formed. Needless to say, such constraints do not seem like the kinds of constraints found in natural language.

On the other hand, there are constraints in natural language that have been argued to be properly Non-Counting. These are the stringsets that are definable from long-distance dissimilation (Heinz *et al.*, 2011). Heinz *et al.* (2011) also show that such constraints belong to subclasses of the Non-Counting region they call Tier-based Strictly Local (TSL). These stringsets are defined with the common notion of phonological tier (Goldsmith, 1976). Like the Strictly Local class, TSL stringsets can be defined with formulae that are conjunctions of negative literals, interpreted under the successor relation after non-tier elements are ignored. Thus the kind of long-distance behavior is limited in some kind of way. TSL stringsets are not as well understood as the other classes (there are not multiple characterizations), but Heinz *et al.* (2011) argue that every markedness constraint in natural language is describable

[9] For those familiar with the FO formulae, here is the definition where $x \rhd y$ means y is a successor of x, and $x < y$ means x precedes y.

$$x \rhd y \stackrel{\text{def}}{=} x < y \wedge \neg(\exists z)[x < z \wedge z < y]$$

with TSL constraints. Of course an important variable here is the tiers.

I will refer to the hypothesis that all markedness constraints are TSL as the weak subregular hypothesis.

(19) (Weak Subregular Hypothesis) Markedness constraints are TSL.

Whether the evidence favors the strong or weak subregular hypothesis will be addressed in section 7.

5.2.4 Monadic Second Order Logic

The next rung up the logical hierarchy and the highest to which we attend is Monadic Second Order (MSO) Logic. The difference between first order and monadic second order logic is that variables over *sets* of elements in the domain are allowed in addition to the variables which vary over individual elements (which FO logic allows). There are several interesting consequences of adding such variables, which I will now review.

First, the two branches in the subregular hierarchies merge at this point because precedence is MSO-definable from successor.[10] So the stringsets that are MSO-definable with successor are exactly the stringsets that are MSO-definable with precedence.

Second, this class of stringsets corresponds exactly to the class of stringsets definable with finite-state acceptors, i.e. the regular class of stringsets (Büchi, 1960).

Third, this class is strictly more expressive than both the Non-Counting and Locally Threshold Testable class (McNaughton and Papert, 1971; Thomas, 1997). It can be shown that the stringset defined by the constraint *ODD-Sibilants (see section 5.1) is not Non-Counting, but it is a regular stringset.[11]

5.3 Further evidence supporting the Subregular Hypotheses

So far in this section, we mentioned the most common types of attested markedness constraints. We did not provide an exhaustive encyclopedia of types in Humboldt's sense, but enough of one to motivate the encyclopedia of categories that was presented. The discussion was designed to convince readers that the markedness constraints found in natural language were present at the lowest levels of the hierarchy and that as one moves up the hierarchy, the kinds of constraints describable at these higher levels become less and less natural from a phonological point of view.

This helped motivate two hypotheses. The Strong Subregular Hypothesis (12) says that markedness constraints are SP or SL. The Weak Subregular Hypothesis (19) says that markedness constraints are TSL.

[10] For those familiar with MSO logic, here is a definition. Individual variables are denoted with x, y, and X denotes a set variable. $x \rhd y$ means y is a successor of x, and $x < y$ means x precedes y.

$$\begin{aligned} \texttt{closed}(X) &\stackrel{\text{def}}{=} (\forall x, y)\big[(x \in X \wedge x \rhd y) \Rightarrow y \in X\big] \\ x < y &\stackrel{\text{def}}{=} (\forall X)\big[(x \in X \wedge \texttt{closed}(X) \Rightarrow y \in X\big] \end{aligned}$$

[11] To see why the set of strings L containing only an even number of sibilants is not Non-Counting, the characterization in (18) can be used. For any k, observe that os^{2k}o belongs to L, but os^{2k+1}o does not.

While these constraints were motivated by appealing to common types of constraints, readers may wonder whether the hypotheses have been subjected to more rigorous empirical investigation. I would like to now give further evidence for the Strong and Weak Subregular hypotheses. First I will discuss studies of stress patterns in terms of the Subregular Hierarchies.

Jim Rogers and his students examined the stress patterns in the stress typology in Heinz (2007, 2009) with respect to the Strictly Local languages. There are 109 distinct patterns in this typology from over 400 languages. Edlefsen *et al.* (2008) report that 72% (of the 109 patterns) are SL_k with $k \leq 6$ and 49% are SL_3. The 28% are which are not SL_6 are unbounded stress patterns and are shown to not be SL for any k. Heinz (2014b) studies the four simplest types of unbounded stress patterns and shows that these are SP_2 once Culminativity (every word contains exactly one stress) is factored out. Culminativity has been argued to be a universal property of stress languages (Halle and Vergnaud, 1987; Hayes, 1995) and therefore this LT constraint may be thought to come for free. While recently Hyman (2009) suggests a more nuanced view, it seems ill-advised at this point to view the result regarding Culminativity in Heinz (2014b) as a rejection of the Subregular Hypotheses. Rogers *et al.* (2013) argue that the other unbounded stress patterns similarly factor into the conjunction of SL and PT constraints or SP and LT constraints. The other unbounded stress patterns continue to be the subject of current research.

Two potential counterexamples come from work by Thomas Graf. Graf (2010a) provides a formal analysis of the stress patterns of Creek and Cairene Arabic, as they have been characterized in the literature. According to Graf's analysis, these stress patterns are not Non-Counting and are properly regular. If the posited linguistic generalizations are correct, by Graf's analysis, these cases would constitute clear counterexamples to the Subregular Hypotheses. A critical aspect of the linguistic generalizations that Graf's result relies on is that there is no secondary stress in these languages. If the secondary stress were perceptible, however, the constraints needed to describe the pattern would become SL. Whether secondary stress is perceptible or not to speakers of these languages is not a settled issue, and so there is some question regarding the accuracy of the linguistic generalizations.

Thus, with only a couple of potential counterexamples meriting further study, the current understanding of the stress typology supports the Strong and Weak Subregular Hypotheses.

A second source of evidence in favor of these hypotheses comes from psycholinguistic experimentation (Lai, 2012, 2014). In a series of artificial language learning experiments, Lai compared how well native English speaking young adults could internalize the phonotactic pattern expressed by a SP constraint like *[+strident, α anterior]...[+strident, $-\alpha$ anterior] and a LT constraint like First/Last Harmony (*#[+strident, α anterior]...[+strident, $-\alpha$ anterior] #). Subjects in these experiments participated in a training session followed by a test session. In the training session, they are told they are going to hear the words of a foreign language. In the test session, they are given two words and are asked which one more likely belongs to the language they just heard.

Subjects belonged to one of three conditions, which determined the kind of training received. In the "Sibilant Harmony" (SH) condition, they were exposed to words which were well-formed according to the constraint *[+strident, α anterior]...[+strident, $-\alpha$ anterior]. In the "First/Last Harmony" (FL) condition, they were exposed to words which were well-formed according to the First/Last Harmony constraint. Subjects in the control condition received no training (and during test were asked which word they thought was a better word).[12] All

[12] Finley and Badecker (2009) found no difference between the absence of training and a control condition where

subjects were given the same test items in the test session.

As reported in Lai (2014), the results of this experiments were unambiguous. As expected, subjects in the control condition behaved according to chance. In the test session, subjects in the SH condition behaved in a manner consistent with internalizing the SP constraint because they consistently chose words in the test session that did not violate the constraint. On the other hand, subjects in the FL condition did not consistently choose words in the test session that did not violate the constraint. In fact, they behaved just like subjects in the SH condition! This is despite the fact that they were exposed to words like [soʃos] in training, which violate the constraint *[+strident, α anterior]...[+strident, $-\alpha$ anterior].

In sum, the experiments of Lai (2012, 2014) show that subjects find it easier to learn SP (or TSL) stringsets as opposed to LT ones. This evidence is consistent with both the Strong and Weak Subregular Hypotheses, but as far as I am aware, no other theory explains these results.

5.4 Constraints: A Summary

A summary of the foregoing section can be made very simply. Phonologists have identified many kinds of constraints on string representations. Stringsets can be classified according to two core computational parameters: the type of ordering relation (successor or precedence) and the type of logical power. Together, these provide a Constraint Definition Language in the sense of de Lacy (2011). With only a few potential exceptions meriting further empirical investigation, the stringsets corresponding to phonological constraints overwhelmingly belong to the SL, SP, and TSL regions in the encyclopedia of categories shown in Figure 2 on page 14, which are arguably the simplest.

Readers may wonder whether it is necessary to allow both ordering relations to be substructures in words.[13] If only the successor relation is permitted, then it is not possible to describe constraints like [+strident, α anterior]...[+strident, $-\alpha$ anterior] at the CNL, P, and FO levels. MSO logic is needed. MSO logic with successor, however, permits any regular stringset to be described. Similarly, if only the precedence relation is permitted, it is not possible to describe constraints like *NC̥ at the CNL, and P levels. FO logic is needed. In other words, the most restrictive theory is the Strong Subregular Hypothesis (12): phonological constraints are defined by banning substrings or subsequences.

Finally, we may wonder why this would be the case. If the Strong Subregular Hypothesis is correct, then the extensions of synchronic constraints are SL or SP stringsets (or conjunctions thereof). Why would this be? The idea expressed in Heinz (2010a) is that human learners generalize in particular ways—and the ways they generalize yield exactly these classes. Synchronic constraints are aspects of grammar, and grammars are learned. They are systems that grow and develop in response to environmental stimuli. The learning biases structure the classes; it is not the case that the nature of the class is independent of the learners. As Dresher (1999) and Heinz (2009) argue, this kind of explanation is not available to learners within OT

the words in the training condition contained words which were well-formed and ill-formed according to each targeted constraint type.

[13] Readers may also wonder why the substructure that picks out the first/last template [#_ ... _ #] is not available. Here the reason is simple: both the successor and precedence relations allow every word to have a model and for distinct words to have distinct models. This is not the case with the substructure indicated by the template [#_ ... _ #] is used to model words.

settings.

6 Transformations

Now we turn to transformations. From an OT perspective, this section is about faithfulness constraints and the map derived from the interaction of all the OT constraints. It is also about the typology of maps generated from a given CON. From a rule-based perspective, this section is about the extensions of individual phonological rules and their composition.

The computational theory of subregular relations is not as well developed as the Subregular Hierarchies. For example, logical characterizations of string relations have not yet been fully carried out. Previous work on subclasses of subregular relations is primarily limited to two classes known as the *left subsequential* and *right subsequential*. Essentially, these are classes of transformations with finite look-ahead; so they are "myopic" in the sense of Wilson (2003). More will be said about these classes in momentarily. I will keep the discussion at a high level and readers can find the definitions of them and other technical details in many different places, including Berstel (1979); Mohri (1997); Roche and Schabes (1997); Lothaire (2005); Sakarovitch (2009). A linguistically motivated treatment is given in Heinz and Lai (2013).

Much recent work at the University of Delaware has been to develop a hierarchy for string relations that is analogous to the one for stringsets shown in Figure 2 on page 14. The most notable advance in this regard has been work by Jane Chandlee in her thesis (Chandlee, 2014), and subsequent publications (Chandlee *et al.*, 2014a; Jardine *et al.*, 2014; Chandlee and Heinz, 2014). This work establishes relational counterparts to the Strictly Local stringsets, discusses their significant coverage of empirical phenomena, and explains how they can be learned.

6.1 The encyclopedia of types: maps

Phonologists are familiar with many ways in which underlying forms can differ from surface forms. Underlying segments may be deleted. Their order may be permuted (metathesis). The features composing the segments may change. Additionally, there may be elements in surface forms which were not present underlyingly (epenthesis).

Additionally, the contexts that trigger these changes are of different types. The contexts may be local to where the changes occur, by which I mean the distance between the trigger and the target falls within some fixed bound. A typical example is where the triggering context is adjacent to the change. For instance regressive nasal place assimilation is typically written in rule format as $[+\text{nasal}] \longrightarrow [\alpha\text{place}] \; / \; \_\_ \; [-\text{sonorant}, \alpha\text{place}]$. Alternatively, the triggers can be found arbitrarily far away, as found in examples of long-distance consonantal harmony (Hansson, 2001; Rose and Walker, 2004) and disharmony (see Suzuki, 1998; Bennett, 2013). Unlike the local cases, there appears to be no fixed bound on the distance between the trigger and the target.

The long-distance cases are of special interest, so I will largely follow the analysis of vowel harmony from Heinz and Lai (2013) to motivate the encyclopedia of categories introduced in the next section. Vowel harmony is a well-studied phenomenon in phonology (van der Hulst and van de Weijer, 1995; Baković, 2000; Finley, 2008; Nevins, 2010; Walker, 2011, and many others). Vowel harmony refers to a systematic pattern of pronunciation in which certain features of vowels which are different at the underlying level are the same at the surface level. Thus vowel harmony has been called a process of assimilation. One reason

it has attracted interest is because the affected vowels are not strictly speaking adjacent since consonants may intervene between them.

Schematically, an example of harmony can be distilled to the following mapping: /+−/↦[++]. We will read this mapping as follows. The underlying form and surface form each contain two vowels. The mapping shows the values of the feature F for each vowel. Virtually all features defining vowels (roundness, height, backness, advanced/retracted tongue root) have been shown to participate in harmony in some language, and F can be understood to be any of these features. At the underlying level, the values of the first and second vowel are /+/ and /−/, respectively. At the surface level, however, both vowels bear the value [+] for feature F. Since consonants are irrelevant, they are not shown, but it should be understood that the mapping above includes consonants which may precede or succeed any of the vowels.

Heinz and Lai (2013) analyze six types of logically possible vowel harmony maps discussed in the phonological literature. These will be discussed, along with two other types, all of which are summarized in Table 5 on page 32.

Following terminology introduced in the OT literature, I will refer to faithful vowels as those whose value of feature F stays constant in the underlying and surface forms. Unfaithful vowels are ones whose value of the feature F is not constant across the two levels.

One logically possible VH map maintains that the unfaithful vowels are always fewer in number than the faithful ones. (In words with an equal number of faithful and unfaithful vowels—so an even number of vowels—which feature values would change would be determined according to some default.) This map has been called been Majority Rules (MR) harmony. (20) shows some examples of this MR.

(20) (Majority Rules Harmony) { (+ + −, + + +), (+ − +, + + +), (− + +, + + +), (− − +, − − −), (− + −, − − −), (+ − −, − − −), . . . }

As has been discussed in the OT literature, this map is the optimal outcome of a two very simple constraints: a markedness constraint banning successive vowels with different values of feature F (Agree(F)) outranking the faithfulness constraint Ident(F). Baković (2000, 26) defines the term this way:

> When Agree[F] is dominant, it winnows the candidate set down to basically two candidates, one with all [αF] segments and the other with all [$-\alpha$F] segments. If IO-Ident[F] gets the next crack at the evaluation process, it will choose the one of these candidates that is least deviant from the input, regardless of the stem/affix or +/− distinctions. In other words, what ends up mattering is the *relative percentages* of [αF] and [$-\alpha$F] vowels in the input: the underlyingly greater number of [$-\alpha$F] vowels in [a map where /+ − −/ → [− − −]] gangs up on the lesser number of [αF] vowels, yielding the problematic effect that I call 'majority rule.' [emphasis in original]

As recognized by Baković, MR is unattested and considered phonologically bizarre. His solution adds certain locally conjoined constraints to CON, which he argues has the effect of ridding Majority Rules maps from the typology, and which he argues is independently needed for analyzing dominant/recessive types of vowel harmony. The point I wish to emphasize here however is that Majority Rules is a logically possible map, which is quite easy to generate in classic OT with a simple markedness constraint (depending on whether arbitrary many consonants may intervene between vowels determines whether Agree(F) is SL or TSL) and

standard faithfulness constraints.

Another possible map is one where the first or last vowel determines the features of the other vowels in the word. This has been called progressive harmony (PH) and regressive harmony (RH), respectively. Examples of a PH map are shown in (21).

(21) (Progressive Harmony) $\{ (++-,+++), (+-+,+++), (-++,---), (--+,---), (-+-,---), (+--,+++), \dots \}$

The inclusion of neutral vowels alters this map only slightly. Neutral are vowels resist harmonizing and either are skipped (in which case they are called transparent) or force subsequent vowels to harmonize with them (in which case they are called opaque). Following Heinz and Lai (2013), I will use the symbols [$\ominus$] and [$\boxminus$] to represent [−F] vowels that are transparent and opaque, respectively, and the symbols [$\oplus$] and [$\boxplus$] to represent [+F] vowels that are transparent and opaque, respectively. With this expanded alphabet, mappings like $/+\ominus-/\mapsto[+\ominus+]$ and $/+\boxminus+/\mapsto[+\boxminus-]$ would also belong to the PH map.

In contrast to the above, sometimes vowel harmony is bounded, in the sense that only the subsequent vowel is affected. I will call this Local Assimilation (LA) and (22) below illustrates this map where only the initial vowel is the trigger (cf. Yucatec Maya (Krämer, 2001)).

(22) (Local Assimilation) $\{ (+--,++-), (-++,--+), (+-+,+++), (-+-,---), \dots \}$

Early analyses of vowel harmony analyzed many patterns with an extension like bounded or unbounded PH or RH (van der Hulst and van de Weijer, 1995), but this type of analysis is present in recent work as well (Nevins, 2010).

Another logically possible, but unattested type of vowel harmony process has been called 'Sour Grapes' (SG) (Padgett, 1995; Wilson, 2003). Informally, SG is like progressive harmony except that later vowels only harmonize if no opaque vowels occur later in the word. If an opaque vowel occurs somewhere after the initial vowel, then non-neutral vowels between it and the initial vowel will not harmonize. (23) illustrates a SG map.

(23) (Sour Grapes) $\{ (+--,+++), (+-\boxminus,+-\boxminus), (+---,++++), (+--\boxminus,+--\boxminus), \dots \}$

Like Majority Rules, Sour Grapes has been argued to be phonologically bizarre. In particular, Wilson (2003) argues that harmony processes never 'look ahead beyond immediately adjacent segments. Wilson refers to the absence of look-ahead as a kind of myopia, and characterizes spreading processes as 'myopic. The Sour Grapes pattern disobeys Wilson's (2003) principle that phonological laws are myopic. In a Sour Grapes pattern, each vowel gets to 'look ahead' arbitrarily far to the end of the word to see if there is an opaque vowel downstream, and only harmonizes if it does not find one.

Classic OT has no difficulty generating SG maps. Under a typical analysis, there is a markedness constraint against segments that are [+F] but share all other features with $\boxminus$ (such as *$\boxplus$). Consequently, underlying /$\boxminus$/ can never surface as [$\boxplus$]. Finley (2008, p. 32) describes the rest of the OT analysis this way.

> Sour grapes harmony patterns occur when a blocker prevents spreading to vowels intervening between the source and the blocker. For the input $[/+-\boxminus/]$ ... the output $[[+-\boxminus]]$ will be optimal rather than the desired $[++\boxminus]$. This type of

> pathology is produced when the harmony-inducing constraint [AGREE(F)] does not localize the violation of harmony. In both the sour grapes candidate $[+-\boxminus]$ and the spreading candidate $[+ + \boxminus]$, there is only one locus of disagreement.... However, because the sour grapes candidate incurs no faithfulness violations, it will emerge as optimal.

Like MR, SG has received attention in the literature because it the optimal outcome of relatively simple constraints in OT (Wilson, 2003; McCarthy, 2004; Finley, 2008).

Other analyses of vowel harmony argue that the right generalization is that vowels in a word harmonize to a particular feature value, if it is present anywhere in the word. This analysis has been called the dominant/recessive (DR) since the feature F appears to have a dominant value (the one that vowels harmonize with) and a recessive value (the one that vowels don't harmonize with). In the example DR map below, the [+] value is the dominant one; so any underlying representation containing the harmonizing feature with the value [+] will surface so that the harmonizing feature in all vowels will also be [+].

(24) (Dominant/Recessive) $\{ (++-,+++), (+-+,+++), (-++,+++), (--+,+++), (-+-,+++), (+--,+++), (---,---), \ldots \}$

A similar analysis of VH patterns (shown in (25)) is one where the root vowel determines the features of the other vowels in the word. This kind of analysis has been termed 'Stem Control' (SC). Here the feature that spreads is determined by its morphological status, and not its inherent vowel as in DR harmony. Typically vowels agree with the closest stem vowel. Following Baković (2000), I use the $\sqrt{+}$ and $\sqrt{-}$ to indicate root vowels that are +F and −F, respectively.

(25) (Stem Control) $\{ (\sqrt{+}+-,\sqrt{+}++), (\sqrt{+}-+,\sqrt{+}++), (\sqrt{-}++,---), (\sqrt{-}-+,---), (\sqrt{-}+-,---), (\sqrt{+}--,\sqrt{+}++), (-\sqrt{+-}+,+\sqrt{+-}-), \ldots \}$

The last logically possible harmony pattern to be discussed I will call Circumambient Unbounded harmony (CU), following Jardine (2014a). This pattern is also like dominant/recessive harmony in that only one value of the feature triggers the harmony. However, CU harmony requires *two* /+F/ triggers which must *surround* the affected vowels, and which may be *arbitrarily far* from them. The examples in (26) illustrate.

(26) (Circumambient Unbounded) $\{ (+--,+--), (++-,++-), (+-+,+++), (+---,+---), (+--+,++++), (-+--+,-++++), \ldots \}$

The term 'circumambient' refers to two surrounding triggers and the term 'unbounded' refers to the absence of a bound on the distance between the two triggers. Yaka is the only language which appears to have CU vowel harmony (Hyman, 1998), though Jardine (2014a) argues Sanskrit n-retroflexion is formally similar. (As discussed further below in 6.3, Jardine (2014a) shows that there is a well-attested, common tonal pattern which is also circumambient unbounded.)

Table 5 is a reproduction of Table 3 from Heinz and Lai (2013), with the additions of local assimilation and circumambient unbounded harmony. It summarizes the encyclopedia of types outlined above. Phonological theory has posited maps like LA, PH, RH, DR, and SC, the consensus appears to be that MR and SG are not only unattested but bizarre. In fact they have been called pathological patterns in some works Wilson (2003, 2004); Finley (2008).

	w	LA(w)	PH(w)	RH(w)	DR(w)	SG(w)	MR(w)	CU(w)
a.	/+ − −/	[+ + −]	[+ + +]	[− − −]	[+ + +]	[+ + +]	[− − −]	[+ − −]
b.	/− + +/	[− − +]	[− − −]	[+ + +]	[+ + +]	[− − −]	[+ + +]	[− + +]
c.	/− − −/	[− − −]	[− − −]	[− − −]	[− − −]	[− − −]	[− − −]	[− − −]
d.	/− + −/	[− − −]	[− − −]	[− − −]	[+ + +]	[− − −]	[− − −]	[− + −]
e.	/+ − ⊟/	[+ + ⊟]	[+ + ⊟]	[− − ⊟]	[+ + ⊟]	[+ − ⊟]	[− − ⊟]	[+ − ⊟]
f.	/+ ⊖ −/	[+ ⊖ −]	[+ ⊖ +]	[− ⊖ −]	[+ ⊖ +]	[+ ⊖ +]	[− ⊖ −]	[+ ⊖ −]
g.	/+ − +/	[+ + +]	[+ + +]	[+ + +]	[+ + +]	[+ + +]	[+ + +]	[+ + +]

Table 5: Example mappings of underlying forms (w) given by local assimilation (LA), progressive harmony (PH), regressive harmony (RH), dominant/recessive harmony (DR), sour grapes harmony (SG), majority rules harmony (MR), and circumambient unbounded harmony (CU). Symbols [+] indicates a [+F] vowel and [−] indicates a [−F] vowel where "F" is the feature harmonizing. Symbols [⊟] and [⊖] are [−F] vowels that are opaque and transparent, respectively. (From Heinz and Lai (2013, p. 57).)

Lastly, setting aside tonal phonology for now, maps like CU are only marginally attested. As before, we ask the question: What principle or principles separates the linguistically-motivated generalizations (PH, RH, DR, SC) from the pathological ones (MR and SG) and the marginally attested ones (CU)?

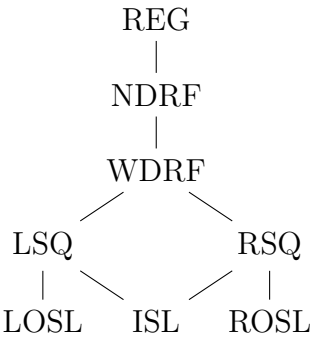
6.2 An encyclopedia of categories: string-to-string maps

Hierarchies of string-to-string transductions are not as well studied nor understood as classes of stringsets. Part of the issue is that string-to-string maps are inherently more complex; they have more parts than a stringset because in fact they are two stringsets—the domain and co-domain—and a map from elements of one to the other. Consequently, properties that converge for stringsets can diverge for string-to-string maps. One example is the class of regular stringsets. Regular stringsets are exactly those describable with MSO logical formulae with successor, deterministic finite-state acceptors and non-deterministic finite-state acceptors. (Informally, a finite-state machine is deterministic only if there is at most one path through the machine for each input; if there are some inputs with more than one path, it is non-deterministic.) However when the corresponding classes of string-to-string maps are considered, these three classes are distinct regions (Engelfriet and Hoogeboom, 2001).

The hierarchies of transductions that are known to exist, such as those described by Roche and Schabes (1997, chapter 1) and Engelfriet and Hoogeboom (2001) focus on the more expressive regions. There is very little work on regions that make the kinds of distinctions made in the Subregular Hierarchies discussed in the previous section.

Therefore, current answers to the questions at the end of the preceding section at present are unlikely to satisfy all readers since they are incomplete.

In this section, I will not explicate all the known regions, but only those that I think are currently most relevant for phonology. These regions are shown in Figure 4 on page 33. As before, lines connecting two regions indicate that the higher region properly includes the lower region. I will begin at the bottom and go up.



Names of the classes of the sets of string pairs			
Non-deterministic classes			
REG	Regular Relations		
NDRF	Non-Deterministic Regular Functions		
WDRF	Weakly Deterministic Regular Functions		
Deterministic classes			
LSQ	Left Subsequential Functions	RSQ	Right Subequential Functions
LOSL	Left Output Strictly Local	ROSL	Right Output Strictly Local
ISL	Input Strictly Local		

Figure 4: Subregular hierarchies of regular relations.

6.2.1 Input Strictly Local Functions

Input Strictly Local function generalize the notion of Strictly Local stringset. Recall the Strictly Local stringsets are Markovian in nature: the well-formedness of a string can be determined by examining the substrings of length k. Equivalently, this means that the well-formedness of any position in the string can be determined by checking the $k-1$ previous symbols. This is illustrated in Figure 5, for the case where $k = 2$.

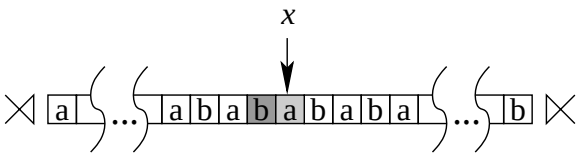


Figure 5: A schematic illustrating the Markovian nature of Strictly k-Local stringsets. Each element x of a string belonging to a strictly 2-local stringset depends only on the previous element. In other words, the lightly shaded cell only depends on the darkly shaded cell.

Input Strictly Local functions are similarly Markovian. The idea is that every element in the input string corresponds to a *string* of symbols in the output string. For any input symbol x its output string u will only depend on x and the previous $k-1$ elements of x in the input string. Figure 6 illustrates, for the case where $k = 2$.

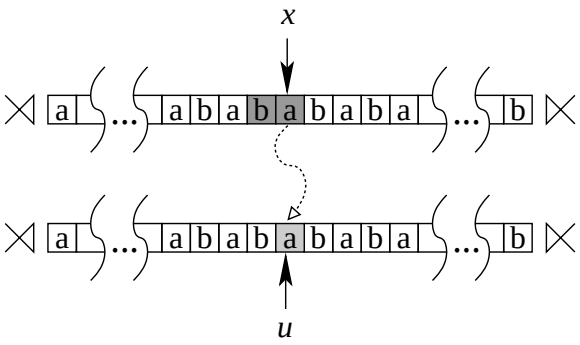


Figure 6: A schematic illustrating the Markovian nature of Input Strictly k-Local functions. For every Input Strictly 2-Local function, the output string u of each input element x depends only on x and the input element previous to x. In other words, the lightly shaded cell only depends on the darkly shaded cells.

Local Assimilation (LA) is ISL with $k = 3$. Basically if the previous two elements are /⋊+/ (or /⋊−/) then if the current input element is a non-neutral vowel, the output string will be [+] (or [−]).

(Chandlee, 2014) shows that ISL functions can model a range of local phonological processes, including substitution, insertion, deletion, and synchronic metathesis. More generally, she shows that given a mapping describable with a rule of the form $A \longrightarrow B \;/\; C \_\_ D$ where the set of strings in CAD is finite and the rule applies simultaneously then it is ISL for some k.

This result may seem counter-intuitive given the current discussion. A reader may wonder whether, especially given the diagram in Figure 6, how ISL functions can model any transformation triggered by any right context at all. As mentioned, every element in the input string corresponds to a *string* of elements in the output. These output strings can be any length, including length zero (the so-called 'empty' string). The option to output the empty string allows the function to wait until it has enough information to decide what to output. But importantly, the amount of input it needs to see to make this decision is *bounded*, by the specified value of k. For example consider regressive nasal place assimilation where underlying /inpa/↦[impa]. Each row in Table 6 shows how the output string is determined by each input element x and the input element preceding x. Since the output string at each point is

element preceding x	input element x	output string u
⋊	i	i
i	n	λ
n	p	mp
p	a	a

Table 6: Illustrating why transformations with right contexts can still be ISL. The symbol λ represents the empty string (the string of length zero).

determined by a window whose size is bounded by k, ISL maps are myopic in Wilson's (2003) sense.

Chandlee also investigated the approximately 5500 phonological processes (from over 500 languages) reported in the P-Base database (v1.95 Mielke, 2008). It was determined that over 95% of these patterns are ISL. Chandlee acknowledges that P-Base ought not be taken as representative of the cross-linguistic distribution of processes that target contiguous versus non-contiguous segments. However, given that it is the most comprehensive collection of processes of which we are aware, she deemed it necessary to survey.

Furthermore, Chandlee (2014); Chandlee *et al.* (2014a) also show how ISL functions can be efficiently learned from finitely many examples in the sense of Gold (1967) and de la Higuera (1997). This stands in stark contrast to the class of regular functions which cannot be so learned. Remarkably, Jardine *et al.* (2014) generalize this result to obtain an even more efficient learning algorithm for this class of functions.

6.2.2 Output Strictly Local Functions

A notable example of a map that Input Strictly Local functions are unable to model are ones like progressive harmony (PH) (21) above. Recall that a mapping like $/+\ -\ -\!-/\mapsto[+\ +\ +\!+]$ belongs to this map, and more generally for all numbers k, $/+\ -^k\ -/\mapsto[+\ +^k\ +]$ and $/-\ -^k\ -/\mapsto[-\ -^k\ -]$. Such a map cannot be Input Strictly Local for any k. This is because whether the last input element surfaces as $[+]$ or $[-]$ depends on an *input* element which is more than k input elements away.

Chandlee (2014) defines Left and Right Output Strictly Local functions (LOSL and ROSL) to address such maps. These capitalize on the output-oriented nature of many phonological processes (Kisseberth, 1970a; Prince and Smolensky, 1993, 2004). They are Markovian like ISL functions, but this time the context is found in the output string, not the input string. Specifically for Left (Right) OSL functions, for any input element x, its output string u will only depend on x and the previous (following) $k-1$ elements of the output string. The idea is that a function is Left or Right, depending on whether the left or right context in the output string matters. Figures 7 and 8 illustrate Left and Right OSL functions, respectively, for the case where $k = 2$.

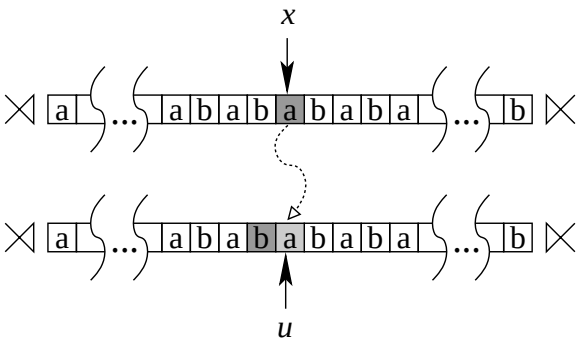


Figure 7: A schematic illustrating the Markovian nature of Left Output Strictly k-Local functions. For every Left Output Strictly 2-Local function, the output string u of each input element x depends only on x and the output element previous to u. As before, the lightly shaded cell only depends on the darkly shaded cells.

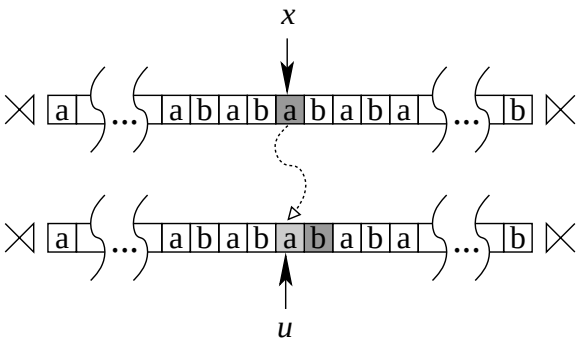


Figure 8: A schematic illustrating the Markovian nature of Right Output Strictly k-Local functions. For every Right Output Strictly 2-Local function, the output string u of each input element x depends only on x and the output element succeeding u. As before, the lightly shaded cell only depends on the darkly shaded cells.

Informally, Left and Right OSL functions can be thought of as characterizing the maps one can describe with rewrite rules that apply left-to-right or right-to-left (cf. the treatment of rule-application by Kaplan and Kay (1994)). This appears to be approximately correct, though certain details are still being worked out. However, we can say with certainty that the map PH is LOSL and the map RH is ROSL. More generally, such functions capture spreading processes such as progressive and regressive nasal spreading.

6.2.3 Subsequential Functions

In the abstract maps for vowel harmony discussed earlier, consonants were ignored. If arbitrary many consonants are allowed to intervene between the vowels then the PH and RH maps will not be LOSL nor ROSL, respectively. For the PH case, this means for all numbers k, $/+C^k-/\mapsto[+C^k+]$ and $/-C^k-/\mapsto[-C^k-]$. Such a map cannot be Left nor Right Output Strictly Local for any k because whether the last input element surfaces as $[+]$ or $[-]$ depends on an output element which is more than k output elements away. In a sense, at input element x, the functions cannot remember whether the preceding vowel in the output string was $[+]$ or $[-]$ because too many $[C]$s intervene.

We therefore move up the hierarchy in Figure 4. I note that as of yet there are no regions for string-to-string maps corresponding to the SP, LT, PT, TSL, or LTT stringsets.[14]

Informally, for Left (Right) Subsequential functions, each logically possible input string is classified as belonging to exactly one of finitely many regular stringsets. For any input element x, the output string u will only depend on x and the regular stringset to which its preceding input string belongs. Figure 9 illustrates left subsequential functions.

Even if arbitrary many consonants are allowed to intervene between the vowels then the PH and RH are in fact left and right subsequential, respectively. To see why, consider Table 7. Subsequential functions can 'remember' up to finitely many pieces of information about the left context; in Table 7, that the first vowel was [+F]. Thus even if k or more Cs then occur

[14]Some work exists that characterizes string-to-string maps which correspond to the NC stringsets (Lautemann *et al.*, 2001). Also the author has work in progress characterizing string-to-string maps for each of these regions.

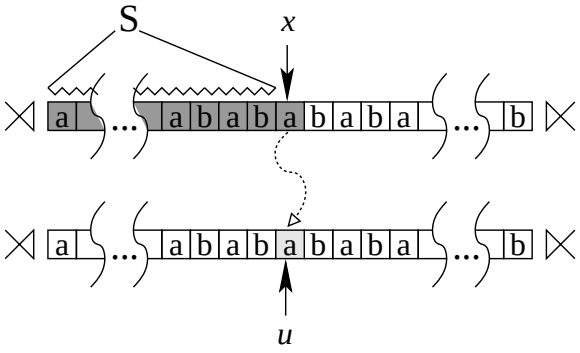


Figure 9: A schematic illustrating Left Subsequential functions. For every Left Subsequential function, the output string u of each input element x depends only on x and the stringset S to which the preceding input string belongs. As before, the lightly shaded cell only depends on the darkly shaded cells.

in the input string, the function simply outputs each C as it reads each C, without changing its memory state.

set to which string preceding x belongs	input element x	output string u
$\rtimes$	+	+
$\rtimes + C^*$	C	C
$\rtimes + C^*$	C	C
...	...	...
$\rtimes + C^*$	C	C
$\rtimes + C^*$	-	+

Table 7: Illustrating why PH is left subsequential even if arbitrarily many Cs intervene between vowels.

In the same way that ISL functions could 'look ahead' by writing empty output strings, subsequential functions can do so do as well. However, like the ISL functions, there is a sense in which left subsequential functions can look into the right context of the input element *only* some finite distance. There is a bound k on how far they can look ahead, which relates to the fact that it can only remember finitely much information about the input string. For this reason it is not possible to remember the *exact* preceding input string.

An example will help make this idea clear. The dominant/recessive (DR) map is neither left nor right subsequential. This is because, for all numbers k, $/-^k + -^k/ \mapsto [+^k + +^k]$ and $/-^k - -^k/ \mapsto [-^k - -^k]$. Such a map cannot be left subsequential because whether the first k input elements all surface as [+] or [−] depends whether the next element is [+] or not. Therefore, even though these functions might output λ for the first k input elements, if the [+] comes next, such functions would have to output k [+] symbols (and one more). But this is impossible because k can be *any* number and left subsequential functions can only classify the

preceding input string into one of *finitely* many categories. Table 8 illustrates this conundrum. For this reason, left subsequential functions are myopic in the sense that they cannot look

set to which string preceding x belongs	input element x	output string u
	$-$	λ
$-$	$-$	λ
$--$	$-$	λ
$\ldots$	$\ldots$	$\ldots$
$---\ldots$	$-$	λ
$---\ldots-$	$+$	$+++\ldots++$

Table 8: Illustrating why the dominant/recessive DR map is not left subsequential. The symbol λ represents the empty string. The problem is that the left subsequential function cannot remember exactly how many $-$ symbols occurred before the first $+$ (It cannot always correctly fill in the '$\ldots$').

unboundedly far into the right context.

Right subsequential functions are similar except that input strings *following* the input element are categorized into finitely many regular stringsets. Also, right subsequential functions can only 'look ahead' into the left context a finite distance (and an argument similar to the one made above shows why). It may be useful to think of right subsequential maps as the 'reverse' of left subsequential maps: if L is a left subsequential map then there is a right subsequential map L^r such that $(w, v) \in L$ iff $(w^r, v^r) \in L^r$ (where x^r is the reverse of the string x so $(abc)^r = cba$). From a processing perspective, one could say that left subsequential functions process strings left-to-right, and right subsequential functions process strings right-to-left.

At the University of Delaware in 2010, the question was asked whether the transformations from underlying to surface forms are left or right subsequential. In other words, we investigated what I will call the Subsequential Hypothesis.

(27) (Subsequential Hypothesis) Phonological transformations are left or right subsequential.

With one interesting class of exceptions discussed below, this hypothesis appears to be well-supported. This matters for two reasons. First, it is a stronger more restrictive hypothesis than the previously understood bound (phonology is regular, see section 4.3). Second, it has been known for quite some time that left subsequential (and right subsequential) functions are learnable in a particular sense (Oncina *et al.*, 1993). The algorithm presented there has even been adapted for use in phonology (Gildea and Jurafsky, 1996). In other words, if phonological transformations are subsequential, then the computational nature of phonological transformations directly provides purchase on the learning problem.

So what is the evidence which favors (27)? I will use the term 'subsequential' to mean either left or right subsequential. Chandlee (2014) proves that ISL, LOSL, and ROSL functions are subsequential; therefore, all the maps they cover are subsequential. Synchronically attested metathesis is also subsequential (Chandlee *et al.*, 2012; Chandlee and Heinz, 2012). Gainor *et al.* (2012) study the extensions of the vowel harmony maps in Nevins (2010) and

input				output
$- - + - -$	$\overset{DR_P}{\mapsto}$	$- - + + +$	$\overset{DR_R}{\mapsto}$	$+ + + + +$

Table 9: Map DR_p converts every $-$ after a $+$ to $+$ (like PH), and map DR_R convert every $-$ before a $+$ to $+$ (like RH). As indicated, the composition of these two maps yields the DR map.

input				output
$+ - - - \boxminus$	$\overset{L}{\mapsto}$	$+ \underline{?} \underline{?} \underline{?} \boxminus$	$\overset{R}{\mapsto}$	$+ - - - \boxminus$
$+ - - - -$	$\overset{L}{\mapsto}$	$+ \underline{?} \underline{?} \underline{?} \underline{?}$	$\overset{R}{\mapsto}$	$+ + + + +$

Table 10: Illustrations of the role of the new symbol $[\underline{?}]$ in the deterministic decomposition into a left subsequential function L and a right subsequential function R.

concludes they are subsequential. Since Nevins assumes a certain degree of underspecification, Heinz and Lai (2013) show that progressive and regressive vowel harmony with no underspecification pace OT (maps PH and RH above) are subsequential. Payne (2013) shows that long-distance consonant dissimilation maps described by Suzuki (1998) and Bennett (2013) is subsequential, and Luo (2013) shows that long-distance consonant assimilation maps described by Hansson (2008) and Rose and Walker (2004) is subsequential.

In some sense, these results are not too surprising because ultimately these results support Wilson's intuition that phonology is myopic. Nonetheless, if phonological myopia is best characterized as subsequentiality (or something stronger like ISL), then there is much concrete to gain: a theory which not only appears sufficiently expressive, but which is also more restrictive than previously entertained, and which has desirable learnability properties.

6.2.4 Weakly Deterministic Functions

Weakly deterministic functions are defined by Heinz and Lai (2013) as those maps that can be defined as the composition of a left subsequential and right subsequential function without the introduction of new alphabetic symbols. Heinz and Lai (2013) show that the dominant/recessive (DR) and stem-control (SC) maps are properly weakly deterministic. In fact the DR map is the composition of a map like progressive harmony (DR_P), which only spreads the dominant feature progressively and a map like regressive harmony (DR_R), which only spreads the dominant feature regressively. Table 9 illustrates. Since DR_p and DR_R are left and right subsequential, respectively, their composition is weakly deterministic.

Heinz and Lai (2013) conjecture that sour grapes (SG) is *not* weakly deterministic. They explain that SG can be described as the composition of a left subsequential function and a right subsequential function, but that crucially the intermediate form requires the use of an additional alphabetic symbol which they write as $[\underline{?}]$. Table 10 (adapted from their paper) illustrate the role an additional symbol plays in the decomposition. Essentially, $[\underline{?}]$ records the fact that this is a minus, which has a $[+]$ in its left context. So the right subsequential function R will rewrite this as $[-]$ or $[+]$ depending on whether there is a $[\boxminus]$ in the right context of the $[\underline{?}]$.

While many theorists have argued in favor of dominant/recessive and stem-control analyses of vowel harmony(Baković, 2000; Krämer, 2003), this is not a settled debate (Nevins, 2010). What Heinz and Lai (2013) show is that DR and SC maps are more computationally complex than PH and RH maps, and that SG maps are even more complex than these. If the debate is settled in favor of DR and SC maps then it means the subsequential hypothesis would be incorrect, and then the most restrictive hypothesis would retreat to the weakly deterministic region.

Jardine (2014a) argues that circumambient unbounded (CU) maps are also not weakly deterministic. He shows that CU maps and SG maps are similar in that in both cases, the information which determines whether a vowel is unfaithful is located in two distinct places: one location is arbitrarily far before the vowel and the other location is arbitrarily far after the vowel. While both CU and SG maps are the same with respect to the first place (an earlier + is required), the CU map also requires a + in the second place but the SG map requires the *absence* of a ⊟.

Jardine's result is perhaps the most serious challenge to the Subsequential Hypothesis (or a revised Weakly Deterministic hypothesis) because the best characterization of Yaka vowel harmony seems to be that it is circumambient unbounded (Hyman, 1998). However, this is the only known example of this type, and it is probably premature to reject the hypothesis on these grounds alone. I will return to this issue below.

6.2.5 Non-deterministic Regular Functions and Regular Relations

Non-deterministic regular functions can be defined in at least two ways (Elgot and Mezei, 1965). First, they can be defined as the composition of a left and right subsequential function, provided the intermediate string is allowed to use additional alphabetic symbols. As we have seen in the example of Sour Grapes (SG) in Table 10, these additional symbols allow certain types of information to become present in the string. Second, non-deterministic regular functions can be defined as those string-to-string functions that can be described with a non-deterministic finite-state transducer.

Both SG and CU maps thus properly belong to the non-deterministic regular function region (Heinz and Lai, 2013; Jardine, 2014a).

Non-deterministic finite-state transducers are a grammatical formalism that can also describe transformations which have more than one output for each input. In fact the class of transformations describable with non-deterministic finite-state transducers are called regular relations. Beesley and Kartunnen (2003) and Hulden (2009) develop toolkits for manipulating regular relations for describing the phonology and morphology of languages.

The Majority Rules map cannot even be described with a non-deterministic finite-state transducer. It is in fact non-regular (Riggle, 2004; Heinz and Lai, 2013). According to the hierarchy presented here, it is the most complex kind of maps under discussion.

Given that many phonological rules are optional, one may wonder whether it is appropriate to model individual transformations (as we have here) as functions instead of relations. There are two responses to this.

The first response is to say that the optionality is handled at a higher level of control than the individual transformation. This is essentially the position adopted in rule-based and constraint-based phonology. In rule-based phonology, the idea was that a rule was marked as optional. When deriving the output from an input and a rule marked as optional is en-

countered, additional, usually random, information is consulted (such as a coin flip) and the outcome determines whether the rule is applied or skipped. Thus the extension of the rule itself is still functional. Similarly, in stochastic OT (Boersma, 1998; Boersma and Hayes, 2001), a given constraint ranking has a functional extension, but a higher-level control process determines which particular constraint ranking will be utilized at any particular time.

The second response is to say that subclasses of regular relations are likely to follow the same lines developed here. (Mohri, 1997) establishes that as long as there is a bound on the amount of optionality, that many properties of subsequential functions are preserved. More recently, Beros and de la Higuera (2014) also show how to generalize subsequential functions in a way that permits a degree of optionality. While subclasses of classes have not been studied the fact that they preserve important aspects of the underlying finite-state transducers and that classes like ISL have automata-theoretic characterizations based on subsequential transducers (Chandlee, 2014; Chandlee *et al.*, 2014a) strongly suggests that subclasses like ISL which permit a degree of optionality are only waiting to be discovered.

6.3 Further evidence

There is some psycholinguistic evidence which supports the hypothesis that phonological transformations are regular. In a series of artificial language learning experiments, Finley (2008) compared how well native English speaking young adults could learn a Majority Rules (MR) type map as compared to a Progressive Harmony (PH) type map. These experiments were artificial learning experiments and subjects were either assigned to the MR, PH, or control conditions. Each subject received a training session and then performed in a test session. The results clearly established that subjects learned the PH harmony pattern, but not the MR pattern.

Finley (2008) also conducted a series of experiments to investigate the learnability of the Sour Grapes (SG) map. This is one way to test the hypothesis that Subsequential Hypothesis. Here the results were inconclusive, probably due to an interference of neutral vowels in the particular paradigm (though see Finley (2015)). However the subsequential hypothesis clearly predicts that SG should be more difficult to learn that PH, and so future research in this vein should be conducted to see whether the prediction is borne out.

An interesting source of evidence in favor of (a revised) Subsequential Hypothesis comes from work by Jardine (2014a). Jardine studies Unbounded Tone Plateauing (UTP) (Kisseberth and Odden, 2003; Hyman, 2011) and concludes that such transformations are also Circumambient Unbounded. In UTP, a string of underlying low tones (or unmarked vowels) are realized as high only if there is a high tone at *both* the left and right edges of this string. Jardine makes a persuasive case for a typological asymmetry between tonal patterns and segmental patterns. Several well-documented cases of UTP exist in the literature, despite the absence of comprehensive typological surveys. On the other hand, despite the existence of several surveys on long-distance harmony, the only CU maps known to be present in segmental phonology come from Yaka vowel harmony and Sanskrit n-retroflexion. Furthermore, Jardine shows that the evidence that the segmental maps are truly unbounded is weaker than the evidence that the UTP cases are unbounded.

In other words, Jardine's work shows that UTP—because of its widespread and well-documented existence—is a counterexample to the Subsequential Hypothesis (27). He suggests that it be revised as follows.

(28) (Revised Subsequential Hypothesis) Segmental transformations in phonology are left or right subsequential.

While Yaka tone and Sanskrit n-retroflexion are arguably counterexamples to this revised hypothesis, I think it is prudent to not reject the hypothesis on these grounds. Unlike UTP, these cases are rare and the evidence that they are truly CU maps, while compelling, is not as strong as it is for the UTP cases. Future research may lead to a better understanding of these languages.

Thus, Jardine shows that tonal patterns are different than segmental patterns, since they are arguably more complex. Paraphrasing Hyman (2011), tonal phonology really can do more than segmental phonology!

6.4 Transformations: a summary

A summary of the foregoing section can be made very simply. Phonologists have identified many ways in which underlying forms are transformed into surface forms. The study of subregular string-to-string maps has not yet been as articulated as the one of subregular stringsets. Nonetheless, the study of the typology of the attested transformations in the light of the existing categories yields similar conclusions. With one interesting class of exceptions which is largely confined to tonal phonology (CU maps), segmental transformations appear to overwhelmingly belong to the simplest maps in the encyclopedia of categories shown in Figure 4 on page 33.

Thus, even though an encyclopedia of categories for string-to-string maps as fine-grained as the one for stringsets does not yet exist, the work to date has nonetheless made important insights. The simplest maps are Markovian on the input or the output (ISL, LOSL, and ROSL), and very many phonological transformations belong to these classes. Transformations which are not Markovian in this sense involve long-distance harmony. Such patterns however are subsequential, which means they are still myopic in an important sense. This stands in contrast to the SG map which is not subsequential and the MR map, which is not even a regular relation. It is anticipated other subregular regions for maps analogous to the SP or TSL regions will be developed that better characterize long-distance transformations.

The asymmetry between tonal CU maps and segmental CU maps noticed by Jardine (2014a) is perhaps the most difficult to interpret. How can it be that some formal mechanism is available to one aspect of the grammar but not to another? Perhaps it is an indicator of the modularity of grammar (Heinz and Idsardi, 2011, 2013). Jardine's work then can be understood as providing support for Hyman's thesis, that tonal phonology is in fact different from segmental phonology (Hyman, 2011). As a separate module of the grammar, it has resources available to it that are not available everywhere else.

7 Summary and Implications for the phonological component

In this section, I would like to review the main lessons for phonological theory to be taken from the computational analyses reviewed so far in this chapter.

7.1 Phonological generalizations have strong computational properties

From the computational perspective, most, if not all, phonological generalizations obey very strong computational laws. This is what typological analysis of phonological constraints and transformations that was reviewed in sections 5 and 6 reveals. Phonological generalizations (both stringsets and maps) belong to small, well-defined regions *within* the region of regular stringsets and maps, and these regions are at the *bottom* of the subregular hierarchies shown in Figures 2 and 4.

7.2 Problems with optimization

Current phonological theories does not account for these laws. Since Prince and Smolensky (1993), optimization has been a central feature of phonological theory including classical OT (Kager, 1999), stratal OT (Kiparsky, 2000), harmonic grammar (Smolensky and Legendre, 2006; Potts *et al.*, 2008), maximum entropy (Goldwater and Johnson, 2003; Hayes and Wilson, 2008), and harmonic serialism (McCarthy, 2008). One of the most compelling features of optimization is the idea that complex patterns within and across languages arise from the interaction of simple constraints. The celebrated examples of syllabification in Berber (Dell and Elmedlaoui, 1985), complex margin avoidance in Yokuts (Kisseberth, 1970b), and the many solutions to the international conspiracy *NC̥ (Pater, 2000) in terms of optimization all attest to this. However, if "complex patterns arising from the interaction of simple constraints" is optimization's greatest strength, it is also its greatest weakness.

As explained in section 4.3, computational analysis has revealed that phonological transformations are regular. But even with regular constraints and a regular GEN, optimization can result in *non-regular* maps (Frank and Satta, 1998). Optimization is very powerful because very complex patterns can indeed arise from simpler constraints. Majority Rules is a case in point (Riggle, 2004; Heinz and Lai, 2013).

This overgeneration problem is not specific to classical OT. It is a problem for Stratal OT, Harmonic Serialism and Maximum Entropy theories as well. This is not a controversial point. It is in fact just one more example of how "complex patterns arise from the interaction of simple constraints." It may be possible to add constraints to CON as in Baković (2000) to avoid generating MR type maps. Different types of constraints, such as targeted constraints (ColinWilson, 2001; Bakovic and Wilson, 2000; Baković, 2004), or ones which operate over turbid structures (Finley, 2008) may be invoked.

However, it is not enough to show that MR is avoided. One must show that all non-regular maps are avoided. Frank and Satta (1998) write "It remains an open problem to characterize precisely the generative capacity...of [Optimality Systems]'s with other assumptions about the formal power of GEN and the constraints." While it still remains an open problem today, the markedness constraints involved in generating Majority Rule patterns are SL or TSL. In other words, even the simplest constraints under optimization can generate non-regular maps. Perhaps future research will show there is an straightforward way to prevent non-regular maps from occurring in the factorial typology, but to me the prospects seem dim.

For even if it were possible to add new constraints (or constraint types) to CON to avoid deriving non-regular maps, there is a problem. These constraints would be in service of deriving a generalization that is already very simple to state: phonological transformations are regular.

It means that in order for optimization to be the right theory of phonology that the constraints over which optimization operates have to be designed to blunt the power of optimization!

And these problems are just to avoid generating non-regular maps. The revised Subsequential Hypothesis limits the kinds of humanly possible segmental maps. The same kinds of problems mentioned above exist for developing a theory of CON which guarantees that segmental maps are always subsequential and avoid generating non-myopic maps like Sour Grapes (Jardine, 2014a).

From a computational perspective then, optimization appears to be too powerful a tool. Critics of optimization have generally focused on the fact OT undergenerates the typology with respect to opacity(Idsardi, 1998, 2000; McCarthy, 2007; Buccola, 2013), but the overgeneration problem is equally pressing. In both cases, new constraint types or optimizing architectures are introduced with the ultimate purpose to make the optimal maps ones that conform to the comptuational generalizations stated in the present chapter.

Thus, from a computational perspective, with respect to the dual goals of developing both an adequately expressive theory and a maximally restrictive theory of phonology, optimization misses the mark. It is neither adequately expressive (because of opaque maps) nor sufficiently restrictive (because it generates non-regular maps like Majority Rules).

Of course the question can never be "Is Optimization the Right Theory?" After all, what is the alternative?

7.3 Organizing phonological theory around these computational properties

One alternative suggested by the work reviewed in this chapter is that the computational properties highlighted here—and not optimization—be taken as the organizing principles of the phonological component of grammar. Constraints on surface forms are, with few exceptions, banning substrings and subsequences (section 5). Phonological transformations which are intuitively 'local' are also among the simplest types of logically possible maps (section 6).

But is this theory adequately expressive? We have focused on maps which correspond to individual rules, so can the theory being suggested handle opacity? This is work in progress, but it appears so. Chandlee *et al.* (2014b) show that ISL functions can represent opaque maps. In fact, Chandlee *et al.* (2015) report that every opaque map described in Baković's (2007) is ISL (and as mentioned Chandlee and her collaborators have established algorithms for learning the ISL functions).

Another potential criticism is that computational properties highlighted here do not take into account phonetic substance. In this way, many possible subregular constraints and maps are not likely phonological ones because they are phonetically unnatural. I will put aside the question of whether phonetically unnatural constraints and maps are phonologically possible and learnable, and just assume for the purposes of discussion that they are not.

In my view, it is a feature and not a bug that formal and substantive issues are separated. I am not so extreme as Hale and Reiss (2000) as to deny substance a role altogether, but I do think science proceeds by factoring complex systems. Ever since Chomsky and Halle (1968, chapter 9), it has been clear that formal and substantive constraints on phonological systems are distinct. While details have not yet been worked out, substantive constraints on phonological systems would simply be *in addition* to the formal constraints being proposed here. This is really no different than the program offered by Chomsky and Halle (1968) or, for

that matter, by Hayes *et al.* (2004), which adopts optimization as a formal system and *adds* substantive constraints to the nature of CON.

Finally, another argument that could be put in favor of optimization is that it readily lends itself to a theory of learning (Tesar, 1995; Tesar and Smolensky, 1998, 2000; Tesar, 2014, and many others). This is a good argument. However, I have tried to highlight the fact that the subregular regions to which phonological patterns appear to belong *also* readily lend themselves to a theory of learning (Oncina and Garcia, 1991; Heinz, 2007, 2010a,b; Chandlee, 2014; Chandlee *et al.*, 2014a; Jardine *et al.*, 2014). Furthermore, if people generalize from data in the way suggested by these learning procedures, it explains the computational nature of the phonological patterns. This is contrast to the learners in optimization-based theories (and variants thereof), where the nature of the phonological patterns is completely explained by what belongs and what does not belong to CON.

7.4 Next steps

If the computational properties highlighted in this chapter are taken seriously, there are many subsequent issues to attend to. There are several possibilities, but I will focus on the following four:

- better characterizing long-distance transformations;
- better understanding the role of abstractness;
- better understanding the non-string representations of words; and
- testing the predictions of these hypotheses in imaginative ways.

Each of these is a current focus of research at the University of Delaware.

Chandlee (2014) provides clear definition of locality and shows that many maps, which are intuitively local (and some that are not) meet this definition. Her definition generalizes the notion of Strictly Local stringsets to maps. Strictly Piecewise stringsets (Rogers *et al.*, 2010) can describe many long-distance constraints (Heinz, 2007, 2010a) as can Tier-Based Strictly Local stringsets (Heinz *et al.*, 2011). What are the corresponding generalizations of the SP and TSL stringsets for maps and what range of the phonological transformations do they cover? Do they have learnability properties like the SP and TSL regions?

It is well known that deterministic regular functions are less expressive than non-deterministic functions (see Figure 4 on page 33. Elgot and Mezei Elgot and Mezei (1965) show a deep connection between non-determinism and abstractness: Basically any non-deterministic function can be described as the composition of two deterministic functions provided the intermediate form is allowed to make use of symbols *not in the input alphabet*. The fact that the 'intermediate' alphabet contains symbols not in the input alphabet introduces a degree of abstractness (the extra symbols represent abstract information). This appears to be exactly the dividing line between dominant/recessive harmony patterns on one side and sour grapes harmony on the other. Better understanding the role of abstractness in maps (and stringsets is important).

Another example of the interplay comes from a theorem by Medvedev (1964), which says that every regular language is the homomorphic image of strictly 2-local language. In layman's terms this means every regular stringset, which can be used to model complex kinds of constraints, can be derived from a strictly 2-local stringset, which belongs to the lowest level of the Subregular Hierarchies. It suggests that what looks complex is actually very simple. But

the trick is that the SL_2 language has a bigger alphabet and the latent information hidden in the more complex regular language is made explicit in the SL_2 language. Thus, by making our alphabet larger and more abstract we simplify constraints we may want to state. But the price is that the alphabet no longer represents observables (so one consequence is learning remains as difficult as before).

A related issue is non-string based representations. The main idea is that there is an interplay between the generative capacity of a formalism, the representation, and power of logic. We have seen this already with respect to how order is represented. Representing order with successor and precedence has the following implication: if you want to represent any regular stringset you will need MSO logic with successor, but only FO logic with precedence. Another example is discussed in in section 8 below has to do with constraints on syllabic structure.

Finally, if these computational properties are to become part of the ontology of a theory of phonology, then much research is possible which tests the psychological validity of this ontology. The artificial language learning experiments by Finley (2008) and Lai (2012, 2014) are just one example of the kind of research that can be carried out. So far this work supports the hypotheses offered here. But currently there are more hypotheses than there are experiments testing them.

8 Representational Issues

Before concluding, I would like to address the issue of representation. The above computational analysis appears to rest on the assumption that words must be modeled as strings. We modeled phonotactic knowledge and markedness constraints with infinite sets of *strings* and we modeled the transformation from underlying to surface forms with infinite sets of *string* pairs. We argued that it is these objects whose nature we are interested in. The nature of these objects directly informs the questions in (2). As with circles (remember circles?), the nature of these objects is to some extent independent of the grammars used to describe them.

Phonologists are well aware that other representations of words exist, which are not based on strings and that phonological theories have employed many kinds of data structures. Non-linear representations of words including autosegmental representations, (Goldsmith, 1976), the grid (Liberman and Prince, 1977), feature-geometric representations (Clements, 1985), and gestural scores (Browman and Goldstein, 1992). Instead of strings, these theories employ graph-like data structures. Therefore, it is reasonable to wonder how much of the foregoing analysis depends on the string-based representation employed.

In this section, I would first like to explain that the extensions are not limited to string representations. The concept of "extension" of a grammar is much more general.

Then I would like to explain why the results in this chapter still matter, even though they use string-based representations. There are two parts to this. The first part reiterates the fact that string-based representations are in fact widely adopted in phonology and explains why they are widely adopted. Insofar as these reasons are compelling, the results in this chapter matter.

The second part argues that even if the right representations is not string-based, studies like the ones reviewed in this chapter are a necessary step for understanding the computational nature of phonological patterns. I will explain why subregular hierarchies like the ones

presented in this chapter for strings exist for these other representational schemes for words, even if they have not yet all been discovered.

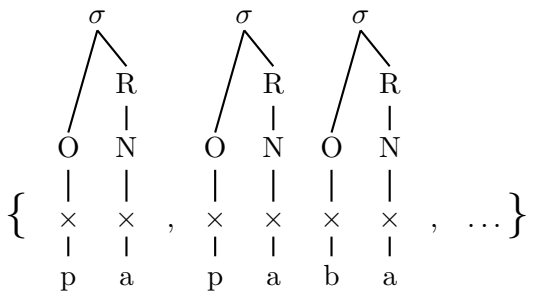
The main conclusion is that the interplay between the choice of representation and the computational principles presented here are likely to be fruitful areas of research in the coming decades.

8.1 Extensions without strings

In section 3, I discussed the concept "extension of a grammar," and suggested that for constraints the extensions are infinite sets of strings and for transformations they are infinite sets of pairs of strings. More generally, the extension is an infinite set of objects, and the objects can be anything so long as it is well-defined.

For instance, A constraint like NoCoda could be defined so that the well-formed elements of its extension includes objects like the ones shown below.

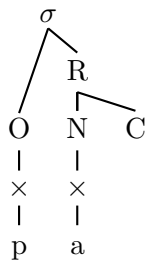
(29)



Non-linear representations such as these are common in phonological theory, and have brought much insight. Similarly, when discussing transformations from underlying to surface forms, the left and right elements in each pair can also be represented with a non-linear, graph-like representation similar to the ones shown in (29).

In order to think about the extensions of constraints and transformations, it is necessary to think about the representations of words. The constraint definition will determine which representations of the "logically possible" representations are well-formed and which are not. For instance, we may wonder whether the following representation is permitted by the theory and if so whether it violates NoCoda.

(30)



At issue of course is the marked structure(s) NoCoda is supposed to define when graph-like representations such as those in the preceding examples are adopted. There are several

possibilities. Just having a node in the graph labeled with "C" for instance could be sufficient, in which case (30) would violate it. Or it may be necessary for the node labeled "C" to dominate a phonetic element (in which case (30) would not violate it). Or it may be that the labels in the preceding examples are just ornaments for phonologists for readability, and that what really matters is that there is consonantal material after the nucleus dominated by the same syllable node.

A more difficult question raises itself with these representations with respect to the constraint ONSET. At the very least this constraint *requires* an "O" node to be present. It does not ban a substructure, suggesting this constraint is at the Propositional level. Here again we see an interplay between the choice of representation and the power of the constraint. To state ONSET, this enriched representation requires a more complex constraint type than the string representation with the latent syllable boundary, which can define ONSET as the Strictly 2-Local constraint (*.V).

These are all interesting possibilities that have been explored to various degrees in the phonological literature. The point I wish to make is a fairly obvious one: representation of words matter when defining constraints or transformations. The extensions of the constraints and transformations will be in terms of these representations. String representations were used throughout this chapter, but some other representation could have been used, and computational analysis could have preceded on those representations instead.

8.2 Justifying string representations

I begin this discussion with the concept of string. Strings are one of the most basic data structures (Lothaire, 1997, 2005). They are sequences of events. They are typically defined inductively with the primitive operation of concatenation and an alphabet: there is a unique string of zero length (the base case) and then if w is a string and there is a symbol a in the alphabet then the concatenation of w and a (written wa) is also a string (the inductive case). It is natural to think of phonological forms in terms of strings. The act of speech can be thought of as a sequence of articulatory or acoustic events. Writing systems are string-based representations of speech.

The most compelling reason to study string representations is that phonologists use them. One reason may be practical: it is more convenient to typeset strings than graph-like representations. But even from a theoretical perspective, the fact that representations were prominently debated at one point in the history of phonological theory (mainly during the 1970s and 1980s) does not mean that such issues are necessarily live today. As Hyman (2014) discusses, phonological theory has moved away from issues of representation, and string-based representations are prominent, even in explaining long-distance harmony (Rose and Walker, 2004; Walker, 2011; Nevins, 2010), a domain in which at one time autosemental spreading analyses seemed particularly well suited (Hayes, 1986). Tonal phonology is less amenable to string-based representations (for discussion, see also Marlo (2007)), but even the principles governing autosegmental representations have yet to address every question posed by every language (Hyman, 2014).

A second reason to work with strings is that the string is a fundamental data structure that has been well-studied (Lothaire, 1997, 2005). This helps make studying stringsets and string-to-string maps the easiest place to start. If we want to understand how computational principles play out with complicated data structures, we better first understand how they play

out with simpler structures like strings.

The particular subregular hierarchies being established for strings are likely to have analogues for other data structures. For instance, Strictly 2-Local sets of tree structures have been studied and they can be used to describe context-free string sets (Rogers, 1994). More generally, regular tree languages are also well-studied (Comon *et al.*, 2007). Closer to issues in phonological theory, Jardine (2014b) shows how Strictly Local autosegmental representations can be defined (and how in certain circumstances the No Crossing Constraint is a Strictly Local constraint). While hierarchies as articulated as the ones for strings do not yet exist for these other data structures, they are in fact a focus in theoretical computer science (Rozenberg and Salomaa, 1997).

In short, the methodological points being made in this chapter stand regardless of whether or not the representations are strings are something else. Computational principles provide a natural encyclopedia of categories with which the typology of phonological generalizations can be illuminated.

9 Conclusion

This has been a long chapter. Thankfully, the conclusion can be brief. Phonology is about how underlying lexical representations are transformed into surface ones. An important question asks about the cross-linguistic nature of these transformations. Grammars are typically conceived as generating patterns; these patterns are extensions of the grammar in the same way the extension of an algebraic equation is the set of points satisfying that equation. Computational analysis studies these *extensions*, and such analysis of phonological generalizations is ongoing. Nonetheless, the results so far reveal that despite the cross-linguistic diversity, there are very strong, specific, universal computational properties shared by almost all phonological patterns. The few potential counter-examples are of special interest and deserve further study.

Explaining these plausibly universal computational properties of phonological patterns is hard for theories that rely on optimization as a central organizing feature of the theory, but is straightforward if the computational properties highlighted within this chapter become the organizing principles themselves. These principles are natural for many reasons, only some of which could be covered here. Also, there is a clear sense in which these principles derive from principles of inference and learning. While there is still much work to do, a theory of phonology built around these computational principles promises to be sufficiently expressive, maximally restrictive, and learnable.

References

Applegate, R.B. 1972. Ineseño Chumash grammar. Doctoral dissertation, University of California, Berkeley.

Applegate, R.B. 2007. *Samala-English dictionary : a guide to the Samala language of the Ineseño Chumash People*. Santa Ynez Band of Chumash Indians.

Baković, Eric. 2000. Harmony, dominance and control. Doctoral dissertation, Rutgers University.

Baković, Eric. 2004. Unbounded stress and factorial typology. In *Optimality Theory in Phonology: A Reader*, edited by John McCarthy. Blackwell, London. ROA-244, Rutgers Optimality Archive, http://roa.rutgers.edu/.

Baković, Eric. 2007. A revised typology of opaque generalisations. *Phonology* 24:217–259.

Baković, Eric. 2013. *Blocking and Complementarity in Phonological Theory*. Bristol, CT: Equinox.

Bakovic, Eric, and Colin Wilson. 2000. Transparency, strict locality, and targeted constraints. In *Proceedings of the 19th West Coast Conference on Formal Linguistics*, edited by Roger Billerey and Brook Danielle Lillehaugen, 43–56. Somerville, Mass.: Cascadilla Press.

Beckman, Jill. 1998. Positional faithfulness. Doctoral dissertation, University of Massachusetts, Amherst.

Beesley, Kenneth, and Lauri Kartunnen. 2003. *Finite State Morphology*. CSLI Publications.

Bennett, William. 2013. Dissimilation, consonant harmony, and surface correspondence. Doctoral dissertation, Rutgers University.

Benua, Laura. 1995. Identity effects in morphological truncation. In *Papers in Optimality Theory*, edited by Jill Beckman, Laura Walsh Dickey, and Suzanne Urbanczyk, 77–136. Amherst, Mass.: GLSA Publications.

Benua, Laura. 1997. Transderivational identity: Phonological relations between words. Doctoral dissertation, University of Massachusetts, Amherst.

Beros, Achilles, and Colin de la Higuera. 2014. A canonical semi-deterministic transducer. In *Proceedings of the Twelfth International Conference on Grammatical Inference (ICGI 2014)*, edited by Alexander Clark, Makoto Kanazawa, and Ryo Yoshinaka, vol. 34, 33–148. JMLR: Workshop and Conference Proceedings.

Berstel, Jean. 1979. *Transductions and Context-Free languages*. Teubner-Verlag.

Blevins, Juliette. 2004. *Evolutionary Phonology*. Cambridge University Press.

Boersma, Paul. 1998. *Functional phonology: Formalizing the interactions between articulatory and perceptual drives*. University of Amsterdam. LOT International Series 11. The Hague: Holland. Http://www.fon.hum.uva.nl/paul/diss/.

Boersma, Paul, and Bruce Hayes. 2001. Empirical tests of the gradual learning algorithm. *Lingustic Inquiry* 32:45–86.

Browman, C., and L. Goldstein. 1992. Articulatory phonology: An overview. *Phonetica* 155–180.

Buccola, Brian. 2013. On the expressivity of optimality theory versus ordered rewrite rules. In *Proceedings of Formal Grammar 2012 and 2013*, edited by Glyn Morrill and MarkJan Nederhof, vol. 8306 of *Lecture Notes in Computer Science 8036*, 142–158. Berlin Heidelberg: Springer-Verlag.

Büchi, J. Richard. 1960. Weak second-order arithmetic and finite automata. *Mathematical Logic Quarterly* 6:66–92.

Chandlee, Jane. 2014. Strictly local phonological processes. Doctoral dissertation, The University of Delaware.

Chandlee, Jane, Angeliki Athanasopoulou, and Jeffrey Heinz. 2012. Evidence for classifying metathesis patterns as subsequential. In *The Proceedings of the 29th West Coast Conference on Formal Linguistics*, 303–309. Cascillida Press.

Chandlee, Jane, Rémi Eyraud, and Jeffrey Heinz. 2014a. Learning strictly local subsequential functions. *Transactions of the Association for Computational Linguistics* 2:491–503.

Chandlee, Jane, and Jeffrey Heinz. 2012. Bounded copying is subsequential: Implications for metathesis and reduplication. In *Proceedings of the 12th Meeting of the ACL Special Interest Group on Computational Morphology and Phonology*, 42–51. Montreal, Canada: Association for Computational Linguistics.

Chandlee, Jane, and Jeffrey Heinz. 2014. Strictly local phonological processes. Accepted with revisions to *Linguistic Inquiry*.

Chandlee, Jane, Adam Jardine, and Jeffrey Heinz. 2014b. Learning repairs for marked structures. Poster presented at the Annual Meeting of Phonology. MIT.

Chandlee, Jane, Adam Jardine, and Jeffrey Heinz. 2015. Representing and learning opaque maps. Poster presented at GALANA 6. UMCP.

Chomsky, Noam. 1951. Morphophonemics of Modern Hebrew. Doctoral dissertation, University of Pennsylvania, Philadelphia. Published by Garland Press, New York, 1979.

Chomsky, Noam. 1956. Three models for the description of language. *IRE Transactions on Information Theory* 113124. IT-2.

Chomsky, Noam. 1965. *Aspects of the theory of syntax*. Cambridge, MA: MIT Press.

Chomsky, Noam, and Morris Halle. 1968. *The Sound Pattern of English*. New York: Harper & Row.

Clements, George N. 1985. The geometry of phonological features. *Phonology Yearbook* 2:225–252.

ColinWilson. 2001. Consonant cluster neutralisation and targeted constraints. *Phonology* 18:147–197.

Comon, H., M. Dauchet, R. Gilleron, C. Löding, F. Jacquemard, D. Lugiez, S. Tison, and M. Tommasi. 2007. Tree automata techniques and applications. Available on: http://www.grappa.univ-lille3.fr/tata. Release October, 12th 2007.

Dell, Francois, and Mohamed Elmedlaoui. 1985. Syllabic consonants and syllabification in Imdlawn Tashlhiyt Berber. *Journal of African Languages and Linguistics* 7:105–130.

Dresher, Elan. 1999. Charting the learning path: Cues to parameter setting. *Linguistic Inquiry* 30:27–67.

Edlefsen, Matt, Dylan Leeman, Nathan Myers, Nathaniel Smith, Molly Visscher, and David Wellcome. 2008. Deciding strictly local (SL) languages. In *Proceedings of the Midstates Conference for Undergraduate Research in Computer Science and Mathematics*, edited by Jon Breitenbucher, 66–73.

Elgot, C. C., and J. E. Mezei. 1965. On relations defined by generalized finite automata. *IBM Journal of Research and Development* 9:47–68.

Endress, Ansgar D., Marina Nespor, and Jacques Mehler. 2009. Perceptual and memory constraints on language acquisition. *Trends in Cognitive Science* 13:348–353.

Engelfriet, Joost, and Hendrik Jan Hoogeboom. 2001. Mso definable string transductions and two-way finite-state transducers. *ACM Trans. Comput. Logic* 2:216–254.

Finley, Sara. 2008. The formal and cognitive restrictions on vowel harmony. Doctoral dissertation, Johns Hopkins University, Baltimore, MD.

Finley, Sara. 2015. Learning non-adjacent dependencies in phonology: Transparent vowels in vowel harmony. *Language* In press.

Finley, Sara, and William Badecker. 2009. Artificial language learning and feature-based generalization. *Journal of Memory and Language* 61:423–437.

Fougeron, Cécile, and Patricia A. Keating. 1997. Articulatory strengthening at edges of prosodic domains. *Journal of the Acoustic Society of America* 101:3728–3740.

Frank, Robert, and Giorgo Satta. 1998. Optimality Theory and the generative complexity of constraint violability. *Computational Linguistics* 24:307–315.

Gainor, Brian, Regine Lai, and Jeffrey Heinz. 2012. Computational characterizations of vowel harmony patterns and pathologies. In *The Proceedings of the 29th West Coast Conference on Formal Linguistics*, 63–71.

Gallagher, Gillian. 2010. Perceptual distinctness and long-distance laryngeal restrictions. *Phonology* 27:435–480.

García, Pedro, and José Ruiz. 2004. Learning k-testable and k-piecewise testable languages from positive data. *Grammars* 7:125–140.

Garcia, Pedro, Enrique Vidal, and José Oncina. 1990. Learning locally testable languages in the strict sense. In *Proceedings of the Workshop on Algorithmic Learning Theory*, 325–338.

Gazdar, G., and G. Pullum. 1982. Natural languages and context-free languages. *Linguistics and Philosophy* 4:469–470.

Gerdemann, Dale, and Gertjan van Noord. 2000. Approximation and exactness in finite state optimality theory. In *Proceedings of the Fifth Meeting of the ACL Special Interest Group in Computational Phonology*, 34–45.

Gildea, Daniel, and Daniel Jurafsky. 1996. Learning bias and phonological-rule induction. *Computational Linguistics* 24:497–530.

Gold, E.M. 1967. Language identification in the limit. *Information and Control* 10:447–474.

Goldsmith, John. 1976. Autosegmental phonology. Doctoral dissertation, MIT, Cambridge, MA.

Goldwater, Sharon, and Mark Johnson. 2003. Learning OT constraint rankings using a maximum entropy model. In *Proceedings of the Stockholm Workshop on Variation within Optimality Theory*, edited by Jennifer Spenader, Anders Eriksson, and Östen Dahl, 111–120. Stockholm: Stockholm University.

Gorman, Kyle. 2013. Generative phonotactics. Doctoral dissertation, University of Pennsylvania.

Graf, Thomas. 2010a. Comparing incomparable frameworks: A model theoretic approach to phonology. *University of Pennsylvania Working Papers in Linguistics* 16:Article 10. Available at: http://repository.upenn.edu/pwpl/vol16/iss1/10.

Graf, Thomas. 2010b. Logics of phonological reasoning. Master's thesis, University of California, Los Angeles.

Graf, Thomas. 2013. Local and transderivational constraints in syntax and semantics. Doctoral dissertation, University of California, Los Angeles.

Hale, Mark, and Charles Reiss. 2000. Substance abuse and dysfunctionalism: Current trends in phonology. *Linguistic Inquiry* 31:157–169.

Halle, Morris. 1959. *The Sound Pattern of Russian*. The Hague, Mouton.

Halle, Morris. 1978. Knowledge unlearned and untaught: What speakers know about the sounds of their language. In *Linguistic Theory and Psychological Reality*. The MIT Press.

Halle, Morris, and Jean-Roger Vergnaud. 1987. *An Essay on Stress*. The MIT Press.

Hansson, Gunnar. 2001. Theoretical and typological issues in consonant harmony. Doctoral dissertation, University of California, Berkeley.

Hansson, Gunnary. 2008. Diachronic explanations of sound patterns. *Language and Linguistics Compass* 2:859–893.

Hayes, Bruce. 1986. Assimilation as spreading in Toba Batak. *Linguistic Inquiry* 17:467–99.

Hayes, Bruce. 1995. *Metrical Stress Theory*. Chicago University Press.

Hayes, Bruce, Robert Kirchner, and Donca Steriade, eds. 2004. *Phonetically-Based Phonology*. Cambridge University Press.

Hayes, Bruce, and Colin Wilson. 2008. A maximum entropy model of phonotactics and phonotactic learning. *Linguistic Inquiry* 39:379–440.

Heinz, Jeffrey. 2007. The inductive learning of phonotactic patterns. Doctoral dissertation, University of California, Los Angeles.

Heinz, Jeffrey. 2009. On the role of locality in learning stress patterns. *Phonology* 26:303–351.

Heinz, Jeffrey. 2010a. Learning long-distance phonotactics. *Linguistic Inquiry* 41:623–661.

Heinz, Jeffrey. 2010b. String extension learning. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, 897–906. Uppsala, Sweden: Association for Computational Linguistics.

Heinz, Jeffrey. 2014a. Computational theories of learning and developmental psycholinguistics. In *The Oxford Handbook of Developmental Linguistics*, edited by Jeffrey Lidz, William Synder, and Joe Pater. Oxford University Press. To appear.

Heinz, Jeffrey. 2014b. Culminativity times harmony equals unbounded stress. In *Word Stress: Theoretical and Typological Issues*, edited by Harry van der Hulst, chap. 8. Cambridge, UK: Cambridge University Press.

Heinz, Jeffrey, and William Idsardi. 2011. Sentence and word complexity. *Science* 333:295–297.

Heinz, Jeffrey, and William Idsardi. 2013. What complexity differences reveal about domains in language. *Topics in Cognitive Science* 5:111–131.

Heinz, Jeffrey, Anna Kasprzik, and Timo Kötzing. 2012. Learning with lattice-structured hypothesis spaces. *Theoretical Computer Science* 457:111–127.

Heinz, Jeffrey, and Regine Lai. 2013. Vowel harmony and subsequentiality. In *Proceedings of the 13th Meeting on the Mathematics of Language (MoL 13)*, edited by Andras Kornai and Marco Kuhlmann, 52–63. Sofia, Bulgaria.

Heinz, Jeffrey, Chetan Rawal, and Herbert G. Tanner. 2011. Tier-based strictly local constraints for phonology. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics*, 58–64. Portland, Oregon, USA: Association for Computational Linguistics.

de la Higuera, Colin. 1997. Characteristic sets for polynomial grammatical inference. *Machine Learning* 27:125–138.

Hulden, Mans. 2009. Finite-state machine construction methods and algorithms for phonology and morphology. Doctoral dissertation, University of Arizona.

van der Hulst, Harry, and Jeroen van de Weijer. 1995. Vowel harmony. In *The Handbook of Phonological Theory*, edited by John A. Goldsmith, 495–534. Cambridge, Mass., and Oxford, UK: Blackwell.

von Humboldt, Wilhelm. 1999. *On Language*. Cambridge Texts in the History of Philosophy. Cambridge University Press. Translated by Peter Heath. Originally published 1836.

Hyman, Larry. 1998. Positional prominence and the 'prosodic trough' in yaka. *Phonology* 15:41–75.

Hyman, Larry. 2011. Tone: Is it different? In *The Blackwell Handbook of Phonological Theory*, edited by John A. Goldsmith, Jason Riggle, and Alan C. L. Yu, 197–238. Wiley-Blackwell.

Hyman, Larry. 2014. How autosegmental is phonology? *The Linguistic Review* 31:363–400.

Hyman, Larry M. 2009. How (not) to do phonological typology: the case of pitch-accent. *Language Sciences* 31:213 – 238. Data and Theory: Papers in Phonology in Celebration of Charles W. Kisseberth.

Idsardi, William. 1998. Tiberian Hebrew spirantization and phonological derivations. *Linguistic Inquiry* 29:37–73.

Idsardi, William J. 2000. Clarifying opacity. *The Linguistic Review* 17:337–350.

Jardine, Adam. 2014a. Computationally, tone is different. Qualifying Paper, University of Delaware. Under Review with Phonology.

Jardine, Adam. 2014b. Representing and learning phonological tiers. Talk presented at the Northeast Computational Phonology Meeting (NECPHON). November 15. NYU.

Jardine, Adam, Jane Chandlee, Rémi Eyraud, and Jeffrey Heinz. 2014. Very efficient learning of structured classes of subsequential functions from positive data. In *Proceedings of the Twelfth International Conference on Grammatical Inference (ICGI 2014)*, edited by Alexander Clark, Makoto Kanazawa, and Ryo Yoshinaka, vol. 34, 94–108. JMLR: Workshop and Conference Proceedings.

Johnson, C. Douglas. 1972. *Formal Aspects of Phonological Description*. The Hague: Mouton.

Jurafsky, Daniel, and James Martin. 2008. *Speech and Language Processing: An Introduction to Natural Language Processing, Speech Recognition, and Computational Linguistics*. 2nd ed. Upper Saddle River, NJ: Prentice-Hall.

Kager, René. 1999. *Optimality Theory*. Cambridge University Press.

Kaplan, Ronald, and Martin Kay. 1994. Regular models of phonological rule systems. *Computational Linguistics* 20:331–378.

Karttunen, Lauri. 1998. The proper treatment of optimality in computational phonology. In *FSMNLP'98*, 1–12. International Workshop on Finite-State Methods in Natural Language Processing, Bilkent University, Ankara, Turkey.

Kiparsky, Paul. 2000. Opacity and cyclicity. *The Linguistic Review* 17:351–366.

Kisseberth, Charles. 1970a. On the functional unity of phonological rules. *Linguistic Inquiry* 1:291–306.

Kisseberth, Charles. 1970b. On the functional unity of phonological rules. *Linguistic Inquiry* 1:291–306.

Kisseberth, Charles, and David Odden. 2003. Tone. In *The Bantu Languages*, edited by D. Nurse and G. Philippson. New York: Routledge.

Kobele, Gregory. 2006. Generating copies: An investigation into structural identity in language and grammar. Doctoral dissertation, University of California, Los Angeles.

Koskenniemi, Kimmo. 1983. Two-level morphology. Publication no. 11, Department of General Linguistics. Helsinki: University of Helsinki.

Krämer, Martin. 2001. Yucatec maya vowel alternations harmony as syntagmatic identity. *Zeitschrift für Sprachwissenschaft* 20:175–217.

Krämer, Martin. 2003. *Vowel Harmony and Correspondence Theory*. Berlin: Mouton de Gruyter.

de Lacy, Paul. 2011. Markedness and faithfulness constraints. In *The Blackwell Companion to Phonology*, edited by M. V. Oostendorp, C. J. Ewen, E. Hume, and K. Rice. Blackwell.

Lai, Regine. 2012. Domain specificity in phonology. Doctoral dissertation, University of Delaware.

Lai, Regine. 2014. Learnable vs. unlearnable harmony patterns. *Linguistic Inquiry* To appear.

Lautemann, Clemens, Pierre McKenzie, Thomas Schwentick, and Heribert Vollmer. 2001. The descriptive complexity approach to {LOGCFL}. *Journal of Computer and System Sciences* 62:629 – 652.
URL http://www.sciencedirect.com/science/article/pii/S0022000000917422

Liberman, Mark, and Alan Prince. 1977. On stress and linguistic rhythm. *Linguistic Inquiry* 8:249–336.

Lothaire, M., ed. 1997. *Combinatorics on Words*. Cambridge, UK, New York: Cambridge University Press.

Lothaire, M., ed. 2005. *Applied Combinatorics on Words*. 2nd ed. Cmbridge University Press.

Luo, Huan. 2013. Long-distance consonant harmony and subsequantiality. Qualifying Paper, UD Linguistics PhD Progam.

Marlo, Michael. 2007. The verbal tonology of lumarachi and lunyala: two dialects of luluyia. Doctoral dissertation, University of Michigan.

McCarthy, John. 2003. OT constraints are categorical. *Phonology* 20:75–138.

McCarthy, John. 2004. Headed spans and autosegmental spreading. Unpublished manuscript, UMass, Amherst.

McCarthy, John J. 2007. *Hidden Generalizations: Phonological Opacity in Optimality Theory*. London: Equinox.

McCarthy, John J. 2008. The gradual path to cluster simplification. *Phonology* 25:271–319.

McNaughton, Robert, and Seymour Papert. 1971. *Counter-Free Automata*. MIT Press.

Medvedev, Yu. T. 1964. On the class of events representable in a finite automaton. In *Sequential Machines; Selected Papers*, edited by Edward F. Moore, 215–227. Addison-Wesley. Originally published in Russian in Avtomaty, 1956, 385–401.

Mielke, Jeff. 2008. *The Emergence of Distinctive Features*. Oxford: Oxford University Press.

Mohri, Mehryar. 1997. Finite-state transducers in language and speech processing. *Computational Linguistics* 23:269–311.

Nevins, Andrew. 2010. *Locality in Vowel Harmony*. Cambridge, MA: The MIT Press.

Ohala, J.J. 1981. The listener as a source of sound change. In *Papers from the parasession on language and behavior: Chicago Linguistics Society*, edited by C.S. Masek, R.A. Hendrik, and M.F. Miller, 178–203.

Oncina, Jose, and Pedro Garcia. 1991. Inductive learning of subsequential functions. Tech. Rep. DSIC II-34, University Politécnia de Valencia.

Oncina, José, Pedro García, and Enrique Vidal. 1993. Learning subsequential transducers for pattern recognition tasks. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 15:448–458.

Padgett, Jaye. 1995. Partial class behavior and nasal place assimilation. In *Proceedings of the 1995 Southwestern Workshop on Optimality Theory*, edited by K. Suzuki and D. Elzinga.

Pater, Joe. 2000. *NC̥. In *Proceedings of NELS*, edited by K. Kusumoto, vol. 26, 227–239. Amherst, MA: GLSA.

Pater, Joe. 2001. Austronesian nasal substitution revisited: What's wrong with *NC (and what's not). In *Segmental Phonology in Optimality Theory: Constraints and Representations*, edited by Linda Lombardi, 159–182. Cambridge: Cambridge University Press. Available (1995) on Rutgers Optimality Archive.

Payne, Amanda. 2013. Dissimilation as a subsequential process. Qualifying Paper, UD Linguistics PhD Progam.

Popper, Karl. 1959. *The Logic of Scientific Discovery*. Basic Books, Inc. New York.

Potts, Christopher, Joe Pater, Rajesh Bhatt, and Michael Becker. 2008. Harmonic grammar with linear programming: From linear systems to linguistic typology. Rutgers Optimality Archive ROA-984.

Potts, Christopher, and Geoffrey Pullum. 2002. Model theory and the content of ot constraints. *Phonology* 19:361–393.

Prince, Alan, and Paul Smolensky. 1993. Optimality Theory: Constraint interaction in generative grammar. Tech. Rep. 2, Rutgers University Center for Cognitive Science.

Prince, Alan, and Paul Smolensky. 2004. *Optimality Theory: Constraint Interaction in Generative Grammar*. Blackwell Publishing.

Riggle, Jason. 2004. Generation, recognition, and learning in finite state Optimality Theory. Doctoral dissertation, University of California, Los Angeles.

Roark, Brian, and Richard Sproat. 2007. *Computational Approaches to Morphology and Syntax*. Oxford: Oxford University Press.

Roche, Emmanuel, and Yves Schabes. 1997. *Finite-State Language Processing*. MIT Press.

Rogers, James. 1994. Studies in the logic of trees with applications to grammatical formalisms. Doctoral dissertation, University of Delaware. Published as Technical Report 95-04 by the Department of Computer and Information Sciences.

Rogers, James, Jeffrey Heinz, Gil Bailey, Matt Edlefsen, Molly Visscher, David Wellcome, and Sean Wibel. 2010. On languages piecewise testable in the strict sense. In *The Mathematics of Language*, edited by Christian Ebert, Gerhard Jäger, and Jens Michaelis, vol. 6149 of *Lecture Notes in Artifical Intelligence*, 255–265. Springer.

Rogers, James, Jeffrey Heinz, Margaret Fero, Jeremy Hurst, Dakotah Lambert, and Sean Wibel. 2013. Cognitive and sub-regular complexity. In *Formal Grammar*, edited by Glyn Morrill and Mark-Jan Nederhof, vol. 8036 of *Lecture Notes in Computer Science*, 90–108. Springer.

Rogers, James, and Geoffrey Pullum. 2011. Aural pattern recognition experiments and the subregular hierarchy. *Journal of Logic, Language and Information* 20:329–342.

Rose, Sharon, and Rachel Walker. 2004. A typology of consonant agreement as correspondence. *Language* 80:475–531.

Rozenberg, G., and A. Salomaa, eds. 1997. *Handbook of Formal Languages: Beyond Words*, vol. 3. Springer.

Sakarovitch, Jaques. 2009. *Elements of Automata Theory*. Cambridge University Press. Translated by Reuben Thomas from the 2003 edition published by Vuibert, Paris.

Scott, Dana, and Michael Rabin. 1959. Finite automata and their decision problems. *IBM Journal of Research and Development* 5:114–125.

Shieber, Stuart. 1985. Evidence against the context-freeness of natural language. *Linguistics and Philosophy* 8:333–343.

Sipser, Michael. 1997. *Introduction to the Theory of Computation*. PWS Publishing Company.

Smolensky, Paul, and Géraldine Legendre. 2006. *The Harmonic Mind: From Neural Computation to Optimality-Theoretic Grammar*. Cambridge, MA: MIT Press.

Suzuki, Keiichiro. 1998. A typological investigation of dissimilation. Doctoral dissertation, University of Arizona, Tucson, AZ.

Tesar, Bruce. 1995. Computational Optimality Theory. Doctoral dissertation, University of Colorado at Boulder.

Tesar, Bruce. 2014. *Output-driven Phonology*. Cambridge University Press.

Tesar, Bruce, and Paul Smolensky. 1998. Learnability in optimality theory. *Linguistic Inquiry* 229–268.

Tesar, Bruce, and Paul Smolensky. 2000. *Learnability in Optimality Theory*. MIT Press.

Thomas, Wolfgang. 1997. Languages, automata, and logic. vol. 3, chap. 7. Springer.

Turing, Alan. 1937. On computable numbers, with an application to the entscheidungsproblem. *Proceedings of the London Mathematical Society* s2:230–265.

Walker, Rachel. 2011. *Vowel patterns in language*. Cambridge: Cambridge University Press.

Wilson, Colin. 2003. Analyzing unbounded spreading with constraints: marks, targets, and derivations. Unpublished manuscript, UCLA.

Wilson, Colin. 2004. Experimental investigation of phonological naturalness. In *Proceedings of WCCFL 22*, 534–546.