The Power of Locality Domains in Phonology*

Thomas Graf

Stony Brook University

Abstract

Domains play an integral role in linguistic theories. This paper combines locality domains with current work on the computational complexity of phonology. The first result is that if a specific formalism — Strictly Piecewise (SP) grammars — is supplemented with a mechanism to enforce first-order definable domain restrictions, its power increases so much that it subsumes almost the full hierarchy of subregular languages. However, if domain restrictions are based on linguistically natural intervals, one instead obtains an empirically more adequate model. On the on hand, this model subsumes only those subregular classes that have been argued to be relevant for phonotactic generalizations. On the other hand, it excludes unnatural generalizations that involve counting or elaborate conditionals. It is also shown that SP grammars with interval-based domains are theoretically learnable unlike SP grammars with arbitrary, first-order domains.

Keywords: locality, phonological domains, first-order logic, tiers, subregular hierarchy, strictly piecewise, learnability

1 Introduction

It has been known for a long time that the phonological systems of natural languages are finite-state in nature and thus generate regular languages (Johnson 1972; Kaplan and Kay 1994). However, this characterization is insufficent in the sense that phonological dependencies instantiate much simpler patterns — the regular

*I am greatly indebted to two anonymous reviewers and the editors of this issue. Their detailed feedback led to major changes that, hopefully, have resulted in a much more accessible and phonologically grounded paper. This paper has also profited tremendously from regular discussions of subregular phonology with Alëna Aksënova, Hyunah Baek, Aniello De Santo, and Chikako Takahashi.

languages only provide an upper bound on phonological expressivity. This realization has led to increased efforts in recent years to pinpoint the computational nature of phonology more precisely (Heinz 2009, 2010; Graf 2010a,b; Rogers et al. 2013; Chandlee 2014; Jardine 2016).

The aforementioned research relies on results presented by McNaughton and Papert (1971), which are reviewed in Rogers and Pullum (2011). Phonological considerations led to the study of two more subregular classes of languages and their relation to those previously established (Rogers et al. 2010; Heinz et al. 2011). Taken together, this research shows how to divide the regular languages into complexity classes, allowing the degree of pattern complexity to be understood with greater precision. A tentative consensus has emerged that the Strictly Local (SL), Strictly Piecewise (SP), and Tier-based Strictly Local (TSL) classes — and the grammars that generate languages in these classes — provide reasonable approximations of natural language phonotactics. These classes are among the least complex in the hierarchy and have cognitive interpretations (Rogers et al. 2013).

However, the claims about the empirical suitability of these classes implicitly assumes that phonological generalizations apply uniformly across a given domain, such as the word. Decades of research have established that phonological generalizations are sometimes best understood as involving multiple domains, with generalizations applying only within or across smaller or larger ones. The literature is so extensive that only some highlights are provided here (Rubach 2008,Selkirk 2011, Scheer 2012, and Shwayder 2015 provide recent surveys). Kiparsky (1982) posits a major distinction between word-level and sentence-level processes that is couched in terms of Lexical Phonology and its various domains of rule application. Booij and Rubach (1984) argue that one also needs a split between morphological and prosodic domains within words, whereas Kaye (1995) proposes a more general system where words may be structurally decomposed into subdomains, reminiscent of how syntactic trees are made up from phrases of various sizes. In the other direction, there is also plenty of evidence for phonological domains above the word level but below the sentence level, such as intonational phrases (Selkirk 1984; Nespor and Vogel 1986). And of course feet and syllables have proven indispensable in the analysis of stress patterns (Hayes 1995). Given the central role of domains in phonology, it is imperative their study be part of the subregular analyses of phonology.

In this paper I show how domains increase the expressivity of the SP class.

The first result establishes that if domains can be postulated freely, the resulting class — called Domain-Based SP (DBSP) — subsumes every subregular class except the regular class itself. The second result shows that the increase in expressive power is much less pronounced if the domains must be natural from a linguistic perspective. Specifically, smaller domains must delimit specific intervals within a larger domain. I call the resulting class *Interval-Based SP grammars* (IBSP).

There are many advantages to understanding phonotactic patterns in terms of the IBSP class. The SL, SP and TSL classes are contained within IBSP so any phonotactic pattern expressible with any of those grammars is expressible with an IBSP grammar. IBSP is more than the sum of these three classes, however. This increased expressivity is useful to capture previously unbounded tone plateauing, which is outside the SL, SP, and TSL classes. Also, the increased expressivity of IBSP is much less than DBSP. IBSP excludes certain kinds of unnatural phonotactic patterns that DBSP does not. Finally, IBSP is naturally parameterized in a way that permits learnabilty results. I conclude phonological domains are thus a welcome and highly useful addition that unifies phonologically natural subregular regions — but they must be interval-based and not arbitrary lest the formalism lose most of its restrictive force.

A clarification is in order regarding the scope of the paper. While some work in the subregular literature has focused on the mappings from underlying forms to surface forms (see Chandlee 2014, Chandlee et al. 2014, Chandlee et al. 2015, and references therein), this paper and the subregular classes mentioned above are about phonotactic generalizations, i.e. surface forms. The formalisms discussed here only need to be able to distinguish well-formed surface forms from ill-formed ones; underlying representations are never considered. Nonetheless I will sometimes describe certain phenomena such as tone plateauing as a process. This is merely a convenient expository device, and only the surface forms produced by these processes are pertinent to this paper.

The paper is laid out as follows. Section 2 familiarizes the reader with the phonologically important subregular classes SL (2.1), SP (2.2), and TSL (2.3). Their formal commonalities and empirical differences are briefly outlined in Sec. 2.4. Building on these observations, I show in Sec. 3.1 that enriching SP grammars with locality domains allows for an elegant account of several problematic phenomena, but at the cost of severe overgeneration. Limiting domain restrictions to precisely defined intervals, on the other hand, keeps generative capacity in check (3.2). Section 3.3 explains why this limitation is also essential for learnability.

2 Subregular Phonotactics

The later sections of this paper require the reader to be familiar with the subregular classes SL, SP, and TSL and how they are defined as fragments of first-order logic. I only present the essentials here. Readers familiar with first-order logic and these subregular classes can skip ahead to Sec. 2.4, where I contrast the three from an empirical perspective. Readers interested in a thorough exposition of first-order logic are invited to consult the textbooks Ebbinghaus et al. (1996) and Enderton (2001). For the subregular classes and how they relate to phonotactics, readers are referred to the overview presented in Heinz (2015) or Heinz (2010). Technical treatments of these classes are given in Rogers et al. (2013) and Heinz et al. (2011).

2.1 Strictly Local Grammars

In the subregular approach to phonology the simplest model of phonotactics is provided by strictly local grammars. A strictly k-local (SL-k) grammar is a finite set of k-grams, each one of which is a description of an illicit sequence of segments. Consider for instance the strictly 2-local grammar $\{\rtimes V, CC, VV, C\ltimes\}$. Here $\rtimes$ and $\ltimes$ denote the left and right edge of the string, respectively. This grammar forbids all words that start with a vowel ($\rtimes$V), or contain two adjacent consonant (CC) or two adjacent consonants (VV), or end in a consonsant (C$\ltimes$). In other words, this grammar describes a language with a CV syllable template. It essentially forbids certain *substrings* of length 2.

Note that C and V are used for the sake of brevity. If the language in question has only the vowels [a], and [u], plus two consonants [p] and [k], the set of sequences that violate the CV template could be compiled out into 12 bigrams.[1]

$$(1)\quad \begin{array}{llll} \rtimes\mathrm{a} & \mathrm{pp} & \mathrm{aa} & \mathrm{p}\ltimes \\ \rtimes\mathrm{u} & \mathrm{pk} & \mathrm{au} & \mathrm{k}\ltimes \\ & \mathrm{kp} & \mathrm{ua} & \\ & \mathrm{kk} & \mathrm{uu} & \end{array}$$

We can increase the length of the k-grams to capture phonotactic generalizations with an extended locality domain. Consider a markedness constraint against intervocalic voiceless fricatives. An SL-3 grammar {VFV} accomplishes this where

V stands for any vowel and F for any voiceless fricative. An SL-2 grammar is insufficient. Banning bigrams VF and FV does not succeed because either one is fine as long as F is not both immediately followed and immediately preceded by vowels. Therefore a grammar that operates with bigrams is either too permissive or too restrictive, only trigrams are large enough to correctly capture this markedness constraint.

SL grammars can be recast in terms of formulas of first-order logic, and this logical perspective is important for the remainder of the paper, in particular Sec. 3. I refrain from giving a full exposition of first-order logic here — a purely intuitive treatment suffices for our purposes. One may think of the grammar for intervocalic voicing above as a single constraint that can be expressed directly in first-order logic:

For all segments x_1, x_2, x_3	$\forall x_1, x_2, x_3\Big[$
if x_1 is immediately followed by x_2	$x_1 \; S \; x_2$
and	$\wedge$
x_2 is immediately followed by x_3	$x_2 \; S \; x_3$
then	$\rightarrow$
they are not labeled V, F, and V, respectively.	$\neg\big(V(x_1) \wedge F(x_2) \wedge V(x_3)\big)\Big]$

In the formula above, the usage of $x_1 \; S \; x_2$ to indicate that x_2 immediately follows x_1 is motivated by the mathematical convention to call x_2 the successor of x_1. As discussed next, replacing S by another relation yields a very different formalism (Rogers et al. 2013).

2.2 Strictly Piecewise Grammars

SL grammars — despite their simplicity — provide a reasonable model of local dependencies in phonology, including those created by iterated processes such as progressive vowel harmony. But they provably cannot regulate dependencies that involve two segments that may be arbitrarily far apart. An example of that is sibilant harmony in Samala, as discussed in Heinz (2015) based on data from Applegate (1972). In Samala, no word may contain both [s] and [ʃ], irrespective of how far the two are apart. Suppose one wanted to account for this with an SL-k grammar. This grammar would need to ban all k-grams that contain both [s] and [ʃ]. For example, with $k = 3$ the grammar would include Csʃ, Cʃs, ʃCs, sCʃ, sʃC, ʃsC, and all the counterparts where consonants are replaced by vowels. But such an SL-k grammar enforces sibilant harmony only as long as the two sibilants are separated by at most $k-2$ segments. If they are farther apart, they do not belong to the same k-gram and hence they are outside the grammar's purview. Therefore long distance processes are beyond the capabilities of SL.

A simple way of accounting for such unbounded dependencies is to maintain the k-gram approach but change how k-grams are interpreted. In an SL grammar the bigram sʃ indicates that s must not be immediately followed by ʃ. But if we switch from *immediately follow* (the relation S) to *follow at any distance* (the precedence relation $\prec$), k-grams enforce unbounded dependencies instead of local ones. Under this interpretation, the strings sʃ and ʃs are *subsequences* of length 2, and as such any string containing either subsequence is considered ill-formed. Grammars where k-grams enforce such unbounded dependencies are called *Strictly k-Piecewise* (SP-k; Rogers et al. 2010).

The first-order logic perspective reveals the close parallel between SL and SP. Both use the same template for formulas but use different relations. Just consider the SL and SP formulas for the set {sʃ, ʃs}. They are identical *modulo* the relation that holds between x_1 and x_2.[2]

(2) a. $\forall x_1, x_2 \Big[x_1 \, S \, x_2 \rightarrow \neg\big(\mathrm{s}(x_1) \wedge \mathrm{ʃ}(x_2)\big) \wedge \neg\big(\mathrm{ʃ}(x_1) \wedge \mathrm{s}(x_2)\big)\Big]$

b. $\forall x_1, x_2 \Big[x_1 \prec x_2 \rightarrow \neg\big(\mathrm{s}(x_1) \wedge \mathrm{ʃ}(x_2)\big) \wedge \neg\big(\mathrm{ʃ}(x_1) \wedge \mathrm{s}(x_2)\big)\Big]$

While the SL interpretation of {sʃ, ʃs} only enforces harmony under adjacency, the SP interpretation applies harmony across the entire string.

2.3 Tier-Based Strictly Local Grammars

Most attested dependencies that hold within a phonological word are either SL or SP, but some require even more power. While there are several ways those outliers could be handled, a natural approach from a linguistic perspective are Tier-based Strictly Local (TSL) grammars (Heinz et al. 2011). TSL grammars are SL grammars with a tier projection mechanism (cf. Goldsmith 1976 and Vergnaud 1977) that creates new locality relations. The SL grammar then restricts the arrangement of tier-adjacent segments rather than string-adjacent segments. Whether a segment ends up on the tier (a TSL grammar has only one tier) depends solely on whether its label is listed in the so-called *tier alphabet* — the structural surroundings of a segment are irrelevant.

A particularly clear demonstration of how TSL differs from SL and SP is primary word stress (Heinz 2014). It is a general property of phonological words that they have exactly one syllable carrying primary stress. Now consider a language where primary stress can be an unbounded number of syllables away from any word edge, e.g. because it sometimes falls on the rightmost heavy syllable, which may be close to the middle of the word. Under such circumstances, neither SL nor SP can guarantee the presence of stress.

Consider two words of the form $\sigma\sigma\cdots\sigma\acute{\sigma}\sigma\cdots\sigma\sigma$ and $\sigma\sigma\cdots\sigma\sigma\sigma\cdots\sigma$, where σ denotes unstressed syllables and $\acute{\sigma}$ stressed ones. The first word obeys the requirement for exactly one primary stress, whereas the latter only contains unstressed syllables and thus violates it. The first word contains the bigrams $\rtimes\sigma$, $\sigma\sigma$, $\sigma\acute{\sigma}$, $\acute{\sigma}\sigma$, and $\sigma\ltimes$. The second word only contains $\rtimes\sigma$, $\sigma\sigma$, and $\sigma\ltimes$. But note that all these bigrams are also part of the first string. Since an SL-2 grammar can only forbid illicit strings by forbidding certain bigrams, the string without primary stress can be marked as ungrammatical by banning at least one of its bigrams. But then the well-formed string with exactly one primary stress would be ruled out, too. For the very same reason, the grammar cannot generate $\sigma\sigma\cdots\sigma\acute{\sigma}\sigma\cdots\sigma\sigma$ without also generating $\sigma\sigma\cdots\sigma\acute{\sigma}\sigma\cdots\sigma\acute{\sigma}\sigma\cdots\sigma\sigma$. These counterexamples generalize to arbitrary SL-k grammars, showing the insufficiency of SL grammars for primary stress.

SP grammars improve on this as they can rule out words with two primary stresses. The strictly 2-piecewise grammar $\{\acute{\sigma}\acute{\sigma}\}$ prevents a stressed vowel from being followed by another stressed vowel, and thus no word can contain more than one primary stress. However, this captures only one half of the requirement that words have exactly one primary stress, it still remains to ensure that every word has at least one primary stress. SP grammars are incapable of doing that because all the subsequences of $\sigma\sigma\cdots\sigma\sigma\sigma\cdots\sigma\sigma$ are also subsequences of $\sigma\sigma\cdots\sigma\acute{\sigma}\sigma\cdots\sigma$. As in the case of SL, one cannot exclude the latter and still generate the former, and this holds for all SP-k grammars. Neither SL nor SP grammars, then, can enforce the presence of at least one primary stress.

A TSL grammar solves the challenge by positing a stress tier. This tier contains all segments, and only those, that carry primary stress (in other words, the tier alphabet consists of all possible instances of $\acute{\sigma}$). By convention the word edges are part of the tier, too. Now if a word contains no primary stress at all, the primary stress tier will be empty. If it contains at least two primary stresses, then the primary stress tier contains at least two segments. The only licit configuration on the tier is $\rtimes\acute{\sigma}\ltimes$ — a word with exactly one primary stress. These tier dependencies are SL-2 because we just have to forbid $\rtimes\ltimes$ (no primary stress) and $\acute{\sigma}\acute{\sigma}$ (more than one primary stress).

The example shows how TSL grammars are endowed with a mechanism for creating new locality domains so that non-local dependencies become local. These domains take the form of tiers, which ignore all segments that do not belong to a specified type T. The logical formulas for this example highlights this:

(3) a. $\forall x_1, x_2\big[x_1\ S_T\ x_2 \rightarrow \neg\big(\rtimes(x_1) \wedge \ltimes(x_2)\big) \wedge \neg\big(\acute{\sigma}(x_1) \wedge \acute{\sigma}(x_2)\big)\big]$

b. $x_1\ S_T\ x_2 \longleftrightarrow x_1 \prec x_2 \wedge T(x_1) \wedge T(x_2) \wedge \neg\exists z\big[x_1 \prec z \wedge z \prec x_2 \wedge T(z)\big]$

The formula in (3a) looks exactly like an SL formula, except that S is replaced by S_T, which is defined in (3b) to hold between two segments iff they are both projected onto the tier T and no segment between them is also present on this tier. Tiers thus replace S with S_T, a restricted version of the precedence relation $\prec$.

Adding a restricted version of $\prec$ to SL in order to obtain TSL increases expressivity of the system in a desirable way. Another approach, however, is to introduce local domains to SP. This is exactly the idea pursued in Sec. 3.

2.4 Relations Between the Three Classes

SL, TSL, and SP carve out different, but overlapping subregular regions.

SL and SP are incomparable. This means there exist patterns which belong to both classes, and some which only belong to one and not the other. TSL properly subsumes SL because any SL grammar can be modeled with a TSL grammar which puts every element on the tier.

Like SL, TSL is incomparable to SP. We already saw in the previous section that the restriction to exactly one primary stress per word is TSL but not SP. In the other direction, *circumambient processes* (Jardine 2016) constitute SP patterns that are not TSL. Circumambient processes have an interval-like nature: between two segments x and y, all z must have a specific shape.

A concrete instance is unbounded tone plateauing, which requires that no low tone L may occur between two high tones H, no matter how far apart those two high tones are. From the perspective of SP, this amounts to blocking HLH. TSL, on the other hand, cannot accommodate this dependency. Since both high tones and low tones are involved, both must be projected on the tier. There is no set of k-grams that can be used to forbid ill-formed sequences like $\mathrm{HL}^n\mathrm{H}$ for all $n > 1$. This is because sequences like HL^n and $\mathrm{L}^n\mathrm{H}$ are well-formed for every $n \geq 0$, including cases where $n \geq k$. But for such a large n, an SL-k grammar will also generate $\mathrm{HL}^n\mathrm{H}$ because no local constraint is violated — the two high tones are so far apart that the grammar cannot "see" the $\mathrm{H}\cdots\mathrm{H}$ configuration.

Although unbounded tone plateauing is SP, there is something unsatisfactory about the way SP accounts for it. Remember that the SP grammar simply rejects every string where L follows H and is itself followed by H, no matter how far apart the three are and what occurs between them. This constraint picks out the correct strings only if we assume a one-to-one relation between strings and phonological words. If strings consist of multiple words in some phrase then SP grammars cannot enforce this constraint. To see why, consider the string $\rtimes$\$HLL\$LLH\$$\ltimes$, where \$ is used for the word edge marker to clearly distinguish it from the string edge markers $\rtimes$ and $\ltimes$. This string does not violate tone plateauing thanks to the intervening word boundary, yet the SP grammar counts the string as ill-formed because it contains the subsequence HLH — the word boundary is not taken into account. Thus, unbounded tone plateauing is only SP if the dependencies are restricted to word domains.

The next section explores this point more carefully. I first show that arbitrary first-order domain restrictions on SP grammars are actually not restrictions at all. Then I show that once domains are restricted to *intervals* so that they more closely resemble phonological proposals, a more appealing picture emerges. A key result is that SL and TSL reduce to special cases of SP grammars operating on interval domains. Then, instead of an SL class for local processes, an SP class for non-local processes, and a TSL class for a little bit of both, there is one unified grammar that combines SP grammars with locality domains.

3 Adding Locality Domains

3.1 Domain-Based Strictly Piecewise Grammars

Let us now look at one way an SP grammar can be limited to a domain (3.1.1). We will see why this takes the expressivity of the resulting formalism beyond what appears to be needed for phonology (3.1.2).

3.1.1 Definition and Examples

Intuitively, confining an SP-k dependency between segments $x_1, x_2, \ldots, x_k$ to a specific locality domain D means that the dependency must be satisfied only if all x_i are inside D ($1 \leq i \leq k$). From the perspective of first-order logic, this is tantamount to making D the antecedent of an implication.

Definition 1 (Domain-Based SP)**.** *A* domain-based strictly k-piecewise *(DBSP-k) grammar is a tuple $G := \langle P, D \rangle$, where P is a strictly k-piecewise grammar and D is a formula of first-order logic (without any free variables).*[3] *We call D the* domain requirement of P *or simply the* domain. *A string s is generated by G iff the first-order formula below is true in s:*

$$\forall x_1, \ldots, x_k [D \rightarrow \bigwedge_{p \in P} \neg \phi_p]$$

Here ϕ_p denotes the first-order equivalent of the labeling information inherent in the strictly piecewise k-gram p (cf. Sec. 2.1). For instance, the bigram *ab* corresponds to the formula $a(x_1) \wedge b(x_2)$. The big *and* operator acts as a shorthand for multiple conjunctions. Assuming that the grammar contains only the bigrams $\rtimes a$, *aa*, and $a \ltimes$, the expanded formula would be:

$$\forall x_1, x_2 [D \rightarrow \neg(\rtimes(x_1) \wedge a(x_2)) \wedge \neg(a(x_1) \wedge a(x_2)) \wedge \neg(a(x_1) \wedge \ltimes(x_2))]$$

Due to the structure of the formula in Def. 1, the conditions enforced by the SP grammar need to be met by positions $x_1, \ldots, x_k$ only if the domain requirement D is satisfied. Note that D can be any first-order formula, so it need not even mention the variables $x_1, \ldots, x_k$. Furthermore, the formulas for SL, SP, and TSL are instances of the template above where D is a conjunction of statements of the form $x_i \; R \; x_j$, with R set to S, $\prec$, or S_T (cf. (2) and (3a)).

DBSP grammars solve the aforementioned problem of SP grammars with unbounded tone plateauing. The problem with the SP account in Sec. 2.4 is that it fails whenever the ban against HLH subsequences applies to domains that are larger than a single word. Consider the DBSP-3 grammar $G := \langle P, W \rangle$ where the SP grammar P only contains the trigram *HLH*. The domain restriction W guarantees that P only applies within words (I assume that word edges are explicitly marked by $):

$$W := \exists l, r\Big[l \prec x_1 \wedge \bigwedge_{1 \leq i < 3} x_i \prec x_{i+1} \wedge x_3 \prec r \wedge \$(l) \wedge \$(r) \wedge \neg \exists z[\$(z) \wedge l \prec z \wedge z \prec r]\Big]$$

This formula encodes a simple statement: x_1, x_2, and x_3 must occur between word edge markers l and r that are not separated by another word edge marker z. In other words, x_1, x_2, and x_3 must belong to the smallest interval spanned by two $ markers. This is the first-order expression for a word domain. The DBSP grammar $\forall x_1, x_2, x_3[W \rightarrow \neg\phi_{HLH}]$ then says that the three positions are subject to the requirement enforced by the trigram *HLH* only if condition W holds.

The full first-order sentence for G with W and ϕ_{HLH} expanded is given here to illustrate how the different pieces of the grammar are combined into a single formula:

$$\begin{aligned} &\forall x_1, x_2, x_3\Big[\\ &\exists l, r\Big[l \prec x_1 \wedge \bigwedge_{1 \leq i < 3} x_i \prec x_{i+1} \wedge x_3 \prec r \wedge \$(l) \wedge \$(r) \wedge \neg \exists z[\$(z) \wedge l \prec z \wedge z \prec r]\Big] \rightarrow \\ &\qquad\qquad\qquad\qquad \neg\Big(H(x_1) \wedge L(x_2) \wedge H(x_3)\Big)\Big] \end{aligned}$$

In this way DBSP grammars can require constraints to hold only within certain domains.[4]

3.1.2 Overgeneration: DBSP = Star-Free

While the ability of DBSP to handle unbounded tone plateauing is welcome, it comes at the price of massive overgeneration: the class of DBSP languages is exactly the class of *star-free* languages. A language is star-free iff it can be defined in first-order logic (McNaughton and Papert 1971). While star-free-ness may be a necessary condition on phonotactic patterns, they are arguably not a necessary one because phonologically outrageous dependencies can be expressed with first-order logic as shown in (4).

(4) Take any a string that contains both a vowel V as its last segment and a sibilant S such that either S is word-initial or the string contains exactly seven consonants. Then intervocalic voicing is enforced iff the voicing value of S (+/−) is the opposite of V's value for round plus S's value for anteriority (where $x + y = -$ iff $x \neq y$).

Consequently, the equivalence of DBSP and the star-free languages entails that locality domains allow for the generation of very unnatural patterns.

However, an even stronger result holds: every star-free language is DBSP-1. So no matter how much one constrains the SP component of a DBSP grammar, the domain restrictions can always make up for any loss of expressivity obtained by weakening the SP grammar.

Theorem 1. *A string language L is* DBSP-1 *iff L is star-free.*

The proof of this equivalence exploits the permissive definition of domain restrictions. Recall from Sec. 3.1.1 that a DBSP grammar only requires domain restrictions to be first-order formulas. The domain restriction D is then inserted into a general template $\forall x_1, \ldots, x_k[D \rightarrow \bigwedge_{p \in P} \neg \phi_p]$. But this procedure is so general that D can be used to impose arbitrary constraints as long as they are first-order definable. And the star-free languages are exactly those that can be defined by arbitrary first-order definable constraints.

Proof. That DBSP $\subseteq$ Star-Free is a corollary of all DBSP languages being first-order definable. So we only need to show that Star-Free $\subseteq$ DBSP.

Let L be some star-free language. Then there is some first-order formula ϕ_L (without free variables) such that all and only strings in L satisfy ϕ_L. Now it holds for any given string w that it satisfies ϕ_L iff it satisfies $\forall x_1[\neg\phi_L \rightarrow \bot]$, where $\bot$ is a shorthand for some formula that is always false. This bi-implication holds because I) ϕ_L contains no free variables that can be bound by $\forall x_1$, and II) by the definition of implication, a proposition p is true iff $\neg p \rightarrow \bot$ is true. Observe in addition that the SP-1 grammar containing only the unigram $\rtimes$ generates the empty language, which means that no string satisfies $\forall x_1[\neg \rtimes (x_1)]$. Putting all of this together, we conclude that L is generated by the DBSP-1 grammar $G := \langle \neg\phi_L, \{\rtimes\} \rangle$. Since L was arbitrary, every star-free language is DBSP-1. $\square$

Even though arbitrary domain restrictions grant SP grammars too much power, this does not force us to forgo them completely — instead, we should tighten our notion of what domain restrictions may look like.

3.2 Interval-Based Strictly Piecewise Grammars

While mathematically correct, the proof of Thm. 1 seems to violate the spirit of what phonological locality domains are about. Instead of picking out certain substructures within which a constraint applies, the domain restriction acts as the constraint while the SP grammar is trivialized to a nonsensical ban against left word edges. Moreover, it is certainly not the case that any first-order formula is a possible phonological domain. Looking once more at our treatment of tone plateauing in Sec. 3.1.1, our primary interest is not to define tightly interwoven conditionals as in the outrageous example (4), but rather to delineate specific *intervals* within a string — and this intuition is also the foundation of phonological notions like feet, syllables, words and sentences, and so on. As I show next, a proper formalization of intervals (3.2.1) does indeed go a long way to fix the overgeneration problems of DBSP (3.2.2).

3.2.1 Introducing k-Intervals

SP constraints which apply within phonologically motivated intervals must specify three items (cf. Jensen 1974; Odden 1994). First, the edges of the interval must be specified, e.g. that the left and right edges are labeled \$ in the domain W presented earlier. Second, the interval must contain a given number of positions, which are subject to the constraints of the SP grammar. So an interval for an SP-k grammar contains k such positions, denoted by $x_1, \ldots, x_k$ in the first order formulas. Finally, there is an additional stipulation on what the remainder of the interval may look like. All these requirements can be represented by an extension of k-grams that I call *k-vals* (short for k-interval).

The advantage of k-vals is that they provide a much more readable specification of complex first-order formulas such as W. Consider once more the DBSP grammar $\langle P, W \rangle$, which forbids HLH subsequences within individual words. We can depict W as a 3-val that requires the edges to be labeled \$, contains three open slots that are subject to the SP grammar, and ensures that no other material within this interval (the *fillers*) is labeled \$.

(5) **3-val for W**

These k-vals are more constrained than arbitrary first-order formulas as we may only specify the number and shape of open slots, the labels of the edges, and what the other positions in the interval may look like. Nonetheless k-vals are still remarkably flexible. In fact, they are so flexible that SP grammars with

k-vals as domain restrictions subsume SL and TSL grammars.

Every SL-k grammar can be reinterpreted as an SP grammar with a k-val that forbids any fillers. Assuming that every node must be labeled l_1 or l_2 or l_3, and so on, this is tantamount to the demand that no filler is labeled l_1 or l_2 or l_3, and so on. Take as a concrete example the SL-2 grammar from (1), which generates words with a CV syllable template over the sound inventory a, u, p, k. An SP grammar with exactly the same bigrams describes the very same CV syllable template if it is coupled with the 2-val in (6).

(6) **2-val for emulating an SL-2 grammar**

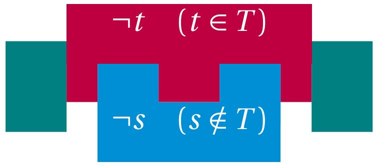
This 2-val can only match intervals where the two empty open slots are adjacent to each other. As a result, the long-distance dependencies of the SP-2 grammar are limited to adjacent segments, making it behave exactly like an SL-2 grammar.

So far our k-vals have displayed a certain asymmetry in that fillers are always restricted, edges are sometimes restricted, and empty slots are never restricted. Eliminating these asymmetries allows us to also emulate standard SP grammars and, more interestingly, TSL grammars. The inclusion of SP is straight-forward: if the k-val puts no restrictions on any segments, then SP grammars operate uninhibited.

(7) **3-val for emulating an SP-3 grammar**

TSL arises from a k-val format that is only slightly more complicated than that for SL. In the case of SL, we wanted to prevent any segments from intervening between open slots. For TSL, we allow fillers as long as they are not part of the tier alphabet T. On the flip side, the open slots are restricted to members of the tier alphabet — or the other way round, they must not be segments that are not members of the tier alphabet.

(8) **2-val for emulating a TSL-2 grammar**



This largely exhausts the range of options for k-vals. For the sake of generality I also allow the left and right edge to be associated with different properties so that an interval may span from, say, a vowel to a specific consonant. But this minor extension does not change the general fact that k-vals are extremely impoverished in comparison to arbitrary first-order formulas. In fact, each component of a k-val amounts to a ban against specific labels. Such bans are SL-1 (or equivalently, SP-1), which is the weakest class in the whole subregular hierarchy. So k-vals define locality domains by a combination of the weakest conceivable grammars, yet when they are added on top of SP grammars they are surprisingly expressive and subsume SP, SL, and TSL while also handling circumambient processes correctly.

3.2.2 Formal Definition and Generative Capacity

With the intuition firmly established, the formal definition of *interval-based SP-k* (IBSP-k) should be much easier to grasp, and it will also be straight-forward to prove that there are DBSP languages that are not IBSP. This strongly suggests that IBSP currently offers the best solution to integrate domain restrictions with phonotactic patterns: it subsumes the phonologically natural classes SL, SP, and TSL, and it also captures circumambient dependencies, yet it remains much more restrictive than DBSP.

I first define k-vals as a combination of four SL-1 grammars,two of which restrict the edges, one the open slots, and one the fillers. Each k-val corresponds to a first-order formula.

Definition 2 (k-val)**.** *A* k-interval, *or simply* k-val, *is a* 5*-tuple* $\langle k, G_l, G_r, G_o, G_f \rangle$ *such that* $k \geq 1$ *and each* G *is an SL-*1 *grammar that specifies, respectively, the left edge, right edge, open slots, and the fillers. The corresponding* k-val formula *(with free variables* $x_1, \ldots, x_k$*) is*

$$\exists l, r \Big[\bigwedge_{g \in G_l} \neg g(l) \wedge \bigwedge_{g \in G_r} \neg g(r) \wedge \bigwedge_{1 \leq i \leq k} \bigwedge_{g \in G_o} \neg g(x_i) \wedge l \prec x_1 \wedge x_k \prec r \wedge$$
$$\forall z \Big[\big((l \prec z \wedge z \prec x_1) \vee (x_k \prec z \wedge z \prec r) \vee \bigvee_{1 \leq i < k} (x_i \prec z \wedge z \prec x_{i+1}) \big) \rightarrow \bigwedge_{g \in G_f} \neg g(z) \Big] \Big]$$

For any empty G, $\bigwedge_{g \in G} \neg g(x)$ is always satisfied.

The overly powerful class DBSP is pruned down to the linguistically more plausible class *interval-based SP* by limiting the domain restrictions to k-val formulas.

Definition 3. *A DBSP-k grammar is* interval-based SP-k *(IBSP-k) iff its domain restriction D is a k-val formula.*

Thanks to our previous observations in (6)–(8) we already know that IBSP contains all languages that are SL, SP, or TSL. But IBSP also goes slightly beyond those classes because it can still express the domain restriction W and thus accounts for circumambient processes in a straight-forward manner.

Theorem 2. IBSP $\supsetneq$ SL $\cup$ SP $\cup$ TSL

Proof. It suffices to show that SL, SP, and TSL are each subsets of IBSP. That inclusion is proper is witnessed by unbounded tone plateauing in a string with multiple words.

SP $\subseteq$ IBSP: For empty G_l, G_r, G_o, and G_f, the corresponding k-val formula reduces to $\exists l, r[l \prec x_1 \wedge x_k \prec r]$. This domain formula D is always satisfied because we may freely assume the presence of a sufficient number of string edge markers $\rtimes$ and $\ltimes$ to satisfy the existence of l and r. So $D \rightarrow \bigwedge_{p \in P} \neg \phi_p$ equals $1 \rightarrow \bigwedge_{p \in P} \neg \phi_p$, which is equivalent to $\bigwedge_{p \in P} \neg \phi_p$. As P may be any arbitrary SP grammar, every SP grammar has an equivalent IBSP grammar.

SL $\subseteq$ IBSP: It suffices to show that k-vals allow us to define S in terms of $\prec$. Let Σ be some fixed alphabet of segments. Suppose G_l, G_r and G_o are empty, but $G_f := \Sigma$. Then the $\forall z[\ldots \rightarrow \bigwedge_{g \in G_f} \neg g(z)]$ part of the k-val formula is false whenever some z satisfies the antecedent of the implication, wherefore the k-val formula holds only if no such z exists. But for any x and y, $x \prec y \wedge \neg \exists z[x \prec z \wedge z \prec y]$ holds iff $x \ S \ y$. Thus the k-val formula reduces to $\exists l, r[l \ S \ x_1 \wedge \bigwedge_{1 \leq i < k} x_i \ S \ x_{i+1} \wedge x_k \ S \ r]$. Again we may freely assume the existence of l and r, so that D is simply $\bigwedge_{1 \leq i < k} x_i \ S \ x_{i+1}$. But then $\forall x_1 \ldots x_k[D \rightarrow \bigwedge_{p \in P} \neg \phi_p]$ is the formula of some SL-k grammar P. As P is arbitrary, every SL grammar has an equivalent IBSP grammar.

TSL $\subseteq$ IBSP: Let Σ be some fixed alphabet of segments and $T \subseteq \Sigma$ our tier alphabet. Suppose G_l and G_r are empty, $G_o := \Sigma - T$, and $G_f := T$. Varying the proof for SL, we see that $x \prec y \wedge \forall z[x \prec z \wedge z \prec y \rightarrow \bigwedge_{g \in G_f} \neg g(z)]$ iff $x \prec_T y$. Again l and r can be assumed freely, so that $\forall x_1 \ldots x_k[D \rightarrow \bigwedge_{p \in P}]$ is the formula of some arbitrary TSL-k grammar P. □

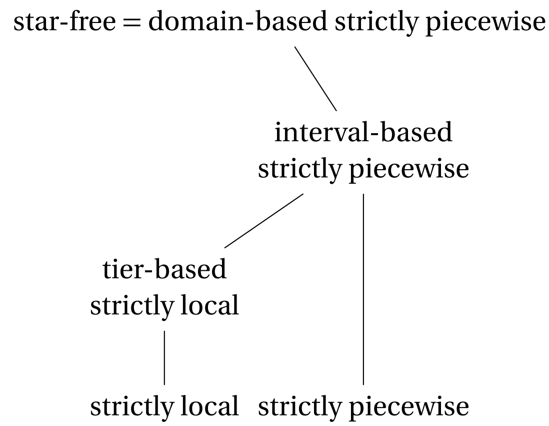


Figure 1: Proper inclusion relations between the subregular classes discussed in this paper

In fact, the existence of circumambient patterns such as tone plateauing is now completely expected because it merely involves co-opting mechanisms of the grammar that are already employed to restrict SP dependencies to the word domain, capture SL and TSL patterns, and so on. And these mechanisms are remarkably weak: k-vals define locality domains with a combination of SL-1 grammars, which define the arguably weakest class of formal languages in the subregular hierarchy. Yet when these intervals restrict SP constraints they furnish a surprising degree of expressivity.

While more expressive than these other subregular grammars, IBSP grammars are still incapable of generating many unattested patterns. This can be inferred from the fact that they are strictly weaker than DBSP grammars.

Theorem 3. DBSP $\supsetneq$ IBSP

Proof. That IBSP is included in DBSP follows immediately from its definition. It remains to show that there are DBSP patterns that are not IBSP. This is witnessed by the language $L := \$sa^*s \cup za^*z\$$, where segments at the word edges must be either s or z, whereas all other segments are a. This language is first-order definable and thus DBSP, but it is not IBSP. In order to rule out strings of the form $\$a^*sa^*sa^*\$$, an IBSP grammar needs a k-val with $G_l = G_r := \{s, z, a\}$ and $G_f := \{\$, s, z, a\}$. But we already saw in the previous proof that IBSP reduces to SL if $G_f := \Sigma$, and L is not SL. $\square$

The DBSP language used in the proof is a formal counterpart to the unattested process of first-last harmony (see Lai 2015 for details). IBSP fails to capture first-last harmony because it cannot integrate the local dependency of word edge adjacency with the non-local harmony constraint. It is known that first-last harmony is locally testable, a generalization of SL. So first-last harmony is still a fairly simple pattern, but the limitations of IBSP explain why it nonetheless does not seem to arise in phonology.

Its inability to integrate local with non-local information also ensures that IBSP captures the old adage that "languages do not count". Consider the language L that contains only strings with at most one instance of an. For an IBSP account, we might try to use the word domain W and forbid anan. But this will also rule out strings where a and n are not adjacent. If we forbid fillers in order to ensure adjacency, then only adjacent instances of an are disallowed. So a word may contain multiple instances of an as long as they are far away from each other. The same problem arises if we take a or n to be one of the word edges: as soon as adjacency is enforced, we can no longer enforce the long-distance dependency of at most one an in the whole word domain. A similar argument can be used to show that the variant of L that contains at least two instances of an is not IBSP. This kind of arbitrary counting is impossible with IBSP. Where counting effects seem to obtain, as with primary stress assignment, they have to be so simple that they can be captured via purely structural means, e.g. tier projection.

As a hypothesis about the computational nature of phonotactics, then, IBSP predicts that there should be no attested patterns that integrate and local and long-distance dependencies in the manner suggested by the previous examples. This does not seem to be entirely true; at least one counterexample is discussed in Heinz (2007) and McMullin (2016). McMullin (2016) explains that sibilant harmony in Samala, which was briefly discussed in Sec. 2.2 as an example of an SP pattern, can overrule a general ban against string adjacent sn, sl, and st. That is to say, sequences like sn are forbidden unless there is another s somewhere to the right of n. De Santo and Graf (2017) take this pattern as the motivation to enhance TSL grammars with a more powerful, structurally conditioned tier projection mechanism. While these grammars correctly capture this interaction between a local and a non-local dependency, IBSP is once again incapable of doing so. The reason is exactly the same as before: in order to enforce adjacency for some segments in the locality domain, we have to enforce adjacency for all segments, which makes it impossible to handle the long-distance dependency correctly.

To sum up, IBSP is incomparable with many subregular language classes and thus less expressive than DBSP. For the most part the limited expressivity of IBSP is a good thing, but a minor extension will be needed in the future to handle cases like the example from Samala where local and non-local information interact. This also includes several suprasegmental phenomena such as tone spreading in Copperbelt Bemba (Jardine 2016) and certain stress patterns (Baek 2017).[5] Nonetheless IBSP grammars offer an insightful and empirically viable unification of all phonologically interesting subregular classes posited so far — a unification that is grounded in locality domains, one of the most natural of linguistic concepts.

3.3 Learnability

The large gap in expressivity between DBSP and IBSP highlights the restricted nature of locality domains in phonology, but it is also an essential factor for learnability. It is a well-known fact that the star-free language are not learnable in the limit from positive text (a corollary of them properly subsuming the class of finite languages, Gold 1967). Consequently, the expressively equivalent class DBSP cannot be learned in the limit from positive text, either.

However, IBSP-k is learnable for any fixed choice of k. This learnability claim is trivial once one realizes that there can be only finitely many distinct languages in this class for any choice of k. There are only finitely many SP-k grammars because with an alphabet of size n there are n^k distinct k-grams and thus $2^{(n^k)}$ distinct grammars. The number of distinct k-vals is similarly limited to $(2^n)^4 = 2^{4n}$ because a k-val consists of four unigram grammars. As a result, the number of IBSP-k languages is at most $2^{(n^k)} \cdot 2^{4n}$. While this number is enormous, Gold (1967) proved that every finite language class is learnable, no matter how large it is.

Theorem 4. *IBSP-k is learnable in the limit from positive text.*

This still leaves open whether IBSP-k is efficiently learnable from just a small amount of data. I suspect that this strong condition holds mainly because IBSP treats locality domains as combinations of SL-1 dependencies, which are efficiently learnable in isolation. More generally SL-k and SP-k are efficiently learnable as well, and so is even TSL (Jardine and Heinz 2016; Jardine and McMullin 2016). The methods there may be applicable to this case as well. Finally, I also expect that phonologically motivated naturalness conditions (clustering segments into classes, minimum prominence demands for interval edges, and so on) greatly prune down the number of possible k-vals, tremendously reducing the amount of required data.

Even if my conjecture turns out to be false, the fact that IBSP is learnable whereas DBSP is not confirms the long-held belief that UG-restrictions on locality domains aid learnability (Heinz 2007, 2009, 2010). A child that comes already equipped with the assumption that dependencies involve k elements and may only be relativized to locality domains of a specific shape can successfully learn any language in the hypothesis space.

Conclusion

The main insight of this paper is that locality domains simplify our current picture of subregular phonology rather than complicate it. IBSP grammars unify the SL, TSL and SP classes by amplifying SP grammars with interval-based domain restrictions. There are a number of welcome consequences to introducing these locality domains:

(9)
a. Overgeneration is largely curbed by ruling out unnatural dependencies involving counting or elaborate conditionals.
b. Undergeneration is mostly avoided since all strictly piecewise, strictly local, and tier-based strictly local languages can still be generated.
c. Circumambient patterns such as tone plateauing arise in a natural fashion from the ability to define domain restrictions.
d. Many blocking effects can be accommodated, too.
e. The internal structure of the language space is still sufficiently limited to allow for learning in the limit from positive text.

This paper constitutes but a first step towards a more thorough exploration of how locality domains may be effectively incorporated into the study of subregular patterns in language. Formally, establishing an abstract characterization of the IBSP class will help in identifying phonotactic patterns that do and do not belong to this class. Empirically, since the domains within which phonological processes apply have been studied by linguists for many years, attempts to further generalize and refine these findings can draw from numerous ideas in the literature. Careful exploration of the formal ramifications of such modifications should yield an even tighter characterization of natural language phonotactics and might also prove fruitful in other areas such as syntax and morphology.[6] It will also be interesting to see whether the expressivity gap between segmental and suprasegmental phonology noted by Jardine (2016) can now be derived from linguistic assumptions about their respective locality domains.

Notes

[1]Alternatively, our representations could operate with feature bundles instead of segments, so that V and C are abbreviations of [−cons] and [+cons], respectively.

[2]The definition of SP grammars in Rogers et al. (2010) also stipulates that no k-gram of an SP grammar may contain any edge markers. While this has mathematical advantages, I do not adopt this additional restriction in an effort to maintain the parallel between SL and SP.

[3]The symbol := is commonly used in mathematics to indicate an assignment of a specific value to a named variable. It should not be confused with =, which denotes equality of two distinct symbols or objects.

[4]This is not the only DBSP grammar that can account for unbounded tone plateauing over domains larger than a word. Generally, it is possible to obtain a different description of tone plateauing that trades a slightly more complicated domain restrictor for a simpler SP grammar. For instance, instead of blocking HLH subsequences within words, a DBSP grammar can be written which bans any instance of L between two high tones that are not separated by a word edge or another high tone. The details are left to the reader.

[5] Another empirically relevant shortcoming of IBSP is its inability to enforce multiple unrelated dependencies in parallel. For instance, it is impossible for a single IBSP grammar to ensure that a word contains exactly one primary stress and obeys sibilant harmony. The limitation to the word domain means that the edges are limited to $, so sibilants and segments with primary stress must both go in the open slots. But then two segments with primary stress may be separated by an unbounded number of sibilants, which makes it impossible to ensure that at least one primary stress is present in the word. Therefore each process requires its own IBSP grammar. A more powerful class of multi-IBSP that handles all those dependencies in parallel would be easy to define as it simply amounts to the intersection closure of IBSP, which is still properly contained by DBSP.

[6]For example, Graf and Heinz (2015) argue that syntax can be regarded as tier-based strictly local at a sufficiently high level of abstraction. This finding relies on simplifying assumptions about movement, though, that are slightly at odds with the current thinking in Minimalist syntax. Interval-based strictly piecewise grammars may offer enough headroom to capture syntactic dependencies in a more general setting.

References

Applegate, Richard B (1972). *Ineseño Chumash grammar*. Doctoral Dissertation, University of California, Berkeley.

Baek, Hyunah (2017). Computational representation of unbounded stress: Tiers with structural features. Ms., Stony Brook University.

Booij, Geert, and Jerzy Rubach (1984). Morphological and prosodic domains in lexical phonology. *Phonology Yearbook* 1. 1–27.

Chandlee, Jane (2014). *Strictly local phonological processes*. Doctoral Dissertation, University of Delaware.

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

Chandlee, Jane, Rémi Eyraud, and Jeffrey Heinz (2015). Output strictly local functions. In *Proceedings of the 14th Meeting on the Mathematics of Language (MoL 2015)*, ed. Marco Kuhlmann, Makoto Kanazawa, and Gregory M. Kobele. 112–125. Chicago, USA.

De Santo, Aniello, and Thomas Graf (2017). Structure sensitive tier projection: Applications and formal properties. Ms., Stony Brook University.

Ebbinghaus, Heinz-Dieter, Jörg Flum, and Wolfgang Thomas (1996). *Mathematical logic*. Springer.

Enderton, Herbert B (2001). *A mathematical introduction to logic*. Academic Press, 2nd edition.

Gold, E. Mark (1967). Language identification in the limit. *Information and Control* 10. 447–474.

Goldsmith, John (1976). *Autosegmental phonology*. Doctoral Dissertation, MIT.

Graf, Thomas (2010a). Comparing incomparable frameworks: A model theoretic approach to phonology. *University of Pennsylvania Working Papers in Linguistics* 16:Article 10.

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

Graf, Thomas, and Jeffrey Heinz (2015). Commonality in disparity: The computational view of syntax and phonology. Slides of a talk given at GLOW 2015, April 18, Paris, France.

Hayes, Bruce (1995). *Metrical stress theory*. Chicago: Chicago University Press.

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 (2010). Learning long-distance phonotactics. *Linguistic Inquiry* 41. 623–661.

Heinz, Jeffrey (2014). Culminativity times harmony equals unbounded stress. In *Word stress: Theoretical and typological issues*, ed. Harry van der Hulst. 255–275. Cambridge, UK: Cambridge University Press.

Heinz, Jeffrey (2015). The computational nature of phonological generalizations. Ms., University of Delaware.

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

Jardine, Adam (2016). Computationally, tone is different. *Phonology* 33. 247–283.

Jardine, Adam, and Jeffrey Heinz (2016). Learning tier-based strictly 2-local languages. *Transactions of the Association for Computational Linguistics* 4. 87–98.

Jardine, Adam, and Kevin McMullin (2016). Efficient learning of tier-based strictly k-local languages. Ms., Rutgers University and University of Ottawa.

Jensen, John (1974). Variables in phonology. *Language* 50. 675–686.

Johnson, C. Douglas (1972). *Formal aspects of phonological description*. The Hague: Mouton.

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

Kaye, Jonathan (1995). Derivations and interfaces. In *Frontiers of phonology*, ed. Jacques Duran and Francis Katamba. 289–332. London: Longman.

Kiparsky, Paul (1982). From cyclic to lexical phonology. In *The structure of phonological representations*, ed. Harry van der Hulst and Norval Smith, volume 1. 131–175. Dordrecht: Foris.

Lai, Regine (2015). Learnable vs. unlearnable harmony patterns. *Linguistic Inquiry* 46. 425–451.

McMullin, Kevin (2016). *Tier-based locality in long-distance phonotactics: Learnability and typology*. Doctoral Dissertation, Uniersity of British Columbia.

McNaughton, Robert, and Seymour Papert (1971). *Counter-free automata*. Cambridge, MA: MIT Press.

Nespor, Marina, and Irene Vogel (1986). *Prosodic phonology*. Dordrecht: Foris.

Odden, David (1994). Adjacency parameters in phonology. *Language* 70. 289–330.

Rogers, James, Jeffrey Heinz, Gil Bailey, Matt Edlefsen, Molly Vischer, David Wellcome, and Sean Wibel (2010). On languages piecewise testable in the strict sense. In *The mathematics of language*, ed. Christan Ebert, Gerhard Jäger, and Jens Michaelis, volume 6149 of *Lecture Notes in Artificial Intelligence*. 255–265. Heidelberg: Springer.

Rogers, James, Jeffrey Heinz, Margaret Fero, Jeremy Hurst, Dakotah Lambert, and Sean Wibel (2013). Cognitive and sub-regular complexity. In *Formal Grammar*, ed. Glyn Morrill and Mark-Jan Nederhof, volume 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.

Rubach, Jerzy (2008). An overview of Lexical Phonology. *Language and Linguistics Compass* 2. 456–477.

Scheer, Tobias (2012). Chunk definition in phonology: Procosodic constituency vs. phrase structure. In *Modules and interfaces*, ed. M. Bloch-Trojnar and A. Bloch-Rozmej. 221–253. Lublin: Wydawnictwo KUL.

Selkirk, Elisabeth (1984). *Phonology and syntax: The relation between sound and structure*. Cambridge, Mass.: MIT Press.

Selkirk, Elisabeth (2011). The syntax-phonology interface. In *The handbook of phonological theory*, ed. John Goldsmith, Jason Riggle, and Alan Yu. Oxford: Blackwell, 2nd edition.

Shwayder, Kobey (2015). *Words and subwords: Phonology in a piece-based syntactic morphology*. Doctoral Dissertation, University of Pennsylvania.

Vergnaud, Jean-Roger (1977). Formal properties of phonological rules. In *Basic problems in methodology and linguistics*, ed. Robert E. Butts and Jaakko Hintikka. 299–318. Dordrecht: Reidel.