Learning local and non-local interactions in tone DRAFT

Adam Jardine
adam.jardine@rutgers.edu

March 19, 2018

Abstract

This paper is the first investigation into the theoretical learnability of tone. It posits a learnable class of *melody-local* grammars that captures the range of local tone patterns, long-distance tone-patterns, and their interactions. This is achieved through a restriction on autosegmental representations to constraints that either operate on a local level or over a 'melody', which captures long-distance patterns. It is shown how this class is learnable from positive data by a provably correct, efficient algorithm, and how melody-local grammars and melody-local learning can capture tone patterns that cannot be learned by tier-projection learners.

1 Introduction

This paper provides the first investigation into the learnability of tone. The extraordinary range of tonal patterns is well-documented (Yip, 2002; Kisseberth and Odden, 2003; Hyman, 2011b; Jardine, 2016). Particularly striking is the variety of long-distance generalizations exhibited in tone. For example, in phrase-final position in Bemba (Bickmore and Kula, 2013), a final high (H) tone spreads to the end of the word, whereas any preceding H tones spread to only one additional tone-bearing unit (TBU). In (1) and throughout this paper, an acute accent indicates a H toned mora; all other mora are low (L). Schematic representations of the forms as strings of H and L TBUs are given to the right.

(1) Bemba (Bickmore and Kula, 2013)

a.	tu-léé-pát-á	'we are hating'	LHHHH
b.	bá-ká-fík-á	'they will arrive'	HHHH
c.	tú-lúb-ul-ulé	'we should explain '	HHLLH
d.	twáá-ku-láá-pá	'we will be drawing (water)'	HHLHHH
e.	*tú-lúb-úl-ulé		*HHHLH
f.	*bá-ká-fik-a		*HHLL

This is a long-distance pattern because unbounded spreading of a H tone is blocked by another, following H tone. For example, in (1b), [bá-ká-fík-á] 'they will arrive' HHHH, a single span of H-tones extends from the beginning of the word to the end of the word, whereas in (1c) [tú-lúb-ul-ulé] 'we should explain' HHLLH, the initial span of H tones is only two TBUs long, because another H tone follows (on the final [é]). Thus, a form like (1e) *[tú-lúb-úl-ulé] HHHLH is ill-formed, because a non-final H has spread over more than TBUs. Conversely, any H tone *not* followed by another H tone must spread to the end of the word; thus (1e) *[bá-ká-fik-a] HHLL is ill-formed. Thus, how much a H tone spreads is dependent on the presence or absence of a following H tone, no matter how far to the right.

Long-distance generalizations like this pose a problem for learning, because a learner must discover dependencies that may hold over a potentially unbounded distance. Distance learning of phonotactic patterns has either made reference to precedence (Heinz, 2010a) or tier-projection (Hayes and Wilson, 2008; Goldsmith and Riggle, 2012; Jardine and Heinz, 2016; McMullin and Hansson, 2015; Jardine and McMullin, 2017; Gallagher and Wilson, to appear).

Expanding on Jardine (to appear), this paper shows that neither precedence nor tier projection can learn the full range of tone patterns, and patterns which show a combination of local and non-local dependencies, like in Bemba, cannot be learned with either. This paper solves this problem by positing a novel class of formal grammars and corresponding learning mechanism that simulatenously keeps track of local constraints and *spans* of tones using a melody tier like that of autosegmental representations. These grammars posit a novel restriction on well-formedness constraints in tone: that they can refer to local configurations *or* long-distance, melody configurations, but never both at the same time. This provides a restrictive theory of tonal well-formedness that explains the typology through how the learner generalizes from positive data.

1.1 The approach

This paper views phonology from the perspective of formal language theory, the formal study of patterns, and grammatical inference, which is concerned with the inductive learning of patterns from finite sets of examples. Formal language-theory provides precise constraint description languages for describing patterns with statements that are categorically true of each member of the pattern.[1] For example, if we think of the Bemba pattern in terms of strings of H- and L-toned TBUs, the following statement is true of any string in the pattern.

(2) *HHHL
A sequence of three H-toned moras followed by a L-toned mora is forbidden.

The constraint in (2) refers to a situation in which a non-final H has spread more than two TBUs, which is ungrammatical in Bemba and thus does not appear in any surface string of TBUs in the language. It thus partially describes the pattern. A formal language-theoretic description of a pattern is thus a statement or series of statements like (2) that are true for all and only the strings in the pattern.

Such statements are different than markedness constraints in Optimality Theory (Prince and Smolensky, 1993, 2004) in that they are inviolable. However, the value in formal language-theoretic descriptions is that they tell us about the structural properties of the pattern. For example, the statement in (2) is a *strictly local* constraint (McNaughton and Papert, 1971; Rogers and Pullum, 2011), because it refers only to a sequence of adjacent units in a string. Patterns describable with strictly local constraints are efficiently learnable, because a learner only has to keep track of sequences of adjacent units (García et al., 1990). Formal language theory can thus be fruitfully applied to phonology because we can ask:

1) What structural properties do phonological patterns share?
2) How do these structural properties allow a learner to navigate the hypothesis space?

[1]It is also possible to study patterns and their inference from a probabilistic perspective (Rabin, 1963; de la Higuera, 2010), though this often builds on results from the categorical perspective. How to incorporate the probabilistic perspective will be discussed at appropriate points throughout the paper.

By characterizing phonological patterns in terms of their structural properties, the formal language-theoretic perspective allows us to posit theories of phonology that meaningfully distinguish between attested and unattested patterns based on the kind of information that humans use when acquiring phonological patterns (Heinz, 2009, 2010a; Heinz et al., 2011; Rogers et al., 2013; Lai, 2015; McMullin and Hansson, 2015).

1.2 The proposal

Formal language-theoretic characterizations of patterns also give us a precise understanding of when a pattern does *not* have a property. For example, while the Bemba pattern is partially describable by the strictly local constraint in (2), the pattern itself as a whole is not strictly local. As already noted, the pattern requires distinguishing between H-tone spans for which another H follows, which must be bounded, and H-tone spans not followed by another H, which cannot be bounded.

(3) a. HHLLH b. *HHLLL
HHLLLH *HHLLLL
HHLLLLH *HHLLLLL
... ...

As to be discussed in more detail below, this cannot be described by a strictly local constraint. Roughly this is because any number of L-toned TBUs can intervene between two H-tone spans, and so we cannot describe the constraint in terms of sequences of adjacent TBUs. Another value of the formal language-theoretic perspective is that we can *prove* this about the Bemba pattern, and thus also that a strictly local learner is *guaranteed* to fail to learn the pattern, no matter how much data it is exposed to.

However, if we look at the autosegmental representations (ARs) of these forms, we can capture this constraint in terms of sequences of adjacent tones on the melody. Autosegmental representations of (1b), (c), and (f) are given in (4).

(4) a. #H #
μ μ μ μ
[bá-ká-fík-á]
'they will arive'

b. #H L H#
μ μ μ μ μ
*[tú-lúb-ul-ulé]
'we should explain

c. * #H L #
μ μ μ μ
*[bá-ká-fik-a]

For the well-formed words in (4a) and (b), the melody tier in the AR ends in a H# sequence. In contrast, the ill-formed word in (4c) contains a HL# sequence in the melody—this indicates a that a L tone occupies at least one TBU before the end of the word, and thus that the last H in the word has not spread to the final TBU. Thus, to ban the set of forms in (3c) in which the final H tone has not spread to the end of the word, we can posit the following statement about the ARs of Bemba forms.

(5) *HL# (melody tier)

A HL sequence of tones at the end of the word is forbidden in the melody tier.

This takes the form of a strictly local constraint, but it necessarily holds over the melody tier of an AR. Thus, a strictly local constraint over the melody tier captures the long-distance generalization that the last H in the word must spread to the final TBU. As illustrated in §3.3, strictly local constraints over AR melodies readily capture the kinds of long-distance patterns attested in tone.

However, the Bemba pattern cannot entirely be captured by a long-distance constraint over the melody: as described in (2), there is also strictly local constraint holding over the original string of TBUs that says a non-final H cannot spread more than two TBUs. The core proposal of this paper is to posit that tones share a property of melody-locality, meaning that tone well-formedness patterns are describable of an *intersection* of strictly local constraints over the melody and strictly local constraints over the TBU string. This paper surveys a range of local tone patterns, long-distance tone patterns, and local/long-distance interactions such as Bemba, and shows that they share this property. It also shows that they do *not* share properties that make them capturable with precedence or tier-projection grammars, and thus learners based on precedence or tier projection are guaranteed to fail to learn such patterns.

This paper then presents an efficient learning mechanism that is guaranteed to learn any pattern that has this melody-local property. The mechanism is simple: given positive example strings of a pattern, it simultaneously learns strictly local constraints over the strings themeselves and strictly local constraints over their melodies. Given that all of the tone patterns surveyed in this paper are melody-local, melody-locality not only provides for a unified, restrictive characterization of tone patterns, but it also is the first model for how tone is learned.

1.3 Outline of the paper

This paper is structured as follows. §2 defines, in formal language-theoretic terms, what it means to be local, and surveys a range of local and non-local tone patterns. §3 defines a property of melody-locality and shows that all of the patterns in §2 share this property. §4 demonstrates a learner for melody-local patterns. §5 demonstrates that the patterns in §2 escape characterization by the precedence or tier-projection learners previously used for phonological learners, §6 discusses representational issues and future work, and §7 concludes. An Appendix collects formal definitions and proofs for concepts used throughout the paper.

2 Local and non-local tone patterns

This section defines a formal language-theoretic notion of locality and then shows that tone includes both patterns that are local according to this defintion as well as a range of patterns that are non-local according to this definition. The patterns reviewed in this section highlight tone's formal properties and the issues they raise for learnability. This will become particularly clear in §5.

This paper adopts Hyman (2001)'s definition of 'tone pattern', given below in (6).

(6) "A language with tone is one in which an indication of pitch enters into the lexical realization of at least some morphemes" (Hyman 2001, p. 1368; Hyman 2006, p. 229).

This definition includes patterns that are sometimes referred to as "pitch accent," but it is not clear that these patterns should be treated as distinct from other tone patterns (Hyman, 2009). This appears to also be true in terms of their formal properties: "pitch accent" systems and uncontroversial tone systems, at least the ones surveyed here (and in Jardine 2016, to appear), exhibit the same kinds of local and non-local patterns.

2.1 What is a pattern?

The term 'pattern' in this paper refers to a well-formedness generalization holding over surface representations. For example, in Kagoshima Japanese, words have a

H tone either on the penultimate or final syllable (Hirayama, 1951; Haraguchi, 1977; Kubozono, 2012).

(7) Kagoshima Japanese (Hirayama, 1951; Haraguchi, 1977; Kubozono, 2012)

a.	hána	'nose'	HL
b.	sakúra	'cherry blossom'	LHL
c.	kagaríbi	'watch fire'	LLHL
d.	kagaribí-ga	'watch fire' + NOM	LLLHL
e.	haná	'flower'	LH
f.	usagí	'rabbit'	LLH
g.	kakimonó	'document'	LLLH
h.	kakimono-gá	'document' + NOM	LLLLH

Such a generalization refers to a *set* of well-formed strings of TBUs. Borrowing notation from formal language theory, we can call this set L_{KJ}.

(8) $L_{\mathrm{KJ}} = \{$H, HL, LHL, LLHL, LLLHL, ..., LH, LLH, LLLH, LLLLH, ...$\}$

This set includes all and only the strings of Hs and Ls such that a single H falls on either the ultimate or penultimate position. Strings such as *LHLLL, which do not conform to the generalization, are not members of the set.

Note that the ellipses imply that L_{KJ} is an infinite set. Thus, for example, LLL-LLLLHL is in L_{KJ}, even if a word of such length may not exist in the vocabulary of any particular speaker of the dialect. However, this models the fact that they *would* find such a word well-formed, if it were created through morphological processes or borrowing. (Indeed, borrowing of English words has created long words such as [makudonarúdo] 'McDonald's', which nevertheless conform to the generalization (Kubozono, 2012).) Thus, in general, this paper assumes that phonological generalizations apply regardless of word length.

By conceiving of well-formedness generalizations as sets, we can characterize them terms of the kind of computation required to recognize members of the set and reject non-members of the set. The remainder of this section looks at tonal patterns through the lense of a computational definition of locality, as a rigorous way of distinguishing local and non-local types of patterns.

It warrants mention that many tone patterns can be seen in terms of *processes* which transform an underlying form to surface form. However, there are good reasons to focus on well-formedness generalizations first. Well-formedness generalizations are an important part of understanding processes, as is made explicit in constraint-based theories of phonology (Scobbie et al., 1996; Prince and Smolensky, 1993, 2004). This is also true for computational characterizations of processes, which can build on computational characterizations of well-formedness constraints (Chandlee and Heinz, 2018). Furthermore, in terms of learning, learning phonotactics is a simpler problem than learning processes. Again, computational characterizations of learning processes can build on computational characterizations of learning phonotactics (Chandlee et al., 2014; Jardine et al., 2014). This paper thus focuses on well-formedness generalizations, and leaves characterizations of processes to future work.

It is common for tone patterns to only apply to a particular domain; in particular, the generalizations in Bemba referring to unbounded spreading hold in phrase-final position. The following assumes that domain knowledge is prespecified: what is the complexity of pattern L that occurs within a particular domain? Domain information can be included by directly encoding different types of boundaries in the representation; how this affects learning is a distinct, if important, learning problem.

2.2 What is a local pattern?

Armed with a notion of well-formedness generalizations as sets, we can rigorously define a notion of *locality*. The term 'local' is oft-used but can have various meanings, sometimes meaning strict adjacency (Gafos, 1996; Chiośain and Padgett, 2001) or adjacency on a tier (Odden, 1994). This paper, however, adopts the following definition, based on the *strictly local* class of formal languages (McNaughton and Papert, 1971; Rogers and Pullum, 2011).

(9) A tone pattern is *local* iff it is describable by a finite set of strings indicating ill-formed sequences of TBUs (and word boundaries) of a set length.

For example, L_{KJ} is local because it is describable with the following set S_{KJ} of strings of length 3. Here, $\#$ is an extra symbol indicating a word boundary.

(10) $S_{\text{KJ}} = \{\#\text{L}\#, \text{HH}, \text{HLL}, \text{HLH}, \text{LL}\#\}$

The set S_{KJ} can be interpreted as follows: $\#$L$\#$ bans monosyllabic L words; HH bans words with two adjacent H tones; HLL and HLH ban words in which a H appears anywhere to the left of penultimate position; and LL# requires that either the penult or final syllable is H (recall that they both cannot be H, as HH has been banned).

The set L_{KJ} is thus exactly the set of strings which contain none of the substrings in S_{KJ}. We can thus think about S_{KJ} as a set of inviolable *forbidden substructure constraints*. As they are inviolable, these constraints are distinct from Markedness constraints in Optimality Theory (Prince and Smolensky, 1993, 2004).

However, forbidden substructure constraints are computationally very simple (Rogers and Pullum, 2011; Rogers et al., 2013) and, as to be shown in more detail in §3, are efficiently learnable (García et al., 1990; Heinz, 2010a). This stems from their cognitive interpretation, which is as follows. For a set S of forbidden substrings of at most length n, we can check to see if a string is well-formed with respect to S simply by scanning through the string with a window of size n. This is shown below in (11) using S_{KJ} for the strings LLHL (which belongs to L_{KJ}) and *LHLL (which does not).

(11)				
a.	$\boxed{\text{\# L L}}$ H L #	# $\boxed{\text{L L H}}$ L #	# L $\boxed{\text{L H L}}$ #	# L L $\boxed{\text{H L \#}}$
	✓ #LL	✓ LLH	✓ LHL	✓ HL#
b.	$\boxed{\text{\# L H}}$ L L #	# $\boxed{\text{L H L}}$ L #	# L $\boxed{\text{H L L}}$ #	
	✓ #LH	✓ LHL	✗ HLL	

In both (11a) and (b), we see a scanning window of length 3 (indicated by a box) move sequentially through the strings #LLHL# and #LHLL#. Below each box is the substring of length 3 that currently appears in the window. If *at any time* one of the substrings in S_{KJ} appears in the window, the form is judged by the procedure to be ungrammatical. In (11a), the scanner moves all the way through the string without encountering a substring from S_{KJ}, so the form is judged to be grammatical. In (11b), the scanner encounters the forbidden substring HLL$\in S_{\mathrm{KJ}}$ on the third step, and so *LHLL is correctly judged to be ungrammatical. Thus, patterns that can be described with forbidden substring grammars are *local* because their well-formedness generalizations depend entirely on information that can be detected in some fixed window.

To give one more useful example, bounded spreading is also a local pattern. In phrase-medial position in Bemba, all H tone spread to a second mora, unless this incurs an OCP violation (Hyman, 2011a; Bickmore and Kula, 2013).

(12) Binary spreading in Bemba (Hyman, 2011a)

a.	tu-la-mu-súm-á	'we bite'	LLLHH
b.	bá-lá-mu-kak-a	'they tie him up'	HHLLL
c.	bá-lá-mu-súm-á	'they bite him'	HHLHH
d.	bá-la-súm-á	'they bite'	HLHH
	*bá-lá-súm-á		*HHHH

In (12), all H tones are followed by a second H tone, except in (12d), as this would result in adjacent H tones and thus violate the OCP. Let us call the set of strings conforming to this binary spreading generalization L_{BS}.

(13) $L_{\mathrm{BS}} = \{$ H, L, HH, LL, HHL, LHH, LLH, LLL, HHLL, LHHL, LLHH, HLHH, HHLH, HHLLL, HHLHH, LLLHH, ... $\}$

To show that this is local, L_{BS} is exactly the set of strings that do not contain the following forbidden substrings.

(14) $S_{\mathrm{BS}} = \{\#\mathrm{HLL}, \mathrm{LHLL}, \#\mathrm{HL}\#, \mathrm{LHL}\#, \mathrm{HHH}\}$

The first two forbidden substrings in S_{BS}, #HLL and LHLL, capture the constraint that a H tone not already following a H tone cannot be followed by two L tones. (Because of the OCP, it is possible that it is followed by a single L just in case another H tone follows, as in the string HLHH.) This excludes, for example, strings like *HLLL and *LHLL. This forces any such H to instead be followed by a H tone: a string like LHHL contains neither of these substrings. The next two forbidden substrings, #HL# and LHL#, capture the same constraint but at the end of the word, instead of preceding a second L tone; thus *LLHL is ill-formed but LLHH is. The third forbidden substring, HHH, prevents a span of more than two Hs in a row: *LHHH, for example, contains this substring. Thus, the forbidden substrings in S_{BS} capture the binary spreading pattern L_{BS}.

Kagoshima Japanese and bounded spreading in Bemba are just two examples, but constraints that are local in this way are common in tone. Constraints referring to some position near the word edge (like that in Kagoshima Japanese), local OCP or *CLASH constraints that forbid H tones on adjacent TBUs (as in Bemba but also in Shona or Kikuyu; Meyers, 1997; Zoll, 2003), local *TROUGH constraints that forbid a single L-toned TBU in between two Hs (e.g., Kihunde; Goldsmith, 1990), and 'edge-based' constraints that restrict contours or plateaus to either the left or right edge of the word (e.g. as in Mende; Leben, 1973, see also §5), all conform to this definition of locality.

2.3 Non-local patterns

However, there are also well-formedness generalizations in tone that are *non*-local according to this definition, and an advantage of our formulation of it is we know exactly when we have a non-local pattern. As an example, take the following *obligatoriness* constraint in Chuave (Donohue, 1997; Hyman, 2011b), in which every word must have at least one H-toned TBU. (The relevant TBU is the mora.)

(15) Tone in Chuave (Donohue, 1997; Hyman, 2011b)

a.	kán	'stick'	e.	gíngódí	'snore'	i.	kóiom	'wing'
	H			HHH			HLL	
b.	gáán	'child'	f.	dénkábu	'mosquito'	j.	komári	'before'
	HH			HHL			LHL	
c.	gáam	'skim'	g.	énugú	'smoke'	k.	koiyóm	'navel'
	HL			HLH			LLH	
d.	kubá	'bamboo'	h.	amámó	'k.o. yam'			
	LH			LHH				

Words of the shape *L, *LL, *LLL, etc., are ill-formed in Chuave—in other words, a H tone is obligatory. We can thus represent the well-formedess condition in Chuave as the set L_{Ch} in (16) of strings with at least one H.

(16) L_{Ch} = {H, HL, LH, HH, LLH, LHL, LHH, HLL, HLH, HHH, LLLH, ...}

Another way to think of this pattern is that it excludes all strings of only Ls. This is non-local because there is no window of fixed length that can distinguish all and only the strings with no H. The scanner diagrams in (17) help to illustrate why.

(17) a. # H $\boxed{\text{L L L} \cdots \text{L}}$ L #

✓ LLL...L

b.* # L $\boxed{\text{L L L} \cdots \text{L}}$ L #

✓ LLL...L

In the well-formed string in (17a), we have an H followed by some arbitrary number of Ls. Our scanner window must accept this sequence of Ls, as the string is well-formed (because it has an H). However, our scanner cannot distinguish this sequence of Ls from the one in the ill-formed string in (17b). The reason for this is that, because it is checking substrings of some bound length, it cannot 'remember' that it has seen the H in (17a), or that it has not seen a H in (17b). Thus it incorrectly will judge both strings to be well-formed. Importantly, increasing the length of substrings that the scanner checks does not help—there will always be some pair of strings like (17a) and (b) that it cannot distinguish.[2] The Appendix cites a formal characterization of this intuition that can used to prove whether or not a pattern is local according to our definition (i.e., whether or not it is a strictly local formal language).

Another common example is *unbounded tone plateauing* (UTP) (Kisseberth and Odden, 2003; Hyman, 2011b; Jardine, 2016), in which surface well-formedness dictates that only one plateau, or unbroken stretch, of high-toned TBUs is allowed in a domain. The following examples are from Luganda (Hyman and Katamba, 2010; Hyman, 2011b).

(18) Luganda (Hyman and Katamba, 2010; Hyman, 2011b)

a.	kitabo	'book'	LLL
b.	mutéma	'chopper'	LHL
c.	kisikí	'log'	LLH
d.	mutémá+bísíkí	'log chopper'	LHHHHH
e.	*mutéma+bisikí	" "	*LHLLLH

Hyman and Katamba (2010) characterize UTP as a process which merges two H tones together, but we can also characterize this pattern as the result of a long-distance *HLH (or *TROUGH, in Yip (2002)'s terms) constraint which bans *any* two H-toned TBUs separated by any number of L-toned TBUs (Hyman, 2011b).[3] The set L_{UTP} of strings that satisfy this constraint are given in (19).

[2]This hinges on our assumption of the generalization holding for strings of arbitrary length. Again, this is usually how phonological generalizations are conceived—as independent of any constraints of word (or morpheme) length. Thus, for example, while the Chuave forms are of up to length 3, Donohue (1997) posits a constraint against all-L words that holds regardless of their length (p. 355). This is the generalization that we study here.

[3]An example of UTP as a pure phonotactic constraint can be seen in Moro nouns (Jenks and Rose, 2011).

(19) $L_{\text{UTP}} = \{$ L, H, LL, LH, HL, HH, LLL, LLH, LHL, LHH, HLL, HHL, HHH, LLLL, LLLH, LLHL, LLHH, LHLL, LHHL, LHHH, HLLL, HHLL, HHHL, ... $\}$

Again, L_{UTP} is exactly the set of strings that do not contain the sequence HL^{n}H, where L^n indicates a sequence of n L-toned TBUs. This is non-local for a similar reason to that of Chuave: there is no scanner window of fixed length that can distinguish between well-and ill-formed strings. This is illustrated diagrammatically in (20).

(20) a. # L $\boxed{\text{L L L} \cdots \text{L}}$ H #
✓ LLL...L

b. * # H $\boxed{\text{L L L} \cdots \text{L}}$ H #
✓ LLL...L

In UTP, a string of the form L^{n}H is well-formed, as shown in (20a), and so the scanner must admit a sequence of n L-toned TBUs. However, the scanner will then incorrectly admit an ill-formed string with two Hs separated by n L-toned TBUs, as shown in (20b). Again, this holds for an arbitrary n.

We have thus illustrated the notion of a long-distance well-formedness constraint in tone with the obligatory H tone constraint in Chuave and the constraint against more than one H-plateau in UTP. So far, this notion has been defined negatively: these patterns *do not* have a particular locality property. Ultimately, we will characterize these patterns in terms of what they *share*, and then exploit that shared property for a learning model. However, such a property must also take into account *interactions* between local and long-distance patterns in tone. We now turn to such patterns.

2.4 Local/non-local interactions

Tone patterns also exhibit interactions between local and long-distance constraints. We review three such cases. First, Arigibi (New Guinea; Donohue, 1997) has a constraint somewhat like UTP: there can only be one H tone in a word. However, unlike UTP, this H tone cannot spread. The relevant TBU here is the mora.

(21) Arigibi (Donohue, 1997)

a.	naː L	'finish'	e.	umú LH	'dog'	h.	olaʔolá LLLH	'red'
b.	tutuː LL	'long'	f.	númo HL	'louse'	i.	tuniʔʌ́ʔʌ LLHL	'all'
c.	vovoʔo LLL	'bird'	g.	mudɛbɛ́ LLH	'claw'	j.	idómai LHLL	'eye'
d.	ɛlaila LLLL	'hot'	f.	ivío LHL	'sun'	k.	núʔʌtama HLLL	'bark'
			g.	ŋgíʔɛpu HLL	'heart'			

This gives rise to what Hyman (2009) calls a *culminativity* constraint: a word can have *at most* one H tone. This is distinguished from obligatoriness constraints in the sense that H-less words are allowed (such as (21d) [ɛlaila] 'hot'). (UTP can thus also be considered a culminativity constraint.) The set of well-formed TBU strings in Arigibi is given below in (40) as L_{Ar}.

(22) $L_{\mathrm{Ar}} =$
{H, L, LL, LH, HL, LLL, LLH, LHL, HLL, LLLL, LLLH, LLHL, LHLL, ...}

This is non-local for the same reasons as UTP: we must ban any second occurence of a span of H tones. However, there is also a local constraint at play: any span of H tones must be exactly one mora long. This local in the same way that bounded spreading is, except that here, the constraint is against HH substrings of consecutive H-toned moras.

A similar constraint holds in Prinmi (Ding, 2006; Hyman, 2009). In Prinmi, words must have exactly one span of H tones.[4] This span can either be one or two moras long.

(23) Prinmi (Ding, 2006)

[4]In word-final position, this can technically be realized as a rising or falling tone; contours are abstracted away from here to focus on the long-distance nature of the pattern. For more on contours, see §6.1.

a.	bɨ́ H	'honey'	e.	bɨ́bᶦoᶦo HLL	'roasted flour with honey'
b.	bɨ́ge HL	'as for honey'	f.	bɨ́ɬíɹu HHL	'sunflower stem'
c.	bɨ́gé HH	'as for sun'	g.	tʃ'ɨn̥ɨ́dʒjɛ̃ LHL	'dog-nose group'
d.	tʃ'ɨmɛ̃́ LH	'dog hair'	h.	tõpúk'ʉ́ LHH	'donkey head'
			i.	ɹətʃiʃṍ LLH	'clean liquor'

j.	bɨ́bᶦoᶦoge HLLL	'as for roasted flour with honey'	m.	tõpúmɜ́ɬe LHHL	'donkey tail'
k.	bɨ́ɬípɜtsɨ HHLL	'sunflower'	n.	dʒjõdʒɨmɜ́ɬe LLHL	'buffalo tail'
l.	tʃ'ɨn̥ɨ́dʒjɛ̃ɹə LHLL	'dog-nose ' groups'	o.	ɹətʃiʃṍgé LLHH	'as for clean liquor'

In terms of tonal phonotactics, all L-toned words are banned (*LLL), as are words with more than one H-tone span (*HLH). Thus, H is culminative, as in Arigibi, but also obligatory, as in Chuave. Furthermore, whereas in Arigibi there is a constraint against H spreading to more than one mora, in Prinmi H can appear on either one or two moras. The set of possible well-formed words in Prinmi can thus be modelled as the set L_{Pr} given in (24).

(24) $L_{\mathrm{Pr}} =$
{H, LH, HH, HL, LLH, LHL, LHH, HHL, HLL, LLLH, LLHL, LLHH, ...}

Thus, like Arigibi and Chuave, L_{Pr} is a long-distance in that there must be exactly one H-tone span in the word, but like Arigibi it is also local in that there is a bound on the length of this H-tone span. The difference is that in L_{Pr}, H-tone spans longer than three are banned (as in bounded spreading in Bemba).

Finally, a perhaps more striking case of a long-distance constraint interacting with bounded spreading occurs in Bemba (Bickmore and Kula, 2013, 2015). This pattern occurs in phrase-final position in Bemba, as opposed to the bounded spreading generalization given in §2.2, which occured in phrase-medial position. In phrase-final position in Bemba, the last H spreads unboundedly to the end of the word, while all preceding Hs undergo bounded spread (Bickmore and Kula, 2013, 2015). The following data are from Northern Bemba, in which the bounded spread is binary (in Copperbelt Bemba, bounded spread is ternary Bickmore and Kula 2013, 2015). In Bemba the relevant TBU is the mora.

(25) Bemba (Bickmore and Kula, 2013)

a.	tu-ka-pat-a	'we will hate'	LLLL
b.	tu-léé-pát-á	'we are hating'	LHHHH
c.	bá-ká-fík-á	'they will arrive'	HHHH
d.	tú-lúb-ul-ulé	'we should explain '	HHLLH
e.	twáá-ku-láá-pá	'we will be drawing (water)'	HHLHHH

A lone H tone, such as in (25b) or (c), or the second of two tones, such as in (d) and (e), must spread to the final TBU. Thus, strings like *HHLLL are ill-formed. However, any preceding H tone, such as can be seen in (25d) and (e), can only spread to two TBUs; thus, HHLLH is well-formed but *HHHLH is not. In other words, unbounded spreading is blocked by a following H tone, anywhere in the word, resulting instead in bounded spreading. (As in the pattern described in §2.2, bounded spreading respects the OCP.)

The set of possible strings of TBUs that are well-formed according to this generalization is given as L_{Be} in (26). Because this set is rather complex, (26) breaks it down into the number of H-spans in the strings.

(26) L_{Be} is exactly the set that is the union of

a. The set of all L strings; {L, LL, LLL, LLLL ...}

b. The set of strings containing a single H span that continues to the end of the string;
{H, HH, HHH, ..., LH, LHH, LHHH, LLHH, LLLH, LHHHH, LLHHH, ... }

c. The set of strings containing more than one H span, where the last H span continues to the end of the string and any preceding H span is maximally of length two;
{HLH, HHLH, HLHH, HHLLH, HHLHH, LHHLH, HHLLHH, HHLLLH, ... }

This is an interaction between a long-distance constraint and a local one. A long-distance constraint adjudicates between H spans that precede another H, and a local constraint enforces bounded spreading for exactly those H spans for which another H follows.

Thus, tone patterns not only include local and non-local constraints, but they also include their interactions. The following section proposes a class of grammars that characterizes this range of patterns in a unified way.

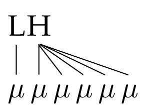
3 The proposal

3.1 Melody constraints

This paper proposes a formal grammar for tone that 1) checks local constraints over the tone specifications of TBUs and 2) checks local constraints over the tonal melodies. The intuition is that tone patterns can have both local and non-local constraints, which can interact, but only in that both must be satisfied at once.

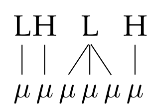
The grammars proposed here take inspiration from the local autosegmental grammars of Jardine (2017, to appear), which use autosegmental representations that obey full specification of tones and TBUs and the OCP. For example, the plateauing example from Luganda, (18c) [mutémá+bísíkí] 'log chopper,' would be represented as in (27a).

(27) a.



[mutémá+bísíkí]
'log chopper' (=18d)

b. *



*[mutéma+bisikí]
(=18e)

The autosegmental representation for the ill-formed *[mutéma+bisikí] is given in (27b). Jardine (2017, to appear) shows that local grammars over these representations can capture long-distance processes like UTP through their ability to posit constraints over the melody, or string of tonal autosegments. Thus, in Luganda, ill-formed words contain a HLH sequence in the melody of their autosegmental representation. This distinguishes (27a), whose melody is LH, from (27b), whose melody is *LHLH, which contains the illicit substring HLH.

The novel contribution of this paper centers around the idea of extracting this melody directly from the surface string. We can define a function $\texttt{mldy}(w)$ that does this by taking a string w of H and L-toned TBUs and replacing each *span* (or unbroken sequence) of Hs or Ls with a single H or L, in the order they appeared. This process is shown step-by-step in (28) for LHHHH (the string of TBUs corresponding to (18)d) [mutémá+bísíkí] 'log chopper') and LHLLH (corresponding to (18)e) *[mutéma+bisikí]). (This function is defined formally in the Appendix.)

(28) a. $\texttt{mldy}(\mathrm{LHHHHH}) = \underset{\mathrm{L}}{\boxed{\mathrm{L}}}\,\underset{\mathrm{H}}{\boxed{\mathrm{H\,H\,H\,H\,H}}} = \mathrm{LH}$

b. $\texttt{mldy}(\mathrm{LHLLLH}) = \underset{\mathrm{L}}{\boxed{\mathrm{L}}}\,\underset{\mathrm{H}}{\boxed{\mathrm{H}}}\,\underset{\mathrm{L}}{\boxed{\mathrm{L\,L\,L}}}\,\underset{\mathrm{H}}{\boxed{\mathrm{H}}} = \mathrm{LHLH}$

In (28a), the string LHHHHH is composed of a span of a single L-toned TBU followed by a span of five H-toned TBUs.[5] The `mldy` function collapses these two spans into a string LH which represents a L span followed by a H span. Similarly, in (28b), the string LHLLLH is composed of a single L, a single H, a span of three Ls, and a H. Collapsing the span of three Ls into a single L, the function obtains LHLH. We have now generated exactly the melody strings from the ARs in (27a) and (b), respectively. We can then posit a local grammar M_{UTP} of forbidden substrings that operates over these melody strings.

(29) $M_{\mathrm{UTP}} = \{\mathrm{HLH}\}$

We use the 'M' notation to indicate forbidden substring grammars that operate over the strings generated by `mldy`. Given this, M_{UTP} will be violated by any string w of TBUs for which $\texttt{mldy}(w)$ contains an HLH sequence. Thus, for example, *LHLLLH is *not* in the set of strings described by M_{UTP}, because $\texttt{mldy}(\mathrm{LHLLLH}) = \mathrm{LHLH}$, which contains the substring HLH. However, LHHHHH *is* in the set of strings described by M_{UTP}, because $\texttt{mldy}(\mathrm{LHHHHH}) = \mathrm{LH}$ does not include the forbidden substring HLH.

In general, $\texttt{mldy}(w)$ for any string w with two Hs separated by at least one L will include the forbidden substring HLH, and thus will be excluded by M_{UTP}. Thus, the grammar M_{UTP} describes exactly the set L_{UTP} from §2.3 representing the long-distance UTP generalization. It has done this by implementing a constraint against a HLH sequence *in the melody* of an AR.

This reduction of a long-distance pattern to a local melody constraint depends on multiple association and the assumption that, on the surface, autosegmental

[5]This function draws inspiration from Jardine and Heinz (2015)'s concatenation operation for generating OCP-obeying ARs from strings. For now, we will gloss over the treatment of contours, which can be straightforwardly dealt with but are not necessary to capture the tone patterns from §2. It will be shown in §6.2 how to adapt our melody function to incorporate contours.

representations that OCP merges all adjacent, like-toned TBUs. Thus, in the UTP case, any stretch of L-toned TBUs falling between two H-toned TBUs will be associated to a single L in the melody, keeping the H autosegments in the melody-local to each other. These assumptions will be discussed in more detail in §6.1, but two points are worth noting here. One, clear surface violations of the OCP are extremely rare, and when they are, they are signaled by downstep. Two, as we shall see, melodies generated with these assumptions allow us to capture to a good approximation the typology of long-distance patterns in tone. The potential utility and consequences of including underspecification and OCP violations will be discussed in §6.1.

3.2 Melody-local grammars

While we have shown that a melody constraint can handle a long-distance well-formedness generalization, we still need local constraints, as demonstrated by the existence of patterns like Kagoshima Japanese and bounded spreading in Bemba. The proposal here is thus that tonal well-formedness is governed by grammars G that take the form in (30).

(30) $G = (S, M)$

In (30), a grammar G is an ordered pair of a forbidden substring grammar S that operates over strings and a melody grammar M that operates over the melodies of strings. We say that a string w satisfies G, written $w \models G$, as defined in (31).

(31) A string w satisfies a melody-local grammar $G = (S, M)$ iff

a. w contains none of the substrings in S, and

b. $\texttt{mldy}(w)$ contains none of the substrings in M.

Crucially, (31) is defined *conjunctively*; a string w satisfies the grammar if and only if it satisfies *both* (31a) and (31b). Herein lies the central claim of this paper: that local and non-local tonal constraints can only interact through conjunction. We can then define ae new version of locality, call it *melody-local*, parallel to the definition of locality in (9).

(32) A tone pattern is *melody-local* iff it is describable by a melody-local grammar.

We can take (32) to be a *hypothesis* about tone: tone well-formedness patterns must be melody-local. This is this a restrictive hypothesis about tone that is tightly connected to a hypothesis about how tone patterns are learned. The remainder of this section provides support for this hypothesis by showing that the attested tone patterns discussed in §2 are all melody-local. It is then shown in §4 that melody-local grammars have an efficient, provably correct learning algorithm.

While melody-local grammars represent long-distance constraints in terms of an autosegmental melody, they represent local information in terms of strings of TBUs, and not autosegmental associations. The string representations of TBUs can be taken as a shorthand for autosegmental representations: even if ARs are the 'correct', psychologically real representation of tone, it is still the case that there are constraints in tone that are local *when viewed in terms of strings of TBUs*. This is discussed in more detail in §6.1. In fact, the clear separation between local and melody constraints offered by melody-local grammars is not present in ARs: as shown in §5, ARs allow arbitrary constraints that refer to both local and melody information. Melody-local grammars, then, are more restrictive than local AR grammars.

3.3 Analyzing the attested typology with melody-local grammars

To begin, we can describe the UTP generalization L_{UTP} with the grammar G_{UTP} as given in (33).

(33) $G_{\mathrm{UTP}} = (S_{\mathrm{UTP}} = \{\}, M_{\mathrm{UTP}})$

As already described, M_{UTP} captures the long-distance constraint that no two H-toned TBUss can appear separated by one or more L-toned TBUs. This is sufficient to describe the pattern: at the local level, Hs and Ls can appear freely. Thus, the local component of the grammar, S_{UTP}, is empty and thus always vacuously satisfied. This indicates that UTP is purely a long-distance well-formedness generalization, but it is still melody-local.

This is also the case for the Chuave pattern, in which at least one H-toned TBU must appear in the word. The set L_{Ch} of strings conforming to this pattern is repeated below in (34) from (16).

(34) L_{Ch} = {H, HL, LH, HH, LLH, LHL, LHH, HLL, HLH, HHH, LLLH, ...}

Again, this is the set that excludes any strings L, LL, LLL, LLLL, etc., that only consist of L-toned TBUs. That is, it excludes any string that consists of a single L-tone span. For any string with exactly one L tone span, the `mldy` function will condense it into a single string L: $\texttt{mldy}(\text{L}) = \texttt{mldy}(\text{LL}) = \texttt{mldy}(\text{LLL}) = \text{L}$. We can then ban all such strings by positing a set $M_{\text{Ch}} = \{\#\text{L}\#\}$ consisting of a single substring #L#, which indicates a L-toned TBU which is both at the beginning and the end of the string. The Chuave pattern L_{Ch} is thus captured by the following melody-local grammar.

(35) $G_{\text{Ch}} = (S_{\text{Ch}} = \{\}, M_{\text{Ch}} = \{\#\text{L}\#\})$

Any string that contains an H-toned TBU will not contain #L# in its melody. For example, for the well-formed string LLH, $\texttt{mldy}(\text{LLH}) = \text{LH}$, which does not contain #L#. However, for the ill-formed string *LLLL, $\texttt{mldy}(\text{LLLL}) = \text{L}$, which does contain #L#. Thus, M_{Ch} captures exactly the set of strings that conform to the Chuave pattern. Note again that S_{Ch} is empty and thus always vacuously satisfied—this means there are no local constraints on the distribution of H- and L-toned TBUs.

In purely local patterns, the local component of the grammar is instead nonempty, though the melody component of the grammar may have content. For example, in §2.2 already established that L_{KJ} is describable by the local set S_{KJ}. Both are repeated in (36).

(36) a. $L_{\text{KJ}} = \{$H, HL, LHL, LLHL, LLLHL, ..., LH, LLH, LLLH, ...$\}$
b. $S_{\text{KJ}} = \{\#\text{L}\#, \text{HH}, \text{HLL}, \text{HLH}, \text{LL}\#\}$

This could be described with an empty M, as the S_{KJ} is sufficient to describe the pattern exactly. However, as shall be seen in §4, this is not the grammar a learner would posit. A learner would also notice that, at the melody level, HLH sequences (that is, words with more than two distinct H spans) and #L# sequences (that is, words with no H spans) do not appear. Thus, we can also describe L_{KJ} with the following melody-local grammar.

(37) $G_{\text{KJ}} = (S_{\text{KJ}}, M_{\text{KJ}} = \{\text{HLH}, \#\text{L}\#\})$

This grammar captures the generalization that the local constraints in Kagoshima Japanese also give rise to an obligatoriness and culminativity constraint: there must be exactly one H tone in the melody.

Our other example of a local pattern, bounded spreading, is also melody-local for the same reasons. Again, this was a pattern in which H tones must spread up to two TBUs, unless that makes it adjacent to another H tone. This pattern, L_{BS}, is repeated below in (38), along with the local grammar S_{BS} from §2.2 that describes it.

(38) a. $L_{\text{BS}} =$ { H, L, HH, LL, HHL, LHH, LLH, LLL, HHLL, LHHL, LLHH, HLHH, HHLH, HHLLL, HHLHH, LLLHH, ... }

b. $S_{\text{BS}} = \{\#\text{HLL}, \text{LHLL}, \#\text{HL}\#, \text{LHL}\#, \text{HHH}\}$

Here, we have no constraints on the melody: the strings in L_{BS} comprise any combination of L and H spans. Thus, a melody-local grammar for L_{BS} consists of S_{BS} and an empty melody grammar $M_{\text{BS}} = \{\}$.

(39) $G_{\text{BS}} = (S_{\text{BS}}, M_{\text{BS}} = \{\})$

Again, because M_{BS} is empty, it is always vacuously satisfied. Thus, the description of the pattern falls entirely to S_{BS}. However, L_{BS} is still melody-local.

For the patterns that show an interaction between local and long-distance constraints, both the content of S and M will play a role in determining the pattern. Let us begin with Arigibi. In Arigibi, every word has at most one H-toned TBU. This pattern, L_{Ar}, is repeated below in (40).

(40) $L_{\text{Ar}} =$
{H, L, LL, LH, HL, LLL, LLH, LHL, HLL, LLLL, LLLH, LLHL, LHLL, ...}

This pattern requires both a local constraint and a constraint on the melody. Like with UTP, we need to forbid HLH substrings in the melody, to prevent distinct H spans in the string. Thus, we posit $M_{\text{Ar}} = \{\text{HLH}\}$, identical to M_{UTP}. However, unlike UTP, we have to restrict H spans to only consist of a single TBU. For this, we posit a local constraint $S_{\text{Ar}} = \{\text{HH}\}$. This results in the grammar in (41).

(41) $G_{\text{Ar}} = (S_{\text{Ar}} = \{\text{HH}\}, M_{\text{Ar}} = \{\text{HLH}\})$

Examples of the interaction of these two constraints are given in Table 1. Recall that for a string to be well-formed with respect to G_{Ar}, it has to satisfy S_{Ar}, and the melody of that string has to obey M_{Ar}.

The string in Table 1a, *LHHHL, is ill-formed because it contains the substring HH and thus does not satisfy S_{Ar}. Thus, it is ill-formed regardless of

	w	$\mathtt{mldy}(w)$	$S_{\mathrm{Ar}} = \{\mathrm{HH}\}$	$M_{\mathrm{Ar}} = \{\mathrm{HLH}\}$
a.	*L$\underline{\text{HH}}$HL	LHL	✗	✓
b.	*HLLHL	$\underline{\text{HLH}}$L	✓	✗
c.	*HL$\underline{\text{HH}}$	$\underline{\text{HLH}}$	✗	✗

Table 1: Ill-formed strings according to G_{Ar}. Offending substrings are underlined.

the fact that $\mathtt{mldy}(\mathrm{LHHHL}) = \mathrm{LHL}$ does satisfy M_{Ar}. In contrast, Table 1b, *HLLHL, satisfies S_{Ar} because it does not contain a HH sequence. However, $\mathtt{mldy}(\mathrm{HLLHL}) = \mathrm{HLHL}$, which does not satisfy M_{Ar}, and so *HLLHL is ill-formed with respect to the grammar. It is also possible to violate both constraints, as in Table 1c, *HLHH.

Thus, G_{Ar} rejects any string with more than one span of Hs, and it rejects any string for which a span of Hs is more than one TBU long. However, any string with at most one H-toned mora will satisfy both S_{Ar} and M_{Ar}; thus, G_{Ar} describes exactly L_{Ar}. To illustrate, some examples from L_{Ar} are shown in Table 2.

	w	$\mathtt{mldy}(w)$	$S_{\mathrm{Ar}} = \{\mathrm{HH}\}$	$M_{\mathrm{Ar}} = \{\mathrm{HLH}\}$
a.	LLL	L	✓	✓
b.	LLHL	LHL	✓	✓
c.	HLLLL	HL	✓	✓

Table 2: Well-formed strings according to G_{Ar}.

The constraint on words in Prinmi is similar in that there has a melody constraint against HLH sequences. The Prinmi pattern is repeated below in (42).

(42) $L_{\mathrm{Pr}} =$
{H, LH, HH, HL, LLH, LHL, LHH, HHL, HLL, LLLH, LLHL, LLHH, ...}

One difference between Prinmi and Arigibi is that whereas Arigibi restricted H tones to a single TBU, Prinmi allows H tones on either one or two TBUs. This local constraint can be modeled with the local grammar S_{Pr} in (43).

(43) $S_{\mathrm{Pr}} = \{\mathrm{HHH}\}$

The local grammar S_{Pr} contains the single forbidden substring HHH, which forbids a H-tone span from spreading more than two TBUs. The other difference between Prinmi and Arigibi is that all-L words are ill-formed; H is obligatory. Thus the melody grammar for Prinmi must be $M_{\text{Pr}} = \{\text{HLH}, \text{\#L\#}\}$, combining Arigibi's constraint against two distinct H spans with Chuave's constraint against L-toned melodies. Table 3 gives examples illustrating that strings that are ill-formed with respect to the Prinmi pattern are correctly rejected by a grammar G_{Pr} that combines S_{Pr} with this melody constraint. This G_{Pr} is given in (44).

(44) $G_{\text{Pr}} = \{S_{\text{Pr}}, M_{\text{Pr}} = \{\text{HLH}, \text{\#L\#}\}\}$

w	$\texttt{mldy}(w)$	S_{Pr}	$M_{\text{Pr}} = \{\text{HLH}, \text{\#L\#}\}$
a. *L$\underline{\text{HHH}}$	LH	✗	✓
b. *$\underline{\text{HHH}}$H	H	✗	✓
c. *LLLL	$\underline{\text{L}}$	✓	✗
d. *HHLLHH	$\underline{\text{HLH}}$	✓	✗
e. *LHHLLLHH	L$\underline{\text{HLH}}$	✓	✗

Table 3: Ill-formed strings according to G_{Pr}. Offending substrings are underlined.

In Table 3a and (b), *LHHH and *HHHH are correctly rejected by S_{Pr} because they contain the forbidden substring HHH, enforcing the constraint against a H-span that has spread to more than two TBUs.

The necessity of both a local and a melody constraint is shown in Table 3c through (e). First, *LLLL does not run afoul of S_{Pr}, but as it contains only L-tones, its melody string is L, which contains the forbidden melody substring #L# and is thus ruled ungrammatical. The ill-formed string *HHLLHH satisfies S_{Pr}, as both H tone spans are exactly two TBUs long. However, it violates the melody grammar M_{Pr}, because $\texttt{mldy}(\text{HHLLHH}) = \text{HLH}$, which contains the forbidden melody substring HLH. The same goes for Table 3e. In this way, the combined local and melody constraints of G_{Pr} work in concert to capture the generalization that words in Prinmi must have exactly one H-tone span, and that this span must be at most two TBUs long. For completeness, Table 4 illustrates how strings in L_{Pr} satisfy both S_{Pr} and M_{Pr}.

	w	$\texttt{mldy}(w)$	S_{Pr}	$M_{\text{Pr}} = \{\text{HLH}, \text{\#L\#}\}$
a.	HLLL	HL	✓	✓
b.	LLHH	LH	✓	✓
c.	LLLHHL	LHL	✓	✓

Table 4: Well-formed strings according to G_{Pr}.

Finally, we turn to the constraints governing bounded and unbounded spread in Bemba. Recall that phrase-finally, the last H-tone span must spread to the end of the word, whereas all previous H-tones must spread up to a maximum of two TBUs. This pattern is repeated below in (45).

(45) L_{Be} is exactly the set that is the union of

a. The set of all L strings; {L, LL, LLL, LLLL ...}

b. The set of strings containing a single H span that continues to the end of the string;
{H, HH, HHH, ..., LH, LHH, LHHH, LLHH, LLLH, LHHHH, LLHHH, ... }

c. The set of strings containing more than one H span, where the last H span continues to the end of the string and any preceding H span is maximally of length two;
{HLH, HHLH, HLHH, HHLLH, HHLHH, LHHLH, HHLLHH, HHLLLH, ... }

A local grammar that produces bounded spreading, S_{BS}, was already given in in (14) and (38); for the phrase-final pattern in Bemba, we adopt this with one small adjustment as S_{Be} below in (46).

(46) $S_{\text{Be}} = \{\#\text{HLL}, \text{LHLL}, \#\text{HL}\#, \text{LHL}\#, \text{HHHL}\}$

The substrings #HLL, LHLL, #HL#, LHL#, which motivate binary spreading of an H, are all as they were in S_{BS}. However, whereas S_{BS} banned HHH sequences wholesale, S_{Be} instead forbids the substring HHHL, which only bans HHH sequences that are followed by an L. This is because HHH sequences *are* allowed in L_{Be}, but only if they are part of a final H span that reaches the end of the word.

The local constraints in S_{Be} do a lot of the work. As shown in Table 5, they force bounded spreading for any H span that does not reach the end of the word, but allow unbounded spreading if the H span does reach the end of the word.

w	S_{Be}
a. *HLLH	✗
b. *LHLLH	✗
c. HHLLH	✓
d. *HHHHLLL	✗
e. LHHH	✓
f. *HHLLL	✓

Table 5: Well-formed and ill-formed strings according to S_{Be}. For the ill-formed strings, offending substrings are underlined.

Table 5a and b contrast *HLLH and *LHLLH, in which a non-final H span is not binary, with HHLLH. The local grammar S_{Be} correctly marks *HLLH as illicit because it contains #HLL—i.e., the initial H has not spread—and *LHLLH because it contains the substring LHLL, indicating some word-medial H-span that is not binary. The well-formed HHLLH, in contrast, contains neither of these substrings. In Table 5d shows a string *HHHHLLL in which a non-final H has spread more than two TBUs. This is illicit because it contains the substring HHHL. This is contrasted with Table 5e, LHHH, which has a H that has spread more than two TBUs but spreads to the end of the word. This does not contain any of the substrings in S_{Be}, and is thus correctly judged well-formed.

Table 5f, however, is *in*correctly judged well-formed by S_{Be}: it has a non-final binary span of Hs, and so does not contain any of the substrings in S_{Be}. Herein lies the long-distance nature of the pattern: the local grammar S_{Be} cannot tell whether or not a H span is the last in the word. In other words, it cannot distinguish between the well-formed HHLLH and the ill-formed *HHLLL. Thus, it cannot force unbounded spreading in the latter.

The melody constraint in M_{Be} in (47), however, can force unbounded spreading of the final H in the word.

(47) $M_{\text{Be}} = \{\text{HL}\#\}$

The forbidden melody substring HL# forbids a span of Ls intervening between a H and the end of the word. In other words, any final H span must spread to the end of the word. This is illustrated below in Table 6.

	w	$\texttt{mldy}(w)$	M_{Be}
a.	HHLLH	HLH	✓
b.	LHHH	LH	✓
c.	*HHLLL	<u>HL</u>	✗

Table 6: Well-formed and ill-formed strings according to M_{Be}. For the ill-formed strings, offending substrings are underlined.

Table 6a and b show that the strings HHLLH and LHHH are well-formed with respect to M_{Be} because their melodies, HLH and LH, respectively, do not end in a HL sequence. In contrast, the melody of the ill-formed *HHLLL does: $\texttt{mldy}(\text{HHLLL}) = \text{HL}$. Thus, the melody constraint M_{Be} forces spreading of the last H to the end of the word. This captures Bickmore and Kula (2015)'s generalization that a phrase-final TBU wants be associated to the final H tone in a word, if one exists.

A combined table in Table 7 shows how a grammar G_{Be} (itself given in (48)) correctly distinguishes strings in L_{Be} from strings not in L_{Be}. Note that the well-formed strings (Table 7a through c) satisfy both S_{Be} and M_{Be}, while the ill-formed strings (Table 7d through f) violate at least one.

(48) $G_{\text{Be}} = (S_{\text{Be}}, M_{\text{Be}})$

Thus, through a combination of forbidden substring constraints operating both locally and over the melody, melody-local grammars are able to capture local patterns, long-distance patterns, and patterns in which local- and long-distance patterns interact.

	w	$\texttt{mldy}(w)$	S_{Be}	M_{Be}
a.	HHLLH	HLH	✓	✓
b.	LHHH	LH	✓	✓
c.	HHLHHH	HLH	✓	✓
d.	*$\underline{\text{LHLL}}$H	LHL	✗	✓
e.	*HHLLL	$\underline{\text{HL}}$	✓	✗
f.	*H$\underline{\text{HHHL}}$LL	$\underline{\text{HL}}$	✗	✗

Table 7: Well-formed and ill-formed strings according to G_{Be}. For the ill-formed strings, offending substrings are underlined.

4 Learning melody-local grammars

4.1 The learning framework

Let us now turn to the learnability of melody-local patterns. The learning criteria adopted here are that of Gold (1967), which are roughly as follows. The Gold framework considers learners (learning algorithms) that take as input a finite set of examples and generalizes a grammar representing a (potentially infinite) pattern. A class of patterns is Gold-style learnable if there is some learner such that, for every pattern in the class, there is some finite sample for which the learner returns that pattern exactly. In other words, there is a learner that is guaranteed to learn every pattern in the class, given a sufficient sample of that pattern.

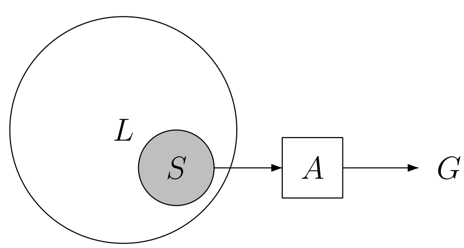


Figure 1: Model of Gold-style learning. A learner A takes as input a finite sample S of pattern L and generalizes to a grammar G. The learner is successful if, given that S is a sufficient sample, G describes L.

Gold-style learning is an abstraction in some ways. First, it poses a strong requirement in that the learner must match the target exactly. However, a learner that satisfies the Gold requirement is *guaranteed* to learn the pattern given a sufficient sample of data. In other words, the learner's behavior is well-understood. We thus gain an understanding about the structural properties of the patterns that make it possible to generalize them from finite data.

Second, the learner is given a 'perfect' sample, in that all of its data is representative of the target pattern. Thus, a Gold-style learner says nothing about how to learn in the face of exceptions. This represents, however, a factorization of the learning problem: we understand first how to learn the class of patterns itself, and then we can use this knowledge to posit gradient or statistical learners that can learn these patterns in the face of exceptions. In other words, stochastic learners can use the structure of categorical learners (Heinz and Rogers, 2010).

Finally, a cognitively plausible learning algorithm computes in a reasonable amount of time. We also further pose a requirement that the algorithm is efficient (de la Higuera, 2010).

The following describes a learner for melody-local patterns that satisfies all of these criteria: given a finite, representative set of examples from a melody-local pattern, it converges to that exact pattern. It also does so in linear time, and is thus very efficient. The following illustrates the correctness and efficiency of the algorithm by showing how it behaves on inputs from two of the patterns from the preceding section. This should not be mistaken for an assertion that the algorithm is successful on natural language data, but rather a demonstration that given the right data, it will efficiently and accurately converge. A full proof of the correctness of the algorithm is given in the Appendix.

4.2 Strictly local learning

The learner posited here is based entirely on the learning of strictly local patterns (García et al., 1990). This proceeds as follows. First, we fix some number k. This will be the length of substrings the learner is sensitive to. Given a string, we can generate its substrings of length k using the same scanning procedure as shown in (11). The following shows that the 3-substrings (substrings for which $k = 3$) of #LLHL# are $\{$#LL, #LLH, LHL, HL#$\}$.

(49) $\underbrace{\boxed{\#\ \mathrm{L}\ \mathrm{L}}}_{\#\mathrm{LL}}\ \mathrm{H}\ \mathrm{L}\ \#$ $\quad$ $\#\ \underbrace{\boxed{\mathrm{L}\ \mathrm{L}\ \mathrm{H}}}_{\mathrm{LLH}}\ \mathrm{L}\ \#$ $\quad$ $\#\ \mathrm{L}\ \underbrace{\boxed{\mathrm{L}\ \mathrm{H}\ \mathrm{L}}}_{\mathrm{LHL}}\ \#$ $\quad$ $\#\ \mathrm{L}\ \mathrm{L}\ \underbrace{\boxed{\mathrm{H}\ \mathrm{L}\ \#}}_{\mathrm{HL}\#}$

More examples are given in Table 8. Note that, for example, the sets of 3-substrings for #LLHL# and #LLLHL# are identical.

String	Substrings of length 3
#LLHL#	{#LL, LLH, LHL, HL#}
#LHL#	{#LH, LHL, HL#}
#LHLL#	{#LL, LHL, HLL, LL#}
#LLLH#	{#LL, LLL, LLH, LH#}
#LLLLH#	{#LL, LLL, LLH, LH#}

Table 8: Examples of 3-substrings for several strings.

This scanning procedure takes linear time in the size of the data—that is, the amount of time it takes to generate substrings from a set of data is directly proportional to the size of the data. This is extremely efficient.

Given k, the learning procedure is very simple. First, the learner sets its initial hypothesis for the grammar to be the set of all possible k-substrings. Note that the size of this set will always be finite, given a finite set of symbols used to create the substrings. For example, for $k = 3$ the initial hypothesis of the learner will be the set S^3_I given below in (50).

(50) $S^3_I = \{$ #H#, #L#, #HH, #HL, #LL, #LH, HHH, HHL, HLH, HLL, LHH, LHL, LLH, LL#, LH#, HL#, HH# $\}$

This means that the learner begins by hypothesizing that *all* forms are ungrammatical. The learner then takes as input examples of the target pattern, keeping track of the k-substrings it observes in the input examples. These *observed k-substrings* are then removed from its hypothesis for the grammar. That is, it considers the observed k-substrings to now be licit. This process is shown for $k = 3$ and examples from the penultimate-or-final H pattern L_{KJ} from Kagoshima Japanese in Table 9.

Table 9 shows the status of the learner after being shown six input strings from the L_{KJ} pattern (in arbitrary order). At each step (i.e., after each new input string), the learner updates its hypothesis by removing all of that string's substrings from its hypothesis. (The notation $S^3_I - X$ means "remove all members of X from S^3_I). For example, in Step 2, after seeing the string #LLHL#, the learner now treats #LL,

Step	String	Hypothesis
0.		S_I^3
1.	#H#	$S_I^3 - \{$#H#$\}$
2.	#LLHL#	$S_I^3 - \{$#H#, #LL, LHL, HL#$\}$
3.	#LHL#	$S_I^3 - \{$#H#, #LL, LHL, HL#, LHL, #LH$\}$
4.	#LLLH#	$S_I^3 - \{$#H#, #LL, LHL, HL#, LHL, #LH, LLL, LLH, LH#$\}$
5.	#LLLLH#	$S_I^3 - \{$#H#, #LL, LHL, HL#, LHL, #LH, LLL, LLH, LH#$\}$
6.	#HL#	$S_I^3 - \{$#H#, #LL, LHL, HL#, LHL, #LH, LLL, LLH, LH#, #HL$\}$
		$= \{$#L#, #HH, HHH, HHL, HLH, HLL, LHH, LL#, HH#$\}$

Table 9: Strictly local learning from a sample of L_{KJ} and where $k = 3$.

#LHL, HL#, LHL, and #LH as licit substrings. Note that the learner *remembers* from step to step the substrings it has seen from previous steps, so the hypothesized grammar of forbidden strings either stays the same or grows smaller with each step. Because the learner only *subtracts* from the hypothesis for the grammar, and never adds to it, means that the learner is *monotonic* (Heinz, 2010b). This means that it only considers increasingly general patterns, and thus avoids the 'subset problem' (Angluin, 1980).

Note that by Step 6, the learner has converged to a grammar that is equivalent to S_{KJ}.[6] That is, it has *generalized* to a grammar that describes an infinite set of strings. Note that the resulting grammar in Table 9 accepts the string #LLLLLHL#, even though this was not given in the input data. This is true for any string in L_{KJ}.

For a target strictly local pattern describable with forbidden substrings of length k, this learning method is guaranteed to converge to a grammar representing exactly the target pattern, given examples that show all of the k-substrings

[6]Techincally, S_{KJ} contains the forbidden 2-substring HH, whereas the output grammar in Table 9 contains only 3-substrings and thus instead contains the 3-substrings #HH, HHH, HHL, LHH, HH#. Note that these 3-substrings are equivalent to the 2-substring HH in that they forbid exactly the set of strings in which a H is immediately followed by another H. This illustrates that any strictly local pattern describable with forbidden k-substrings is also describable with forbidden $k+1$-substrings (Rogers et al., 2013).

in the pattern that are not forbidden (García et al., 1990). Furthermore, as already noted, it is extremely efficient. Importantly, the learner requires knowing k in advance. This is not an uncommon assumption for phonological learners (Hayes and Wilson, 2008). Also, it is a logical necessity: without knowing k in advance, strictly local learning is theoretically impossible (Rogers and Pullum, 2011; Rogers et al., 2013). However, we can posit that k for scanning and learning can be seen as coming from constraints on working memory, e.g. Miller (1956)'s 'magical number' of 7 ± 2 objects. All of the examples here conform to this generalization; a k of at most 4 suffices.

4.3 Melody-local learning

Having established a method for strictly local learning, we can extend it directly to learning constraints on the melody. To learn a melody-local grammar, which consists of both local constraints over a surface string and constraints over its melody, the learner must learn both in parallel. Learning the local constraints proceeds as outlined in the previous section, with the learner removing from its hypothesis S all k-substrings it sees in each string. Learning the melody constraints proceeds in an almost identical fashion. The learner also has a hypothesis M for its melody grammar, and updates this by removing from M all k-substrings it sees *in the melody* of each string.[7] Thus, the melody-local learner follows the following two procedures at each step; that is, for each new data point w.

(51) For each data point w,

a. Update hypothesis S using the k-substrings of w.

b. Update hypothesis M using the k-substrings of $\texttt{mldy}(w)$.

Because we are assuming the OCP, the learner's initial hypothesis for the melody grammar need not contain adjacent sequences of Ls and Hs. Thus, for example, when $k = 3$ the initial state of the hypothesis for the melody grammar, M_I^3, is as follows.

(52) $M_I^3 = \{\#H\#, \#L\#, \#HL, \#LH, LHL, HLH, HL\#, LH\#\}$

[7] It is possible to keep the length of substrings for the local learning and the length of the substrings for the melody learning are distinct, but to simplify the exposition this discussion keeps them the same.

To continue with the example using L_{KJ}, let us set $k = 3$ and consider the same data sample as in Table 9. Again, the local portion of the learning (as per step (51a) above) proceeds exactly as in Table 9. The melody portion of the learning thus proceeds (as per step (51b)) as given in Table 10.

Step	String	Melody	Hypothesis for melody grammar
0.			M_I^3
1.	#H#	#H#	$M_I^3 -$ {#H#}
2.	#LLHL#	#LHL#	$M_I^3 -$ {#H#, #LH, LHL, HL#}
3.	#LHL#	#LHL#	$M_I^3 -$ {#H#, #LH, LHL, HL#}
4.	#LLLH#	#LH#	$M_I^3 -$ {#H#, #LH, LHL, HL#, LH#}
5.	#LLLLH#	#LH#	$M_I^3 -$ {#H#, #LH, LHL, HL#, LH#}
6.	#HL#	#HL#	$M_I^3 -$ {#H#, #LH, LHL, HL#, LH#, #HL}
			= {#L#, HLH}

Table 10: Strictly local melody learning from a sample of L_{KJ} and where $k = 3$.

As shown in Table 10, learning proceeds exactly as strictly local learning, except that the 3-substrings are extracted from the melody of the string instead of the string itself. Thus, for example, at Step 4 in Table 10, the learner is presented with the string LLLH, and thus extracts the 3-substrings from *the melody of* LLLH, that is, LH. Combining both the results in Table 9 and Table 10, we get the grammar in (53).

(53) $G = ($ $S =$ {#L#, #HH, HHH, HHL, HLH, HLL, LHH, LL#, HH#},
$M =$ {#L#, HLH} $)$

This grammar is equivalent to the grammar G_{KJ} given in (37).[8] Thus, a melody-local learner running both a strictly local learning process and a melody-local process in parallel will converge to the correct grammar. It also does so efficiently: searching for both the local substrings and melody substrings in the data

[8]Phonologists may be uncomfortable with the 'duplication' of the forbidden substrings #L# and HLH, which appear both in S and M. However, removing this duplication is trivial—we can safely minimize the grammar by removing any substring in S that also appears in M without changing the pattern described by G. Adding this step to the algorithm would guarantee the learner never returns a grammar with any such duplication.

takes time linear in the size of the data.[9] In order to correctly learn the grammar, the learner must observe all k-factors that are not forbidden. However, as this example has shown, these can be observed in a small number of data points.

Let us look at an example involving an interaction between the local constraints and the constraints on the melody. Recall that in Prinmi, a local constraint against a H span spreading more than two moras combined with a constraint that required exactly one H-tone span per word. Thus, strings like *LHHH are ill-formed, because a H-tone has spread more than two TBUs, and strings like *LLLL and *LHLLH are ill-formed, because they contain more than one H-span. Strings like the set in (54), however, are well-formed.

(54) LHHL, LHLL, LLHL, HH, HHLL, LLLH

The learning procedure outlined in (51) learns a grammar equivalent to the grammar G_{Pr} in (44), given the strings in (54) as data points and a k value of 3. The steps for local learning are given in Table 11, and the steps for melody learning are given in Table 12.

Step	String	Hypothesis for local grammar
0.		S_I^3
1.	#LHHL#	$S_I^3 -$ {#LH,LHH,HHL,HL#}
2.	#LHLL#	$S_I^3 -$ {#LH,LHH,HHL,HL#,LHL,HLL,LL#}
3.	#LLHL#	$S_I^3 -$ {#LH,LHH,HHL,HL#,LHL,HLL,LL#,#LL}
4.	#HH#	$S_I^3 -$ {#LH,LHH,HHL,HL#,LHL,HLL,LL#,#LL,#HH,HH#}
5.	#HHLL#	$S_I^3 -$ {#LH,LHH,HHL,HL#,LHL,HLL,LL#,#LL,#HH,HH#}
6.	#LLLH#	$S_I^3 -$ {#LH,LHH,HHL,HL#,LHL,HLL,LL#,#LL,#HH,HH#}
		= {#L#, HHH, HLH}

Table 11: Learning local grammar for L_{Pr}; $k = 3$.

As $k = 3$, the local portion of the learner begins with the hypothesis S_I^3, the same as for learning L_{KJ}. As the data comes in, it removes the 3-substrings it sees in each data point from this hypothesis. Thus, for example, in Step 1 of Table 11, the learner sees the string LHHL and thus removes the 3-substrings #LH, LHH,

[9] The `mldy` function also takes linear time to calculate.

HHL, and HL# from its hypothesized set of forbidden substrings. We see that by Step 4, it has converged to a grammar that only includes #L#, HHH, and HLH as forbidden substrings. This is not exactly the set given in S_{Pr}in (43) for the local grammar for L_{Pr}—it includes both HHH, which bans spreading beyond more than two TBUs, but also HLH and #L#, which are the result of the culminativitiy and obligatoriness constraints on H. However, this is not incorrect—no string in L_{Pr} contains either as a 3-substring. The learner is thus correct in not removing them from its hypothesis. In general, given data points from L_{Pr}, the learner will never see the 3-substrings HHH, #L#, and HLH and thus will correctly never remove them from its hypothesis.

Step	String	Melody	Hypothesis for melody grammar
0.			M_I^3
1.	#LHHL#	#LHL#	$M_I^3 - \{\text{\#LH, LHL, HL\#}\}$
2.	#LHLL#	#LHL#	$M_I^3 - \{\text{\#LH, LHL, HL\#}\}$
3.	#LLHL#	#LHL#	$M_I^3 - \{\text{\#LH, LHL, HL\#}\}$
4.	#HH#	#H#	$M_I^3 - \{\text{\#LH, LHL, HL\#, \#H\#}\}$
5.	#HHLL#	#HL#	$M_I^3 - \{\text{\#LH, LHL, HL\#, \#H\#, \#HL}\}$
6.	#LLLH#	#LH#	$M_I^3 - \{\text{\#LH, LHL, HL\#, \#H\#, \#HL, LH\#}\}$
			$= \{\text{\#L\#, HLH}\}$

Table 12: Strictly local melody learning from the sample of L_{Pr} from Table 11; $k = 3$.

In terms of melody learning, for the same sample of data the learner will correctly converge to the melody grammar M_{Pr} for L_{Pr} originally given in (44). This is shown in Table 12. As shown in Table 12, the melodies for the strings in (54) are LHL, H, HL, and LH, from which the learner will extract all possible melody 3-substrings except for #L# and HLH. Its hypothesis for the melody grammar thus converges to $\{\text{\#L\#, HLH}\}$, which identical to M_{Pr}. Again, given strings from the pattern L_{Pr}, the learner will never see a string whose melody contains the 3-substrings #L# and HLH, and thus will always consider them forbidden.

Thus, the learning procedure outlined in (51) returns the following grammar given the data in (54).

(55) $$G = \left(\begin{array}{l} S = \{\text{\#L\#, HLH, HHH}\} \\ M = \{\text{\#L\#, HLH}\} \end{array} \right)$$

This grammar is equivalent to the grammar G_{Pr} in (44) for L_{Pr}—all and only the strings in L_{Pr} satisfy the grammar G in (55).[10] The learner thus correctly converges to a grammar for L_{Pr} given a finite set of examples of the pattern.

We have thus far seen, using L_{KJ} and L_{Pr}, how the learning procedure in (51) correctly and efficiently learns melody-local patterns given finite samples of data. While these were only two examples, we can *guarantee* that this will always happen, given a sufficient sample of data. A full proof is given in the Appendix, but the basic idea has already been illustrated in the above examples. If the learner sees all k-substrings, both local and in the melody, that are *not* forbidden in the pattern, then it will correctly remove them from its local and melody hypotheses for the grammar. Conversely, given data points from the sample pattern, it will never any k-substrings that are forbidden either locally or in the melody. It will thus correctly never remove these k-substrings from its local and melody hypotheses for the grammar. This guarantees that, given samples exhibiting all allowed k-substrings in the pattern, it will always converge to the correct grammar.

5 Comparison to previous approaches

Previous approaches to long-distance learning in phonology have appealed to either precedence relations (Heinz, 2010a) or tier-projection (Hayes and Wilson, 2008; Goldsmith and Riggle, 2012; Jardine and Heinz, 2016; McMullin and Hansson, 2015; Jardine and McMullin, 2017). However, neither can learn the full range of tonal patterns discussed in this paper.

We shall focus on tier-projection. Tier-projection implements a notion of relative locality by ignoring some subset of units in the surface string (Heinz et al., 2011). For example, we can capture L_{Ar} (the Arigibi pattern in which two H-toned TBUs are banned) using a tier projection in which L-toned TBUs are ignored. This is depicted schematically in (56).

(56) a.	H ↑ L H L L L	b. *	H H ↑ ↑ L H L L H

In (56a), the tier projected for the string LHLLL is H, whereas for *LHLLH (which is ill-formed according to L_{Ar}) it is HH. We can thus describe all and only

[10] To obtain G_{Pr} exactly, we simply apply the minimization procedure described in Footnote 8.

the strings in L_{Ar} by a forbidden substring grammar {HH} that is interpreted as holding over the tier projection that only includes H-toned TBUs. In terms of formal language theory, such patterns are called tier-based strictly local (Heinz et al., 2011), and are efficiently learnable even if the tier is not specified in advance (Jardine and Heinz, 2016; Jardine and McMullin, 2017). Other models of long-distance phonotactic learning use this basic idea of tier projection, even though it is usually augmented with feature representations and statistical learning methods (Hayes and Wilson, 2008; Goldsmith and Riggle, 2012; Gallagher and Wilson, to appear).

However, it can easily be shown that tier-projection is not sufficient for the other long-distance tone patterns discussed in this paper. First, consider the same tier projection for unbounded tone plateauing pattern, L_{UTP}, in which every string must have at most one span of H-toned TBUs. The examples below in (57) show the tier projection for a string that is well-formed with respect to L_{UTP} (57a) and that for a string that is ill-formed (57b).

(57) a.
```
  H H H
  ↑ ↑ ↑
L H H H L L
```
b. *
```
H H       H
↑ ↑       ↑
H H L L L H
```

As shown in (57a) and (b), a string LHHHLL with a single, unbroken stretch of three H-toned TBUs projects a tier HHH, as does an ill-formed string *HHLLLH with one span of two H-toned TBUs followed by another, distinct span of a single H-toned TBU. Thus, tier projection cannot distinguish between a single plateau and multiple, distinct plateaus, because the intervening L-toned TBUs (or lack thereof) have been ignored.

This same issue occurs in Prinmi, in which words can contain only at most one span of H-tones, as in Arigibi or UTP. However, Prinmi allows spreading of H-tone up to two TBUs, it is possible for there to be more than one H-toned TBU to appear in the string, as shown below in (58a). Thus, we cannot ban HH sequences on a projection of H-toned TBUs.

(58) a.
```
  H H
  ↑ ↑
L H H L L
```
b. *
```
  H     H
  ↑     ↑
L H L L H
```

However, as shown in (58b), ill-formed strings like *LHLLH do contain two Hs on the H-toned TBU tier projection. Thus, a grammar operating over this tier projection cannot distinguish between (58a) and (b). As in L_{UTP}, ignoring the L-toned TBUs loses all 'local' information that distinguishes a single spans of TBUs from distinct spans of single H TBUs. Futhermore, we cannot take the intersection of the local constraint with a constraint over the tier-projection, because they would disagree: the local constraint would allow HH sequences, whereas the tier-projection constraint would forbid them.

This is also the case for L_{Be}, the long-distance constraint blocking unbounded spread in Bemba. The examples in (59) compare the tier projections for strings that are well-formed (59a, b) and ill-formed (59c) with respect to this pattern.

(59)	a.	b.	c.
	H H H	H H	*H H
	↑ ↑ ↑	↑ ↑	↑ ↑
	H H L L L H	L L L H H	H H L L L L

In Bemba, a H tone followed by another H tone must spread to at most two TBUs, as in (59a) HHLLLH. However, a H tone that is *not* followed by another H tone must spread to the end of the word; thus (59c) *HHLLLL is ill-formed. As already established, this is a long-distance pattern, because it requires distinguishing H-tone spans that are followed by another H tone and those that are not, no matter how far to the right the second H-tone span is. However, a tier projection of H-toned TBUs cannot distinguish between all well-formed and ill-formed strings in the language. As shown in (59b) and (c), the tier projection of the ill-formed *HHLLLL is identical to that of a string LLLHH, which is well-formed because its single H-tone span has reached the end of the word. Thus no grammar over this tier projection can distinguish these two strings. Again, removing L-toned TBUs from consideration makes it impossible to distinguish one H-tone span from another—in this case, a word-final H-tone span with a word-medial one. Thus, Bemba is not describable with tier-projection, nor is it describable with the intersection of local constraints with a tier-projection constraint.

To summarize, several of the tone patterns surveyed here are impossible to characterize with grammars that operate over tier projections. This has a consequence for models of phonological learning: tier-projection learners, regardless of whether they are categorical (as in Jardine and Heinz 2016; Jardine and McMullin 2017) or probabilistic (as in Hayes and Wilson 2008; Goldsmith and Riggle 2012; Gallagher and Wilson to appear), are guaranteed to fail to learn these patterns, because these patterns simply cannot be represented by tier projections.

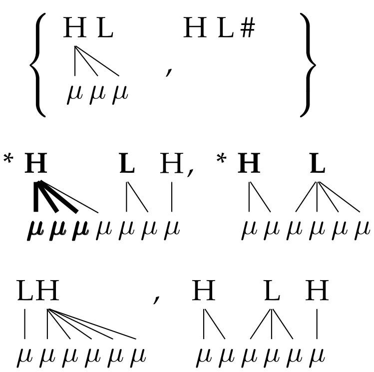
Another option is precedence-based learning a la Heinz (2010a). Indeed, L_{UTP} is describable by precedence grammars (Graf, 2017). However, obligatoriness constraints like Chuave cannot be captured by precedence grammars (Rogers et al., 2013). Furthermore, patterns like Prinmi and Bemba, which also include local generalizations, cannot be captured by precedence grammars, as precedence grammars are necessarily 'blind' to local information (and thus neither can they be captured by Graf (2017)'s interval-based strictly piecewise grammars, which generalize precedence and tier-projection grammars).

Finally, Jardine (2017) presents a class of local AR grammars which can capture a range of local and long-distance constraints and their interactions (Jardine, to appear). These grammars are based on forbidden sub*graphs* of ARs. For example, if we view the Bemba pattern in terms of ARs, we can model the constraints against bounded and unbounded spreading with the forbidden AR substructures in (60a).[11]

(60) a.

b.

c.



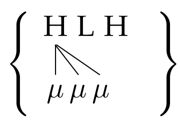
The structures in (60a) specify a nonfinal H tone that has spread more than two TBUs and a melody in which ends in a HL sequence, respectively. As highlighted in bold in (60b), ARs in which a nonfinal H tone has spread more than two TBUs, or in which a final H tone has not spread to the end of the word, contain one of these as substructures. Thus, the grammar in (60a) marks them as ill-formed. As shown in (60c), ARs that are well-formed with respect to the pattern do not contain either substructure.

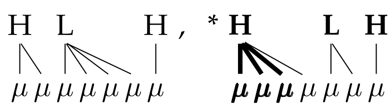
Thus, while forbidden substructure grammars over AR can equally capture these patterns, searching for subgraphs can be computationally tax-

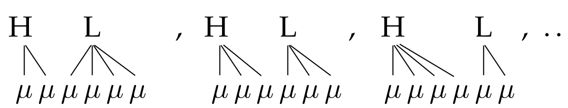
[11]For brevity, this grammar abstracts away from the complete set of constraints that obtain bounded spreading, but these also can be captured with forbidden AR substructures.

ing (Eyraud et al., 2012). While ARs are highly structured (Kornai, 1995; Jardine and Heinz, 2015), it is not currently known whether they can be efficiently searched for observed subgraphs in the same way that strictly local learners search for observed substrings.

Furthermore, forbidden substructure grammars over ARs are less restrictive than melody-local grammars. As in the first substructure in (60a), ARs can refer to both melody information and local associations to TBUs. Thus, these constraints can express arbitrary combinations of non-local and local information. For example, consider the hypothetical pattern Bemba′ as described by the below constraint in (61a).

(61) a. 

b. 

c. 

The constraint in (61a) bans a H tone followed by another from spreading more than two TBUs (61b), but a H tone not followed by another H is allowed to spread arbitrarily (61c).

Patterns like Bemba′ appear to be unattested. It is also not melody-local. This is because the constraint in (61a) refers to both local and melody information: a H tone spreading to three or more mora is, on its own, allowed (as witnessed by (61c)), and a HLH melody is, own, allowed. Only when they coincide in the particular configuration in (61a) are they forbidden. (For a proof that this is not melody-local, see the Appendix.) Thus, melody-local grammars are more restrictive than local grammars over ARs because they constrain the interaction between long-distance and local constraints.

Thus, the melody-local grammars posited by this paper are currently the best approach to characterizing and learning tone, because they capture the range of local and long-distance patterns found in tone with a restrictive, efficiently learnable class of grammars.

6 Discussion

6.1 Autosegmental representations and strings

This paper has represented tone patterns in terms of strings. While, as discussed in §3.1, long-distance constraints utilize the autosegmental concept of the melody, local constraints are represented in terms of strings of H- and L-toned TBUs. This may seem counter to the long-standing idea that tone should be represented in terms of ARs (though see Cassimjee and Kisseberth 1998; Shih and Inkelas forthcoming). However, this is not the case: we can view these string representations as a 'shorthand' for ARs. The following discusses this in more detail, first showing how autosegmental associations between tones and TBUs can be represented in terms of string symbols, and also how contours can be straightforwardly incorporated into the current proposal.

First, Jardine (2017) shows that several major types of tone patterns that have traditionally been treated with autosegmental analyses can be viewed in terms of local constraints over ARs. For many of these patterns, local AR constraints can be recast as strictly local constraints over strings of TBUs. For example, consider the the classic case of directional association in Mende (Leben, 1973, 1978), in which a melody 'expands' to fill a string of TBUs by associating tones to TBUs in a one-to-one, left-to-right fashion. Examples with a HL melody are given below in (62).

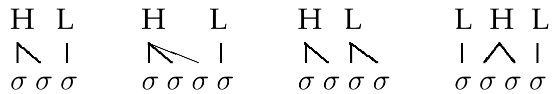
(62) a. HL b. HL c. * H L
| | | \ \ |
σ σ σ σ σ σ σ σ

Left-to-right direction of association in Mende dictates that it is the final tone that spreads; thus (62b) is well-formed but (62c) is ill-formed. Jardine (2017) shows that such directional patterns are describable with a version of strictly local constraints that forbid substructures of ARs. The Mende pattern, for example, is partially describable by the constraint in (63a) forbidding a nonfinal H tone from spreading to more than one TBU.

(63) a. * H L b. *HHL
\ \
σ σ

Continuing with our assumption of the OCP and full specification (though see below), this constraint is equivalent to forbidding a sequence of two H-toned TBUs followed by a L-toned TBU. To see why, consider the following comparison of ARs that violate (63a) with their equivalent representation as strings of TBUs.

(64) AR:



String: HHL HHHL HHLL LHHL

As can be seen in (64), for any AR that contains (63a), its equivalent string of TBUs will contain the substring HHL. This means that we can equally discuss this pattern in terms of (63b), the forbidden substring HHL. In this way, strictly local constraints over strings of TBUs like that in (63b) can serve as a shorthand for autosegmental constraints like that in (63a). In general, the constraints on spreading discussed in Jardine (2017) that result from directional or quality-specific (that is, restricted to a H or L tone; see Zoll, 2003) association are also describable with strictly local constraints over strings of TBUs, and thus are also melody-local. While human learners may indeed be using AR structures to learn, we can at least approximate how they are learning with string grammars.

In fact, there is reason to believe that melody-local grammars more accurately capture the independent behavior of melody constraints and local constraints. As discussed in the previous section regarding the unattested pattern Bemba′, the AR substructure constraints of Jardine (2017, to appear) can simultaneously refer to non-local melody information as well as local associations between tones and TBUs. That melody-local constraints are sufficient to capture the range of patterns discussed here and in Jardine (2017, to appear), *and* that they disclude unattested patterns like Bemba′, is thus evidence that human learners learn melody and local constraints independently. This fact is not predicted by local AR grammars. Thus, while string constraints may capture patterns like Mende less 'naturally', they are more restrictive than local AR constraints and are still expressive enough to capture the range of attested patterns. And, unlike AR constraints, there is a known method for efficiently learning them.

Another phenomenon which ARs capture naturally are contour tones, but these can also be captured with melody-local grammars with slight modifications to the `mldy` function. For example the discussion in Prinmi in §2.4 abstracted away from word-final contours (as noted in Fn. 4). In Prinmi, it is also possible to have a word-final falling- or rising-toned syllable. Below, falling and rising tones are respectively indicated as [â] and [ǎ] on vowels and F and R in the schematic representations.

(65) Prinmi contours (Ding, 2006)

a.	bɨ̂	'honey'	b.	tʃ'ɨ̌	'dog'	c.	dʒjõdʒɨ̌	'buffalo'
	F			R			LR	

Strictly speaking, in terms of strings these forms violate the generalization posited in §2.4 for Prinmi that each form must have at least one span of H-toned TBUs. However, viewed autosegmentally, we can break F and R contours down into HL and LH sequences, respectively, in the melody, as shown in (66) for (65c). Then we can maintain the generalization, expressed by the melody-local grammar in §3.3 for Prinmi, that really the generalization is that there must be exactly one H *in the melody*.

(66) L H

μ μ

dʒjõdʒɨ̌ 'buffalo' (=65c)

We can capture this with a melody-local grammar by adding a function `cntr` that 'expands' contour-toned TBUs (but leaves H and L-toned TBUs as is).[12]

(67)
$$\begin{array}{lcl} \texttt{cntr}(\mathrm{F}) & = & \mathrm{HL} \\ \texttt{cntr}(\mathrm{R}) & = & \mathrm{LH} \\ \texttt{cntr}(\mathrm{LR}) & = & \mathrm{LLH} \end{array}$$

We can then posit that `mldy` operates not on the surface string of TBUs, but on the output of `cntr`.

(68)
$$\begin{array}{lcl cl} \texttt{mldy}(\texttt{cntr}(\mathrm{F})) & = & \texttt{mldy}(\mathrm{HL}) & = & \mathrm{HL} \\ \texttt{mldy}(\texttt{cntr}(\mathrm{R})) & = & \texttt{mldy}(\mathrm{LH}) & = & \mathrm{LH} \\ \texttt{mldy}(\texttt{cntr}(\mathrm{LR})) & = & \texttt{mldy}(\mathrm{LLH}) & = & \mathrm{LH} \end{array}$$

As shown in (68), this obtains the correct autosegmental melody for the contour-toned forms in Prinmi. Because each of the resulting strings in (68) contain exactly one H, they then conform to the melody grammar M_{Pr} for Prinmi from §3.3. Thus, adapting melody-local grammars to accomodate contours is straightforward.

One issue this does not address is the observation that ARs naturally capture the kind of spreading constraints that result in contours. For example, Hyman

[12]We can posit this as a universal, as `cntr` would operate vacuously on all of the pure H- and L-strings of the patterns discussed in §3.3.

(2014) points out that in Aghem, an underlying sequence /HL/ of syllables surfaces as a [HF] sequence, because the H has spread to the following L-toned syllable. This is commonly attested, whereas a mapping of /HL/ to [HR] is unattested or rare. Hyman points out that ARs capture this asymmetry because the No-Crossing Constraint (Goldsmith, 1976; Hammond, 1988) prevents generating a [HR] sequence from an underlying /HL/, whereas feature- or string-based representations of tone cannot make a distinction between the two mappings. While this is an important point, capturing such asymmetries has not been a point of this paper. Instead, what this paper has shown is that such constraints should be independent of melody constraints, and furthermore that this independence is difficult to capture with ARs. Future work can examine how ARs may be recruited to further restrict the local portion of melody-local grammars.

6.2 Other representational issues

So far, much of this paper has assumed a constant `mldy` function operating over strings of L- and H-toned TBUs. In the previous section, this was extended to TBUs with falling and rising and contours. This section explores three further extensions of the function that may vary on a language-specific basis: OCP violations, latent tones, and underspecification. While these phenomena did not bear on the specific results of this paper, it is worth addressing how future work may incorporate them into the current proposal.

First, as already noted, the `mldy` function assumes the OCP in generating melody strings. In an influential paper, Odden (1986) argued against the OCP as a hard universal. However, his arguments focus on the OCP as a constraint on underlying, lexical forms; the only examples of OCP violation *on the surface* listed in Odden (1986) are signaled by phonetic downstep. For example, Odden lists the contrasting APRs in (69) for two nouns in Kishambaa:

(69) OCP violations in Kishambaa (Odden, 1986)

a.	H nyoka 'snake' [nyóká]	b.	HH ngoto 'sheep' [ngóꜜtó]

While different surface pronunciation of the two forms: the first, (69a) 'snake' is pronounced with two level H tones, [nyóká], and (69b) 'sheep' is pronounced with a H followed by a downstepped H; [ngó$^{!}$tó].

If surface OCP violations are indicated by downstep, then they can be detected by the `mldy` function.[13] If downstep marks are included in the inventory of tone symbols, we can modify the function such that H sequences separated by downstep are output as adjacent, yet separate H tones in the resulting melody string.

(70) $\texttt{mldy}(\mathrm{LHH^{!}HH}) = \underset{\mathrm{L}}{\boxed{\mathrm{L}}}\,\underset{\mathrm{H}}{\boxed{\mathrm{H\,H}}}\,^{!}\,\underset{\mathrm{H}}{\boxed{\mathrm{H\,H}}} = \mathrm{LHH}$

Thus, OCP violations that are signaled by downstep can be straightforwardly incorporated into the current proposal. Since the ability of melody-local grammars to capture long-distance processes is based on collapsing stretches of adjacent H-toned TBUs to single H tones on the melody, the melody-local proposal predicts that surface OCP violations should block long-distance processes. Clear surface OCP violations do not play into the long-distance patterns discussed in this paper, but future work could look to see if such cases exist.

It should be noted that downstep does not universally indicate adjacent HH tone sequences. For instance, in Dschang-Bamileke (Tadadjeu, 1974; Hyman, 2011b), a surface H$^{!}$H sequence of TBUs corresponds to a HLH sequence in the melody. The difference between Shambaa and Dschang-Bamileke shows that how the `mldy` function treats downstep must be language-specific. How a learner may discover this is thus an open question; likely, it depends on alternations (see, e.g., Pulleyblank (1986)'s extensive discussion of downstep and alternations). However, like OCP violations, latent tones were not integral to the long-distance patterns in this paper, and thus we can safely separate this learning problem from the question of learning long-distance patterns in tone.

Finally, there is the question of underspecified TBUs—i.e., a distinction between ∅ and L-toned TBUs. This can have an effect on long-distance processes. For example, UTP is attested in Saramaccan (Roundtree, 1972; Good, 2004; McWhorter and Good, 2012), but there is a H/L/∅ distinction in the language, and plateauing only occurs over ∅ TBUs. In terms of the melody, *H∅H sequences are forbidden but HLH sequences are allowed. Given an inventory $\{\mathrm{H}, \mathrm{L}, \varnothing\}$,

[13] Hyman (2011b) lists Dioula Odienne (Braconnier, 1982) as a possible example of tautomorphemic OCP violation not marked by downstep, but he also gives an alternate, OCP-obeying analysis based on underspecification.

melody-local grammars can describe this pattern, and the melody-local learner can learn it.

However, this assumes that the distinction between ∅ and L TBUs is available to the learner. Such representations are likely determined by alternations. For example, in Saramaccan the distinction between L-toned TBUs and ∅-toned TBUs is based on whether or not they show a plateauing alternation, as they are (reported to) be pronounced identically. Thus, as with downstep, learning with language-specific `mldy` functions will work in tandem with the learning of alternations. However, here we have factored out the problem of learning representations. What has been shown here is that *if* we have the right representation, *then* we are guaranteed to learn the kind of long-distance well-formedness patterns that exist in tone. Thus, while melody-local grammars do not completely solved the learning problem in Saramaccan, they are a large step towards solving it.

7 Conclusion

Melody-local grammars provide a restrictive, learnable theory of tonal phonotactics. They form a hypothesis that constraints tone patterns hold either over an autosegmental melody or over local sequences of adjacent TBUs. This hypothesis was shown to be superior to conceptions of long-distance interactions based in tier-projection or unrestricted local grammars over ARs. Thus, this paper has taken an important step towards solving the learning problem by identifying a structural property of tonal patterns that can be used to learn them. This structural property can form the basis of further work on learning the representation of tone, learning tonal processes, and the statistical learning of tone.

Appendix

This appendix collects the formal details of the paper. Standard notation for set theory is used. Let Σ be a fixed, finite alphabet of symbols and Σ^* be the set of all strings over Σ, including λ, the empty string. For a symbol $\sigma \in \Sigma$, σ^n denotes the string resulting from n repetitions of σ. Let $|w|$ indicate the length of a string w. For two strings $w, v \in \Sigma^*$, let wv denote their concatenation (likewise for $w \in \Sigma^*$ and $\sigma \in \Sigma$, $w\sigma$ denotes their concatenation). A *stringset* (or formal language) is a subset of Σ^*; this corresponds to the notion of pattern discussed in §2.1. Let $\rtimes$ and $\ltimes$ represent special boundary symbols not in Σ that represent the beginning and end of words, respectively; thus, $\rtimes w \ltimes$ is the string w delineated with the boundary strings. (These correspond to the # boundary familiar to phonologists.)

A.1 Strictly local grammars and k-factors

A string u is a *k-factor* of w if $|u| = k$ and $w = v_1 u v_2$ for some $v_1, v_2 \in \Sigma^*$; that is, u is a substring of w of length k. The *k-factors* of w are given by the following function $\texttt{fac}_k$:

$$\texttt{fac}_k(w) \overset{\text{def}}{=} \begin{cases} \{u \mid u \text{ is a } k\text{-factor of } \rtimes w \ltimes\} & \text{if } |\rtimes w \ltimes| > k \\ \{\rtimes w \ltimes\} & \text{otherwise} \end{cases}$$

For instance, $\texttt{fac}_3(\text{LHLL}) = \{\rtimes\text{LH}, \text{LHL}, \text{HLL}, \text{LL}\ltimes\}$. We extend $\texttt{fac}_k$ to stringsets in the natural way; i.e. for $L \subseteq \Sigma^*$, $\texttt{fac}_k(L) = \bigcup_{w \in L} \texttt{fac}_k(w)$.

We can now define strictly local grammars. A strictly k-local (SL$_k$) grammar is a set $S \subseteq \texttt{fac}_k(\Sigma^*)$; that is, a subset of all of the possible k-factors that can appear in strings in S. For example, for $\Sigma = \{\text{L}, \text{H}\}$,

$$\texttt{fac}_2(\Sigma^*) = \{\rtimes\text{H}, \rtimes\text{L}, \text{HH}, \text{HL}, \text{LH}, \text{LL}, \text{H}\ltimes, \text{L}\ltimes\}.$$

Then, for example, $S_{\text{alt}} = \{\rtimes\text{H}, \text{HH}, \text{LL}, \text{L}\ltimes\}$ is a SL$_2$ grammar because $\rtimes$H, HH, LL, and L$\ltimes$ are all 2-factors of strings in Σ^* (for example, they are all in $\texttt{fac}_2$(HHLL)).

The for a SL$_k$ grammar S, the stringset described by S, written $L(S)$, is thus the set of strings that contain no k-factors in S; that is,

$$L(S) \overset{\text{def}}{=} \{w \in \Sigma^* \mid \texttt{fac}_k(w) \cap S = \emptyset\}$$

For example,

$$L(S_{\text{alt}}) = \{\text{LH}, \text{LHLH}, \text{LHLHLH}, ...\},$$

that is, the set of strings of alternating Hs and Ls, as this is exactly the set of strings that contain none of the 2-factors in S_{alt}.

A stringset L is thus strictly k-local iff $L = L(S)$ for some SL$_k$ grammar S. We say a stringset is strictly local if it is strictly k-local for some k.

The learning procedure for the class of strictly k-local stringsets amounts to the function $\texttt{SLlearn}_k$ defined as follows. For a finite set $D \subset \Sigma^*$,

$$\texttt{SLlearn}_k(D) \stackrel{\text{def}}{=} \texttt{fac}_k(\Sigma^*) - \texttt{fac}_k(D)$$

That is, $\texttt{SLlearn}_k$(D) returns the set of possible k-factors minus the set of k-factors observed in D. This means that $\texttt{SLlearn}_k$ returns a Strictkly $k-$local grammar consisting of all of the k-factors *not* observed in D. It should be noted that this is a 'batch' conception of the learner, as opposed to the sequential learner presented in the main text. They are equivalent, however. The sequential version of the learner takes some finite sequence of data points d_1, d_2, d_3, ..., d_n and returns, at each data point d_i, $\texttt{SLlearn}_k(\{d_1, d_2, d_3, ..., d_i\})$.

The following theorem asserts the correctness of $\texttt{SLlearn}_k$.

Theorem 1 *For a target strictly k-local stringset L and a sample D of L such that $\texttt{fac}_k(D) = \texttt{fac}_k(L)$, $\texttt{SLlearn}_k(D)$ returns a strictly k-local grammar S such that $L(S) = L$.*

Proof: We show first that $w \in L$ implies $w \in L(S)$ and then that $w \in L(S)$ implies $w \in L$. Since $\texttt{fac}_k(D) = \texttt{fac}_k(L)$, then $\texttt{fac}_k(\Sigma^*) - \texttt{fac}_k(D) = \texttt{fac}_k(\Sigma^*) - \texttt{fac}_k(L)$. Since $S = \texttt{SLlearn}_k(D) = \texttt{fac}_k(\Sigma^*) - \texttt{fac}_k(D)$, then $S = \texttt{fac}_k(\Sigma^*) - \texttt{fac}_k(L)$. Thus for every $w \in L$, $\texttt{fac}_k(w) \cap S = \emptyset$, so $w \in L(S)$.

Because L is a strictly k-local set, there is some strictly k-local grammar S' such that $L(S') = L$. Note that for any string w that if $\texttt{fac}_k(w) \subseteq \texttt{fac}_k(L)$, then $\texttt{fac}_k(w) \cap S' = \emptyset$, and so $\texttt{fac}_k(w) \in L$. Because $S = \texttt{fac}_k(\Sigma^*) - \texttt{fac}_k(L)$. For $w \in L(S)$, then $\texttt{fac}_k(w) \cap (\texttt{fac}_k(\Sigma^*) - \texttt{fac}_k(L)) = \emptyset$ and so $\texttt{fac}_k(w) \subseteq \texttt{fac}_k(L)$. Thus $w \in L(S)$ implies that $w \in L$. $\square$

A.2 Melody-local grammars and their learning

Having defined strictly local stringsets and their learning, we can now define melody-local stringsets.

First, we define the mldy function recursively as follows. For $w \in \Sigma^*$,

$$\texttt{mldy}(w) \stackrel{\text{def}}{=} \begin{cases} \lambda & \text{if } w = \lambda, \\ \texttt{mldy}(v)\sigma & \text{if } w = v\sigma^n,\, v \neq u\sigma \text{ for some } u \in \Sigma^* \end{cases}$$

That is, $\texttt{mldy}(w)$ returns λ if $w = \lambda$, otherwise it returns $\texttt{mldy}(v)\sigma$, where v is the longest string not ending in σ. For example,

$$\begin{aligned} \texttt{mldy}(\text{HHLLLH}) &= \texttt{mldy}(\text{HHLLL})\text{H} \\ &= \texttt{mldy}(\text{HH})\text{LH} \\ &= \texttt{mldy}(\lambda)\text{HLH} \\ &= \lambda\text{HLH} = \text{HLH} \end{aligned}$$

For a stringset $L \subseteq \Sigma^*$ let $\texttt{mldy}(L) = \{\texttt{mldy}(w) \mid w \in L\}$.

A melody strictly k-local grammar M is thus, like a strictly k-local grammar, a subset of the possible k factors of Σ. That is, $M \subseteq \texttt{fac}_k(\Sigma^*)$. The difference is that we interpret a melody strictly k-local grammar using the mldy function. The stringset described by M is as follows:

$$L(M) \stackrel{\text{def}}{=} \{w \in \Sigma^* \mid \texttt{fac}_k(\texttt{mldy}(w)) \cap M = \emptyset\}$$

Thus, for example, if $k = 3$ and $M = \{\text{HLH}\}$, then HHLLLH $\notin L(M)$, because $\texttt{mldy}(\text{HHLLLH}) = \text{HLH}$ and $\texttt{fac}_k(\text{HLH}) \cap M = \{\text{HLH}\}$. However, HLLLL$\in L(M)$, because $\texttt{mldy}(\text{HLLLL}) = \text{HL}$ and $\texttt{fac}_k(\text{HL}) \cap M = \emptyset$.

We can then define a k-melody-local grammar G as a tuple $G(S, M)$ where S is a strictly k-local grammar and M is a melody strictly k-local grammar.[14] The stringset described by G is thus

$$L(G) \stackrel{\text{def}}{=} L(S) \cap L(M),$$

that is, the set of strings that satisfy both S and M. We say a stringset is melody-local if it is k-melody-local for some k.

[14]We could consider grammars where the length of the k-factors in S is different from that of M, but for simplicity, and without loss of generality, we shall consider them to be the same here.

Learning melody-local stringsets is a straightforward extension of learning strictly local stringsets. If we fix k, we can define a learning function that takes an input D and outputs the following result:

$$\texttt{MLlearn}_k(D) \stackrel{\text{def}}{=} \big(\texttt{SLlearn}_k(D), \texttt{SLlearn}_k(\texttt{mldy}(D))\big)$$

That is, $\texttt{MLlearn}_k(D)$ returns a tuple, the first of which is obtained by running a strictly k-local learning on D, the second of which is a melody strictly k-local grammar obtained by running strictly k-local learning on $\texttt{mldy}(D)$. The following theorem asserts the correctness of $\texttt{MLlearn}_k$.

Theorem 2 *For a target k-melody-local stringset L and a sample D of L such that $\mathtt{fac}_k(D) = \mathtt{fac}_k(L)$ and $\mathtt{fac}_k(\texttt{mldy}(D)) = \mathtt{fac}_k(\texttt{mldy}(L))$, $\texttt{MLlearn}_k(D)$ returns a k-melody-local grammar G such that $L(G) = L$.*

Proof: Almost immediate from Thm. 1. If L is k-melody-local, then there is some k-melody-local grammar $G' = (S', M')$ such that $L(G') = L$. Let $G = (S, M)$. Because $\mathtt{fac}_k(D) = \mathtt{fac}_k(L)$ and $\mathtt{fac}_k(\texttt{mldy}(D)) = \mathtt{fac}_k(\texttt{mldy}(L))$, from Thm. 1 we know that $L(S) = L(S')$ and $L(M) = L(M')$. Thus $L(G) = L(G') = L$. □

A.3 Abstract Characterization

We can posit an abstract characterization for melody-local patterns independent of a particular grammar formalism to describe them. This allows us to prove whether or not a pattern is melody-local. We base this off of the abstract characterization of strictly local stringsets. Strictly local stringsets can be characterized by the property of *suffix substitution closure* (Rogers and Pullum, 2011; Rogers et al., 2013), which can be used to prove that a pattern is not strictly local.

Theorem 3 (Suffix substitution closure (Rogers and Pullum, 2011)) *A stringset L is SL_k iff for any string x of length $k-1$ and any strings u_1, u_2, w_1, and w_2,*

$$\text{if } u_1 x u_2 \in L \text{ and } w_1 x w_2 \in L, \text{ then } u_1 x w_2 \in L$$

This means that, for any $u_1 x u_2 \in L$, and for any $w_1 x w_2 \in L$, then, as long as x is of length $k-1$, then we can freely replace u_2 with w_2 and be guaranteed to produce another string in L. For example, for the stringset L_{KJ} (penultimate or final H tone) from the main text, we can set x to be HL (because $k = 3$, x must be of length 2), and u_1, u_2, w_1, and w_2 as in (72).

(71) $L_{\mathrm{KJ}} = \{$H, HL, LHL, LLHL, LLLHL, ..., LH, LLH, LLLH, ...$\}$

(72)
$$\begin{array}{l}
\underbrace{\text{LLLL}}_{u_1}\ \underbrace{\text{LH}}_{x}\ \underbrace{\lambda}_{u_2}\ \in L_{\mathrm{KJ}} \\
\underbrace{\text{L}}_{w_1}\ \underbrace{\text{LH}}_{x}\ \underbrace{\text{L}}_{w_2}\ \in L_{\mathrm{KJ}} \\
\hline
\underbrace{\text{LLLL}}_{u_1}\ \underbrace{\text{LH}}_{x}\ \underbrace{\text{L}}_{w_2}\ \in L_{\mathrm{KJ}}\quad \checkmark
\end{array}$$

Thus, u_1xu_2 is LLLLLH, which is a member of L_{KJ}, and w_1xw_2 is LLHL, which is also a member of L_{KJ}. If we substitute u_2 for w_2 in the former, then we obtain a new string $u_1xw_2 =$ LLLLLHL, which is also in L_{KJ}. We can do this for any x of length 2. Another example is given below in (73) for $x =$ LL.

(73)
$$\begin{array}{l}
\underbrace{\text{LLLL}}_{u_1}\ \underbrace{\text{LL}}_{x}\ \underbrace{\text{HL}}_{u_2}\ \in L_{\mathrm{KJ}} \\
\underbrace{\text{L}}_{w_1}\ \underbrace{\text{LL}}_{x}\ \underbrace{\text{LLLHL}}_{w_2}\ \in L_{\mathrm{KJ}} \\
\hline
\underbrace{\text{LLLL}}_{u_1}\ \underbrace{\text{LL}}_{x}\ \underbrace{\text{LLLHL}}_{w_2}\ \in L_{\mathrm{KJ}}\quad \checkmark
\end{array}$$

To show that a stringset is *not* strictly local, we show that suffix substitution closure fails for some x no matter the size of k. Recall the stringset L_{Ch} (at least one H) from the main text.

(74) $L_{\mathrm{Ch}} = \{$H, HL, LH, HH, LLH, LHL, LHH, HLL, HLH, HHH, LLLH, ...$\}$

If, as in L_{KJ}, we set $k = 3$ and choose the string LL, then L_{Ch} fails suffix substitution closure for $x =$ LL and u_1, u_2, w_1, w_2 chosen as shown in (75).

(75)
$$\begin{array}{l}
\underbrace{\text{L}}_{u_1}\ \underbrace{\text{LL}}_{x}\ \underbrace{\text{H}}_{u_2}\ \in L_{\mathrm{Ch}} \\
\underbrace{\text{H}}_{w_1}\ \underbrace{\text{LL}}_{x}\ \underbrace{\text{L}}_{w_2}\ \in L_{\mathrm{Ch}} \\
\hline
\underbrace{\text{L}}_{u_1}\ \underbrace{\text{LL}}_{x}\ \underbrace{\text{L}}_{w_2}\ \notin L_{\mathrm{Ch}}\quad ✗
\end{array}$$

Because $u_1xw_2 =$ LLLL is not a member of L_{Ch}, L_{Ch} is not strictly 3-local. Furthermore, there is no k for which L_{Ch} is strictly k-local, because we can simply replace x with L^{k-1} ($k-1$ repetitions of L).

(76)
$$\begin{array}{llll}
\underbrace{\text{L}}_{u_1} & \underbrace{\text{L}^{k-1}}_{x} & \underbrace{\text{H}}_{u_2} & \in L_{\text{Ch}} \\
\underbrace{\text{H}}_{w_1} & \underbrace{\text{L}^{k-1}}_{x} & \underbrace{\text{L}}_{w_2} & \in L_{\text{Ch}} \\
\hline
\underbrace{\text{L}}_{u_1} & \underbrace{\text{L}^{k-1}}_{x} & \underbrace{\text{L}}_{w_2} & \notin L_{\text{Ch}} \quad ✗
\end{array}$$

This shows that, no matter what $k-1$, suffix substitution in this case will produce a string LL^{k-1}L, which is not a member of L_{Ch}. Thus, L_{Ch} fails suffix substitution closure for any k. This is a formal version of the intuitive 'scanning' proof given in §2.3, (17).

From the suffix substitution closure characterization of strictly local stringsets, we can posit *melody-dependent suffix substitution closure* as the abstract characterization of melody-local stringsets.

Theorem 4 (Melody-dependent suffix substitution closure (MSSC)) *A stringset L is melody-local iff, for some k,*

a. $\texttt{mldy}(L)$ *is strictly k-local and*

b. for any strings w_1, w_2, u_1, u_2 and for any string x, $|x| = k-1$,

$$w_1xw_2 \in L \text{ and } u_1xu_2 \in L \text{ and } \texttt{mldy}(w_1xu_2) \in \texttt{mldy}(L) \text{ implies } w_1xu_2 \in L$$

Proof: Recall that a stringset is melody-local iff it is describable by some melody-local grammar $G = (S, M)$. Thm. 4 follows directly from the definition of $L(M)$. Thm. 4 follows from suffix substitution closure for $L(S)$ plus the additional requirement that $L(G) = L(S) \cap L(M)$. □

melody-dependent suffix substitution closure adds two conditions on suffix substitution closure. First, Thm. 4a states that $\texttt{mldy}(L)$ (the stringset consisting of the melodies of all strings in L) must be strictly k-local. Second, Thm. 4b adds to the antecedent of the suffix substitution closure implication that $\texttt{mldy}(w_1xu_2)$ must be in $\texttt{mldy}(L)$. As an example, take L_{Ch}. First, note that $\texttt{mldy}(L_{\text{Ch}})$ (given below in (77)), is strictly 3-local, as witnessed by the melody strictly k-local grammar $M_{\text{Ch}} = \{\rtimes\text{L}\ltimes\}$ (i.e., it does not contain the string L).

(77) $\texttt{mldy}(L_{\text{Ch}}) = \{\text{H}, \text{HL}, \text{LH}, \text{HLH}, \text{LHL}, \text{HLHL}, \ldots\}$

It is also then true that L_{Ch} satisfies Thm. 4 for $k = 3$. While L_{Ch} fails the implication in (75) for suffix substitution closure, this implication holds for melody-dependent suffix substitution closure, because $\texttt{mldy}(\text{LLLL}) = \text{L}$ is not a member of $\texttt{mldy}(L_{\text{Ch}})$, and so it does not matter that LLLL $\notin L_{\text{Ch}}$.

(78)
$$\begin{array}{rl}
\underbrace{\text{L}}_{u_1}\ \underbrace{\text{LL}}_{x}\ \underbrace{\text{H}}_{u_2} & \in L_{\text{Ch}} \\
\underbrace{\text{H}}_{w_1}\ \underbrace{\text{LL}}_{x}\ \underbrace{\text{L}}_{w_2} & \in L_{\text{Ch}} \\
\texttt{mldy}(\underbrace{\text{L}}_{u_1}\ \underbrace{\text{LL}}_{x}\ \underbrace{\text{L}}_{w_2}) & \notin \texttt{mldy}(L_{\text{Ch}}) \quad \checkmark \\
\hline
\underbrace{\text{L}}_{u_1}\ \underbrace{\text{LL}}_{x}\ \underbrace{\text{L}}_{w_2} & \notin L_{\text{Ch}}
\end{array}$$

It is thus the case that L_{Ch} satisfies melody-dependent suffix substitution closure.

To give an example that does not, recall the Bemba′ pattern discussed in §5. More explicitly, this is the set $L_{\text{Be}'}$ as follows.

(79) $L_{\text{Be}'}$ is exactly the set that is the union of

a. The set of all L strings; {L, LL, LLL, LLLL ...}

b. The set of strings containing a single H span (i.e. L_{UTP}):

$L_{\text{UTP}} = \{$ L, H, LL, LH, HL, HH, LLL, LLH, LHL, LHH, HLL, HHL, HHH, LLLL, LLLH, LLHL, LLHH, LHLL, LHHL, LHHH, HLLL, HHLL, HHHL, ... $\}$

c. The set of strings containing more than one H span, where all H spans but the last is maximally of length two;

{HLH, HHLH, HLHH, HHLLH, HHLHH, LHHLH, HHLLHH, HHLLLH, ... }

There are no constraints on the melody in this pattern; thus $\texttt{mldy}(L_{\text{Be}'})$ is the full set of alternating strings of Hs and Ls.

(80) $\texttt{mldy}(L_{\text{Be}'}) = \{\text{H}, \text{L}, \text{HL}, \text{LH}, \text{HLH}, \text{LHL}, \text{HLHL}, \ldots\}$

We can show that this fails melody-dependent substitution closure using example strings like those in (61) from the main text.

(81)
$$
\begin{array}{rl}
\underbrace{\text{HHHH}}_{u_1}\ \underbrace{\text{L}^{k-1}}_{x}\ \underbrace{\text{L}}_{u_2} & \in L_{\text{Be}'} \\
\underbrace{\text{L}}_{w_1}\ \underbrace{\text{L}^{k-1}}_{x}\ \underbrace{\text{H}}_{w_2} & \in L_{\text{Be}'} \\
\texttt{mldy}(\underbrace{\text{HHHH}}_{u_1}\ \underbrace{\text{L}^{k-1}}_{x}\ \underbrace{\text{H}}_{w_2}) & \in \texttt{mldy}(L_{\text{Be}'}) \\
\hline
\underbrace{\text{HHHH}}_{u_1}\ \underbrace{\text{L}^{k-1}}_{x}\ \underbrace{\text{H}}_{w_2} & \notin L_{\text{Be}'} \qquad ✗
\end{array}
$$

In this case, u_1xw_2 = HHHHL^{k-1}H, in which a non-final H span has spread more than two TBUs (as in (61b) in the main text). This satisfies the melody constraint (because, e.g., HHLLLH$\in L_{\text{Be}'}$ and so HLH$\in \texttt{mldy}(L_{\text{Be}'})$), but it is not in $L_{\text{Be}'}$, so it fails the implication, for any k. Thus, Bemba′ is not melody-local.

References

Angluin, D. (1980). Inductive inference of formal languages from positive data. *Information and Control*, 45(2):117–135.

Bickmore, L. S. and Kula, N. C. (2013). Ternary spreading and the OCP in Copperbelt Bemba. *Studies in African Linguistics*, 42.

Bickmore, L. S. and Kula, N. C. (2015). Phrasal phonology in Copperbelt Bemba. *Phonology*, 32:147–176.

Braconnier, C. (1982). *Le système du dioula dOdienné, tome 1*. Institut de Linguistique Appliquée, Publication 86. Abijan, Ivory Coast: University of Abidjan, Ivory Coast.

Cassimjee, F. and Kisseberth, C. (1998). Optimal domains theory and Bantu tonology. In Kisseberth, C. and Hyman, L., editors, *Theoretical Aspects of Bantu Tone*, pages 265–314. CSLI.

Chandlee, J., Eyraud, R., and Heinz, J. (2014). Learning Strictly Local subsequential functions. *Transactions of the Association for Computational Linguistics*, 2:491–503.

Chandlee, J. and Heinz, J. (2018). Strictly locality and phonological maps. *LI*, 49:23–60.

Chiośain, M. N. and Padgett, J. (2001). Markedness, segment realization, and locality in spreading. In Lombardi, L., editor, *Segmental phonology in Optimality Theory*, pages 118–156. Cambridge University Press.

de la Higuera, C. (2010). *Grammatical Inference: Learning Automata Grammars*. Cambridge University Press.

Ding, P. S. (2006). A typological study of tonal systems of Japanese and Prinmi: towards a definition of pitch-accent languages. *Journal of Universal Language*, 7:1–35.

Donohue, M. (1997). Tone systems in New Guinea. *Linguistic Typology*, 1:347–386.

Eyraud, R., Janodet, J.-C., and Oates, T. (2012). Learning substitutable binary plane graph grammars. In *Proceedings of the 11th International Conference on Grammatical Inference (ICGI 2012)*, JMLR Workshop and Conference Proceedings, pages 114–128.

Gafos, A. (1996). *The articulatory basis of locality in phonology*. PhD thesis, Johns Hopkins University.

Gallagher, G. and Wilson, C. (to appear). Accidental gaps and surface-based phonotactic learning: a case study of South Bolivian Quechua. *Linguistic Inquiry*.

García, P., Vidal, E., and Oncina, J. (1990). Learning locally testable languages in the strict sense. In *Proceedings of the Workshop on Algorithmic Learning Theory*, pages 325–338.

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

Goldsmith, J. (1976). *Autosegmental Phonology*. PhD thesis, Massachussets Institute of Technology.

Goldsmith, J. and Riggle, J. (2012). Information theoretic approaches to phonological structure: the case of Finnish vowel harmony. *NLLT*, 30(3):859–896.

Goldsmith, J. A. (1990). *Autosegmental & Metrical Theory*. Basil Blackwell, Inc.

Good, J. (2004). Tone and accent in Saramaccan: charting a deep split in the phonology of a language. *Lingua*, 114:575–619.

Graf, T. (2017). The power of locality domains in phonology. *Phonology*, 34:385–405.

Hammond, M. (1988). On deriving the Well-Formedness Condition. *LI*, 19(2):319–325.

Haraguchi, S. (1977). *The Tone Pattern of Japanese: An Autosegmental Theory of Tonology*. Tokyo: Kaitakusha.

Hayes, B. and Wilson, C. (2008). A maximum entropy model of phonotactics and phonotactic learning. *LI*, 39:379–440.

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

Heinz, J. (2010a). Learning long-distance phonotactics. *LI*, 41:623–661.

Heinz, J. (2010b). String extension learning. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 897–906. Association for Computational Linguistics.

Heinz, J., Rawal, C., and Tanner, H. G. (2011). Tier-based strictly local constraints for phonology. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics*, pages 58–64, Portland, Oregon, USA. Association for Computational Linguistics.

Heinz, J. and Rogers, J. (2010). Estimating strictly piecewise distributions. In *Proceedings of the 48th Annual Meeting of the ACL*. Association for Computational Linguistics.

Hirayama, T. (1951). *Kyuusyuu hoogen Onchoo no Kenkyuu (Studies on the Tone of the Kyushu Dialects)*. Tokyo: Gakkai no shinshin-sha.

Hyman, L. (2001). Tone systems. In Haspelmath, M., nig, E. K., Oesterreicher, W., and Raible, W., editors, *Language typology and language universals: an international handbook*, volume 2, pages 1367–1380. Berlin & New York: Walter de Gruyter.

Hyman, L. (2006). Word-prosodic typology. *Phonology*, 23:225–257.

Hyman, L. (2009). How not to do typology: the case of pitch-accent. *Language Sciences*, pages 213–238.

Hyman, L. (2011a). The representation of tone. In Oostendorp, M. V., Ewen, C. J., Hume, E., and Rice, K., editors, *The Blackwell Companion to Phonology*. Blackwell.

Hyman, L. (2011b). Tone: Is it different? In Goldsmith, J. A., Riggle, J., and Yu, A. C. L., editors, *The Blackwell Handbook of Phonological Theory*, pages 197–238. Wiley-Blackwell.

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

Hyman, L. and Katamba, F. X. (2010). Tone, syntax and prosodic domains in Luganda. In Downing, L., Rialland, A., Beltzung, J.-M., Manus, S., Patin, C., and Riedel, K., editors, *Papers from the Workshop on Bantu Relative Clauses*, volume 53 of *ZAS Papers in Linguistics*, pages 69–98. ZAS Berlin.

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

Jardine, A. (2017). The local nature of tone-association patterns. *Phonology*, 34:385–405.

Jardine, A. (to appear). The expressivity of autosegmental grammars.

Jardine, A., Chandlee, J., Eyraud, R., and Heinz, J. (2014). Very efficient learning of structured classes of subsequential functions from positive data. In *Proceedings of the 12th International Conference on Grammatical Inference (ICGI 2014)*, JMLR Workshop Proceedings, pages 94–108.

Jardine, A. and Heinz, J. (2015). A concatenation operation to derive autosegmental graphs. In *Proceedings of the 14th Meeting on the Mathematics of Language (MoL 2015)*, pages 139–151, Chicago, USA. Association for Computational Linguistics.

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

Jardine, A. and McMullin, K. (2017). Efficient learning of tier-based strictly k-local languages. In Drewes, F., Martín-Vide, C., and Truthe, B., editors, *Language and Automata Theory and Applications, 11th International Conference*, Lecture Notes in Computer Science, pages 64–76. Springer.

Jenks, P. and Rose, S. (2011). High tone in Moro: effects of prosodic categories and morphological domains. *Natural Language and Linguistics Theory*, 29:211–250.

Kisseberth, C. and Odden, D. (2003). Tone. In Nurse, D. and Philippson, G., editors, *The Bantu Languages*. New York: Routledge.

Kornai, A. (1995). *Formal Phonology*. Garland Publication.

Kubozono, H. (2012). Varieties of pitch accent systems in Japanese. *Lingua*, 122:1395–1414.

Lai, R. (2015). Learnable versus unlearnable harmony patterns. *LI*, 46:425–451.

Leben, W. R. (1973). *Suprasegmental phonology*. PhD thesis, Massachussets Institute of Technology.

Leben, W. R. (1978). The representation of tone. In Fromkin, V., editor, *Tone—A Linguistic Survey*, pages 177–219. Academic Press.

McMullin, K. and Hansson, G. O. (2015). Locality in long-distance phonotactics: evidence for modular learning. In *NELS 44, Amherst, MA*. in press.

McNaughton, R. and Papert, S. (1971). *Counter-Free Automata*. MIT Press.

McWhorter, J. and Good, J. (2012). *A Grammar of Saramaccan Creole*. De Gruyter Mouton.

Meyers, S. (1997). OCP effects in Optimality Theory. *NLLT*, 15(4):847–892.

Miller, G. A. (1956). The magical number seven, plus or minus two: Some limits on our capacity for processing information. *Psychological Review*, 63(2):81–97.

Odden, D. (1986). On the role of the Obligatory Contour Principle in phonological theory. *Language*, 62(2):353–383.

Odden, D. (1994). Adjacency parameters in phonology. *Language*, 70(2):289–330.

Prince, A. and Smolensky, P. (1993). Optimality Theory: Constraint interaction in generative grammar. *Rutgers University Center for Cognitive Science Technical Report*, 2.

Prince, A. and Smolensky, P. (2004). *Optimality Theory: Constraint Interaction in Generative Grammar*. Blackwell Publishing.

Pulleyblank, D. (1986). *Tone in Lexical Phonology*. Dordrecht: D. Reidel.

Rabin, M. O. (1963). Probabilistic automata. *Information and Control*, (6):230–245.

Rogers, J., Heinz, J., Fero, M., Hurst, J., Lambert, D., and Wibel, S. (2013). Cognitive and sub-regular complexity. In *Formal Grammar*, volume 8036 of *Lecture Notes in Computer Science*, pages 90–108. Springer.

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

Roundtree, S. C. (1972). Saramaccan tone in relation to intonation and grammar. *Lingua*, 29:308–325.

Scobbie, J. M., Coleman, J. S., and Bird, S. (1996). Key aspects of declarative phonology. In Durand, J. and Laks, B., editors, *Current Trends in Phonology: Models and Methods*, pages 685–709. Manchester, UK: European Studies Research Institute (ESRI), University of Salford.

Shih, S. and Inkelas, S. (forthcoming). Autosegmental aims in surface optimizing phonology. Ms. available at lingbuzz/002520.

Tadadjeu, M. (1974). Floating tones, shifting rules, and downstep in Dschang-Bamileke. In Leben, W. R., editor, *Papers from the 5th Annual Conference on African Linguistics*, Studies in African Linguistics, Supplement 5.

Yip, M. (2002). *Tone*. Cambridge University Press.

Zoll, C. (2003). Optimal tone mapping. *LI*, 34(2):225–268.