Autosegmental Input Strictly Local Functions

Jane Chandlee
Tri-College Department of Linguistics
Haverford College
jchandlee@haverford.edu

Adam Jardine
Department of Linguistics
Rutgers University
adam.jardine@rutgers.edu

Abstract

Autosegmental representations (ARs; Goldsmith, 1976) are claimed to enable local analyses of otherwise non-local phenomena (Odden, 1994). Focusing on the domain of tone, we investigate this ability of ARs using a computationally well-defined notion of locality extended from Chandlee (2014). The result is a more nuanced understanding of the way in which ARs interact with phonological locality.

1 Introduction

Autosegmental representations (ARs; Goldsmith, 1976) have long been claimed to capture non-local processes in a local way (Odden, 1994). Focusing on tone, we explore this claim from a computational perspective using a precisely defined notion of locality. Specifically, we use *quantifier-free (QF) first-order (FO) logical transductions*, which have been shown (Chandlee and Lindell, in prep.) in strings to correspond to the *input strictly local* (ISL) functions (Chandlee, 2014), a phonologically relevant notion of computational locality (Chandlee and Heinz, 2018).

We extend QF transductions to ARs to define the *autosegmental input strictly local* (A-ISL) functions and give a partial abstract characterization of this class. We then examine a variety of commonly attested tone patterns that are both ISL and not ISL (i.e., local and non-local in terms of strings) to see if they are A-ISL (i.e., local over ARs). Our exploration reveals a four-way division among tone patterns: those that are local without ARs (i.e., both ISL and A-ISL), those that are local only with ARs (i.e., A-ISL but not ISL), those that are local with strings but not ARs (ISL but not A-ISL), and those that are not local at all (neither ISL nor A-ISL). The conclusion is that ARs do not automatically make non-local processes local, but that this ability is more nuanced. We also briefly explore an alternative notion of locality that extends A-ISL.

The remainder of the paper is structured as follows. In §2 we review ARs, and in §3 we provide the mathematical preliminaries. In §4 we define the ISL functions. §5 extends that characterization to ARs. §6 presents analyses for a selection of tone processes, and §7 discusses the significance of these results and open questions. §8 concludes.

2 Autosegmental Representations

In ARs (Clements, 1976; Goldsmith, 1976) phonological primitives are arranged in distinct strings or *tiers*, with an *association* relation relating units on different tiers. Examples from Arusa (Levergood, 1987; Odden, 1994) are shown in Figure 1, both as strings and as ARs. Here and throughout the paper, an acute accent [á] indicates a high tone; unmarked vowels bear low (L) tones.

In the underlying representation (UR) on the left-hand side of Figure 1, the second vowel of /olórika/ 'chair' bears a H tone, and both vowels of /sídáy/ 'good' bear a H tone. The AR for /sídáy/ 'good' shows a single H on the "autonomous" tonal tier that is associated to both vowels on the vowel tier. Tones associate to *tone-bearing units* (TBUs), which could be vowels, moras, or syllables depending on the language (Yip, 2002).

ARs have been argued to provide natural accounts of many tone and segmental processes, particularly non-local ones. To illustrate, when preceded by the word [olórika] 'chair' in the phrase [olórika siday] 'good chair', /sídáy/ 'good' is pronounced instead [siday], with two low tones. This is explained through a *map* in which the underlying H tone is deleted following another H tone in the AR. This is shown schematically in Figure 1. In terms of the string, this is a non-local process: The tone of the second vowel in /olórika/ affects the tone of the vowels in /sídáy/, which are two TBUs away. But in the AR, the process is

Transactions of the Association for Computational Linguistics, vol. 7, pp. 157–168, 2019. Action Editor: Eric Fosler-Lussier.
Submission batch: 8/2018; Revision batch: 12/2018; Published 4/2012.

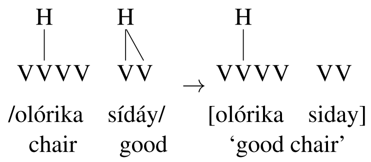


Figure 1: AR map from Arusa.

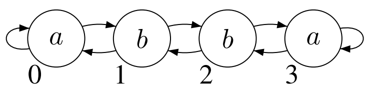


Figure 2: String model for $abba$. Upper curved arrows depict s, lower curved arrows depict p, and the labels on the nodes depict P_a and P_b.

"local" because the H tones are adjacent on the tonal tier.

The computational properties of ARs have been studied before (Kay, 1987; Wiebe, 1992; Bird and Ellison, 1994; Kornai, 1995; Jardine, 2017). However, tone *maps*, such as the one exemplified in Figure 1, have not yet been studied under a computational notion of locality.

3 Preliminaries

3.1 Strings

Let Σ be a finite alphabet of symbols and Σ^* the set of all strings over Σ. Let $|w|$ indicate the length of $w \in \Sigma^*$. For two strings w and v let wv be their concatenation, and for a set $L \subseteq \Sigma^*$ of strings and a string w, by wL we denote $\{wv | v \in L\}$.

The *n-suffix* of a string w is the last n symbols of w: $\texttt{Suff}^n(w) = v$ such that $uv = w$ and $|v| = n$. The *prefixes* of a string $w \in \Sigma^*$, $\texttt{prfs}(w) \stackrel{\text{def}}{=} \{u \mid w = uv \in L \text{ for some } v \in \Sigma^*\}$. The *common prefixes* of a set $L \subseteq \Sigma^*$, $\texttt{cmprfs} \stackrel{\text{def}}{=} \bigcap_{w \in L} \texttt{prfs}(w)$. The *longest common prefix* of a set $L \subseteq \Sigma^*$, is $\texttt{lcp}(L) \stackrel{\text{def}}{=} u$ such that $u \in \texttt{cmprfs}(L)$ and $\forall u' \in \texttt{cmprfs}(L)$, $|u| \geq |v|$.

3.2 Models

We consider finite models (Libkin, 2004). A *model* is a tuple $\langle D; f_1, ..., f_n, R_1, ..., R_m \rangle$ where D is a finite *domain* of elements, $f_1, ..., f_n$ are a set of functions over the domain, and $R_1, ..., R_m$ are a set of relations over the domain. (We do not use models with constants.) We assume functions to be unary and relations either unary or binary.

We can talk of a set of models of the same *signature*, where signature refers to a set $\mathcal{S} = \{f_1, ..., f_n, R_1, ..., R_n\}$ of named functions and relations. In particular, we consider models of strings of the signature $\{p, s, P_{\sigma \in \Sigma}\}$, where p and s are the *predecessor* and *successor* functions, respectively, indicating the order of positions in the string, and for every $\sigma \in \Sigma$ there is a unary relation P_σ indicating the labels of each position.

Figure 2 shows the example of the string $abba$ represented as a model $\langle \{0, 1, 2, 3\}; p, s, P_a, P_b \rangle$ where: $p(0) = 0$; for $i \in \{1, 2, 3\}$, $p(i) = i-1$; for $i \in \{0, 1, 2\}$, $s(i) = i+1$; $s(3) = 3$; $P_a = \{0, 3\}$; and $P_b = \{1, 2\}$. The first position is its own predecessor and the last is its own successor, to encode word boundaries.

3.3 Logics and Transductions

We define transductions from models to models using logical interpretations (Courcelle, 1994; Engelfriet and Hoogeboom, 2001; Filiot and Reynier, 2016). A fixed signature induces a FO logical language $L_\mathcal{S}$ where each variable x is a term, $f_i(x)$ for any function $f_i \in \mathcal{S}$ is a term, and $R_i(t_1, ..., t_k)$ for any $R_j \in \mathcal{S}$ and terms $t_1, ..., t_k$ is an atomic formula. We also allow $t_1 \approx t_2$, representing equality, as an atomic formula. We define $L_\mathcal{S}$ in the usual way. By $\varphi(x_1, ..., x_k)$ we denote a formula φ in $L_\mathcal{S}$ such that $x_1, ..., x_k$ are free variables in φ. For $\varphi(x_1, ..., x_k)$ and a model M in the signature, we write $M \models \varphi(d_1, ..., d_k)$ when $\varphi(x_1, ..., x_k)$ is true in M when $x_1, ..., x_k$ are evaluated to $d_1, ..., d_k$ in D. For an input signature $\mathcal{I}$ and an output signature $\mathcal{O}$, a *logical transduction* τ is a definition of each function and relation in $\mathcal{O}$ in $\mathcal{I}$ as follows:[1]

- A unary predicate $\varphi_D(x)$ defined in $L_\mathcal{I}$;
- For each function $f \in \mathcal{O}$, a definition $f(x) \approx y \stackrel{\text{def}}{=} \varphi_f(x, y)$ for some $\varphi_f(x, y)$ in $L_\mathcal{I}$;
- For each unary relation $P \in \mathcal{O}$, a definition $P(x) \stackrel{\text{def}}{=} \varphi_P(x)$, for some $\varphi_P(x)$ in $L_\mathcal{I}$;
- For each binary relation $R \in \mathcal{O}$, a definition $R(x, y) \stackrel{\text{def}}{=} \varphi_R(x, y)$, for some $\varphi_R(x, y)$ in $L_\mathcal{I}$.

[1] These definitions are usually relativized over a *copy set*, such that the output model may be larger than the input model, but we abstract away from this.

For an input structure M_I with domain D over the signature $\mathcal{I}$, its output $\tau(M_I) = M_O = \langle D'; f_1, ..., f_n, R_1, ..., R_m\rangle$ where

- For each $d \in D$ there is a copy $d' \in D'$ iff $M_I \models \varphi_D(d)$.
- For copies $d'_1, d'_2 \in D'$ of some $d_1, d_2 \in D$ and for each function $f \in \mathcal{O}$, $f(d'_1) = d'_2$ iff $M_I \models \varphi_f(d_1, d_2)$.
- For copy $d' \in D'$ of some $d_1 \in D$ and for each unary relation $P \in \mathcal{O}$, $d' \in P$ iff $M_I \models \varphi_P(d)$.
- For copies $d'_1, d'_2 \in D'$ of some $d_1, d_2 \in D$ and for each binary relation $R \in \mathcal{O}$, $(d'_1, d'_2) \in R$ iff $M_I \models \varphi_R(d_1, d_2)$.

For example, we can define a string transduction that takes a string w over $\Sigma = \{a, b\}$ and returns a string w' over $\Sigma' = \{a, b, c\}$ that is identical to w except that every b following another b is rewritten as a c. Let $\mathcal{S}' = \{p', s', P'_a, P'_b, P'_c\}$ be the signature for strings over Σ'. Then let τ be defined as $\varphi_D(x) \stackrel{\text{def}}{=} \texttt{True}$, $\varphi_{s'}(x,y) \stackrel{\text{def}}{=} s(x) \approx y$ $\varphi_{p'}(x,y) \stackrel{\text{def}}{=} p(x) \approx y$, $\varphi_{P'_a}(x) \stackrel{\text{def}}{=} P_a(x)$, $\varphi_{P'_b}(x) \stackrel{\text{def}}{=} P_b(x) \land \neg P_b(p(x))$, and $\varphi_{P'_c}(x) \stackrel{\text{def}}{=} P_b(x) \land P_b(p(x))$, where $\texttt{True}$ is any unary predicate that is always true for models in $\mathcal{S}$.

In this transduction, $\varphi_{P'_b}(x)$ is only true for input positions labeled b that do *not* follow another b; conversely, $\varphi_{P'_c}(x)$ is only true for input positions labeled b that do follow another b. Thus, for example, the output under τ for the string model for $abba$ from Figure 2 is a model for $abca$ as shown in Figure 3.

As $\varphi_D(x)$ is set to $\texttt{True}$, τ copies every input element. Furthermore, the definitions of $\varphi_{s'}(x,y)$ and $\varphi_{p'}(x,y)$ mean that p' and s' are identical to s and p in the input. Similarly, $\varphi_{P'_a}(x)$ is defined to mirror P_a. Thus, the only change is made to bs following other bs, which are written out as cs.

In the definitions that follow, we will often omit formulas for relations and functions whose definition is $\texttt{True}$ or identical to their input formula (as in $\varphi_D(x)$, p', and s' in the above example).

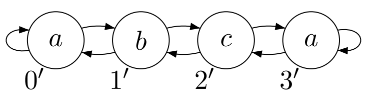


Figure 3: Output model under τ as defined above given the model from Figure 2, where each element i' is a copy of i from Figure 2.

4 Input Strictly Local String Transductions

We now connect logical transductions to the ISL class. A proper subset of the regular relations, the ISL functions determine an output string for a given input string based only on contiguous substrings of bounded length (Chandlee, 2014; Chandlee et al., 2014). This means that the computation of the output string is based on a very limited amount of information present in the input, and that information is locally contained (i.e., within a bounded window around the current position) rather than "global" in nature (i.e., referring to some property of the entire string). This limitation on the information available to the function is what establishes ISL as a proper subclass of the regular relations (i.e., not all string-to-string relations can be computed with such limited information). Despite its reduced expressivity, ISL can model a significant range of local segmental phonological maps and has been argued to provide a well-defined notion of phonological locality (Chandlee and Heinz, 2018).

Chandlee (2014) gives the ISL functions an abstract characterization based on the notion of *tails*. Informally, the tails of a string x with respect to a function f is a set of strings (y, v) where y is a possible "input extension" of x and v is the contribution to the output string that y itself is responsible for. Formally: $\texttt{tails}_f(x) = \{(y, v) \mid f(xy) = uv \land u = \texttt{lcp}(f(x\Sigma^*))\}$. The abstract characterizations of ISL functions is given in Definition 1.

Definition 1 (Input Strictly Local Function) *A function f is ISL iff there is a k such that for all $u_1, u_2 \in \Sigma^*$, if $\texttt{Suff}^{k-1}(u_1) = \texttt{Suff}^{k-1}(u_2)$ then $\texttt{tails}_f(u_1) = \texttt{tails}_f(u_2)$.*

If two strings share a $(k-1)$-suffix but have different sets of tails, the function is not k-ISL, a fact we use in some of the analyses to follow.

Chandlee and Lindell (in prep.) provide a logical characterization of ISL functions as those functions which can be described by quantifier-free (QF) FO string transductions. A QF transduction τ is a FO transduction such that no quantifiers appear in any formulas in τ (as in the example in the previous section).

QF logical transductions can be applied not only to strings but to models in general. We can thus use them as a general notion of locality to compare strings and ARs. We first view tone patterns as

ISL string functions, then use QF transductions in §5 to define A-ISL functions over ARs and in §6 to compare and contrast ISL with A-ISL with respect to a variety of tone patterns.

4.1 Bounded Tone Shift is ISL

We thus first illustrate the ISL class of local string functions by analyzing some example tone patterns. To do this, we analyze these patterns as functions operating over strings of TBUs.

In all of the analyses that follow, the goal is to isolate tone processes in order to classify them individually in terms of the notions of ISL/A-ISL. In some cases this isolation requires certain restrictions on the input forms, and we will note those assumptions in each case. We will briefly discuss in §7 how the interaction of multiple tone processes can affect computational classification, though a complete answer to that question will ultimately be left for future work.

First, we contrast *bounded* shift and *unbounded* shift. In bounded shift a tone appears some fixed number of TBUs away from its underlying position. In Rimi (Meyers, 1997), a tone shifts one TBU to the right. In Example (1), /−a/, /−mu/, and /−i̧/ have a L tone but surface with an H when the preceding vowel is underlyingly high.

(1) Rimi (Schadeberg, 1979; Meyers, 1997)
 a. /u-pų́m-a/ $\mapsto$ [u-pųm-á] 'to go away'
 b. /rá-mu-ntu/ $\mapsto$ [ra-mú-ntu] 'of a person'
 c. /mu-tém-i̧/ $\mapsto$ [mu-tem-í̧] 'chief'

We can represent this pattern as a string transduction with an alphabet $\Sigma = \{\varnothing, \text{H}\}$, where $\varnothing$ is an "unspecified" or toneless TBU (as in the first vowel in Example (1a)) and H is a high-toned TBU.[2] Using this notation, the examples from (1) are as below in Example (2).

(2) a. $\varnothing\text{H}\varnothing \mapsto \varnothing\varnothing\text{H}$
 b. $\text{H}\varnothing\varnothing \mapsto \varnothing\text{H}\varnothing$

In general, then, the Rimi case is the function below. Here, to isolate the shifting process we restrict our domain to strings with one underlying H, as in Rimi sequences of successive H tones are subject to other rules (see Meyers, 1997). This also assumes a final H does not shift.

$$\varnothing^m\text{H}\varnothing^n \mapsto \varnothing^{m+1}\text{H}\varnothing^{n-1} \text{ for } n > 0,$$
$$\varnothing^m\text{H} \mapsto \varnothing^m\text{H}$$

This is easily defined with the QF transduction below, where the predicates $\varnothing(x)$ and $\text{H}(x)$ refer to input position labels and $\varnothing'(x)$ and $\text{H}'(x)$ label the output positions. $\texttt{first}(x) \stackrel{\text{def}}{=} p(x) \approx x$ and $\texttt{last}(x) \stackrel{\text{def}}{=} s(x) \approx x$, and the successor/ predecessor functions are unchanged from the input: $s'(x) \stackrel{\text{def}}{=} s(x)$ and $p'(x) \stackrel{\text{def}}{=} p(x)$.

$$\begin{aligned}\varnothing'(x) &\stackrel{\text{def}}{=} \varnothing(p(x)) \vee \texttt{first}(x) \wedge \\ &\qquad \neg(\texttt{last}(x) \wedge \text{H}(x)), \\ \text{H}'(x) &\stackrel{\text{def}}{=} (\text{H}(p(x)) \wedge \neg\texttt{first}(x)) \wedge \\ &\qquad \vee(\texttt{last}(x) \wedge \text{H}(x))\end{aligned}$$

For an input position x, its corresponding output position is labled $\varnothing$ only if x's predecessor is labeled $\varnothing$ or if x was the first position (and not the final H). Conversely, x's corresponding output position is labeled H if its predecessor is an H or if x is the last position and is an H. In this way, the "shifting" behavior is captured by determining the output label of each position via the predecessor of its corresponding input position. Thus, bounded tone shift is QF-definable and so ISL.

4.2 Unbounded Tone Shift is Not ISL

In contrast, *unbounded* tone shift is not ISL. In unbounded shift, underlying tones move some arbitrary distance to another fixed position. One example is in Zigula.

(3) Zigula (Kenstowicz and Kisseberth, 1990)
 a. ku-gulus-a 'to chase'
 b. ku-lombéz-a 'to ask'
 c. ku-lombez-éz-a 'to ask for'
 d. ku-lombez-ez-án-a 'to ask for e. o.'

The contrast between Example (3a) and (3b) shows that roots must underlyingly contrast in tone. However, this tone does not appear in this underlying position, but instead shifts to the penultimate TBU, as demonstrated dramatically by Example (3d).

Following the string representation scheme, we can represent this as the following function. Here,

[2] This abstracts away from consonants, consistent with the assumption that tone operates over prosodic units and usually does not interact with segmental information (Yip, 2002).

we restrict the domain to strings with at most one H (as, like in Rimi, multiple H tones trigger other processes; interested readers are referred to Kenstowicz and Kisseberth, 1990).

$$\varnothing^n \mapsto \varnothing^n$$
$$\varnothing^m \mathrm{H} \varnothing^n \mapsto \varnothing^{m+n-1} \mathrm{H} \varnothing$$

There is no QF transduction for this map.

Lemma 1 *The Zigula map is not ISL.*

Proof: Let f be the Zigula map. For any k, and for any $n \geq k$, the strings $\mathrm{H}\varnothing^n$ and $\varnothing^n$ have the same $(k-1)$-suffix: $\varnothing^{k-1}$. However, their tails differ. First, $\mathtt{lcp}(f(\mathrm{H}\varnothing^n\Sigma^*)) = \varnothing^{n-1}$, because $f(\mathrm{H}\varnothing^n\lambda) = \varnothing^{n-1}\mathrm{H}\varnothing$ but $f(\mathrm{H}\varnothing^n\varnothing) = \varnothing^n\mathrm{H}\varnothing$; the longest prefix these outputs share is $\varnothing^{n-1}$. Thus $(\lambda, \mathrm{H}\varnothing) \in \mathtt{tails}_f(\mathrm{H}\varnothing^n)$. However, clearly $(\lambda, \mathrm{H}\varnothing) \notin \mathtt{tails}_f(\varnothing^n)$. □

These two examples show how the ISL property can be used to formalize the notion of ''locality'' in tone. The shift pattern in Rimi is ISL, whereas the shift pattern in Zigula is not ISL.

5 Autosegmental ISL Transductions

In this section we demonstrate how using an AR model instead of a string model allows us to capture some phenomena with QF transductions when we wouldn't otherwise be able to do so. First, we define AR models.

5.1 Autosegmental Models

Autosegmental models include two strings, a string of TBUs (e.g., vowels, syllables, moras) and a string of tones. The positions of these strings are related to each other by the successor and predecessor functions, just as in string models. In addition, the association relation relates positions from the tone string to positions of the TBU string. We thus have the signature in Example (4), where D is the domain, p and s are the successor functions, and A is the association relation.

(4) $\langle D; p, s, A, P_H, P_V \rangle$

An example is shown in Figure 4, with an explicit model for the underlying Arusa phrase /olórika sídáý/ 'good chair' from Figure 1 (for the sake of simplicity, word boundaries have been omitted).

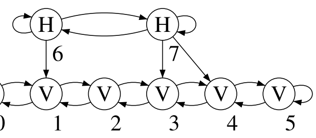
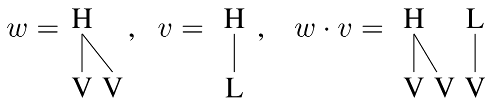


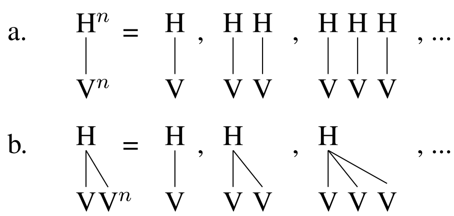
Figure 4: Example of an autosegmental model.

Figure 4 depicts a model with the domain $\{0, 1, 2, 3, 4, 5, 6, 7\}$ and the association relation $a = \{(6, 1), (7, 3), (7, 4)\}$ relating tones to their TBUs. Whereas some authors define association as symmetric (Kornai, 1995), to simplify the formulas in our analyses we define it as antisymmetric, although this assumption has no bearing on QF-definability. Note that association only holds *between* the tiers: Elements on the same tier cannot be associated. Also, for ease of reading we will represent AR models in the remainder of the paper as originally presented in Figure 1 (i.e., without the nodes, position numbers, successor relation, and predecessor relation pictured).

Furthermore, we use notation following that for strings. First, for any ARs w and v, we denote by $w \cdot v$ the concatenation of w and v; that is, the concatenation of each tier of w to its corresponding tier in v, preserving all association lines (see Example (5)).

(5)


This allows us to use the usual conventions from strings: For an AR w, w^n represents the AR consisting of n repetitions of w; w^0 is the empty AR whose tiers are both λ and thus has no association lines. Let Γ^* refer to the set of all ARs. We also use the abbreviations below in Example (6),

(6)


We will classify patterns that can be analyzed with QF FO formulas as *A-ISL* when the model is an autosegmental model instead of a string model.

Definition 2 *An input–output map is A-ISL if it can be described with a QF transduction where the model is an AR model.*

The structure of ARs and the nature of QF transductions gives us the following necessary (but not sufficient) property of A-ISL maps.

Theorem 1 *If a AR map is A-ISL, then the individual map on each tier is an ISL function.*

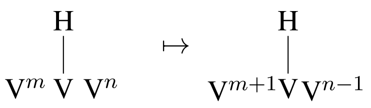
Proof: (Sketch.) For any tier w in an AR, the output for each position in w will depend on some QF formula $\varphi(x)$ with exactly one free variable. Because $\varphi(x)$ is QF, no other variables can be introduced, thus for any $A(t_1, t_2)$ that appears in $\varphi(x)$, both t_1 and t_2 must be of the form $p(...p(x))$ or $s(...s(x))$; that is, t_1 and t_2 represent members of the same tier. Since members of the same tier are never associated to each other, $A(t_1, t_2)$ will always evaluate to `False`. This essentially reduces $\varphi(x)$ to a QF formula over the string signature for that tier. □

As every map that is A-ISL satisfies Theorem 1, we use it below to prove some maps are not A-ISL.

However, although Theorem 1 is a *necessary* condition for A-ISL maps, it is not a *sufficient* condition. In §6.5 we show that an unbounded spreading map satsifies Theorem 1 but is not A-ISL. Briefly, this is because defining the associations in unbounded spreading cannot be defined with QF formulas. Thus, A-ISL maps are also restricted in how associations can change from input to output. While a full abstract characterization of A-ISL must also express these constraints, for concerns of space we save this for future work.

5.2 Bounded Tone Shift is A-ISL

Rimi bounded tone shift was already analyzed as ISL in §4.1. It is also A-ISL. When viewed as a function from ARs to ARs, it is as given below.



$(n > 0)$

Figure 5 shows an AR model representing the map from Example (1a).

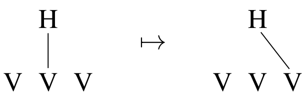


Figure 5: AR representation of bounded shift in Rimi.

We can define this map with a QF transduction. As nothing changes with respect to position labeling or the successor and predecessor functions for either string, we only give the association relation.

(7) $$A'(x, y) \overset{\text{def}}{=} (A(x, p(y)) \wedge \neg\texttt{first}(y)) \vee (\texttt{last}(y) \wedge A(x, y))$$

Informally, the formula in (7) states that positions x and y are associated in the output model if x is associated to the predecessor of y in the input model, or if x is associated to y in the input and y is final (because, as with the string representation, we assume that a final H cannot shift). Thus, the bounded shift pattern in Rimi, which was ISL given a string representation, is also A-ISL.

5.3 Unbounded Tone Shift is A-ISL

Zigula unbounded tone shift is also A-ISL. Again, in this pattern a H tone shifts to the penultimate position in the word. This map is shown in Figure 6, which gives an example for the mapping from Example (3d).

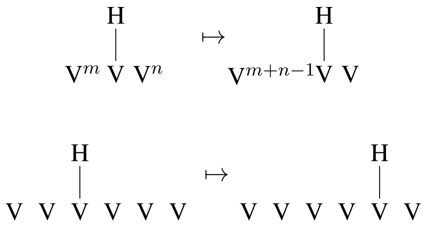


Figure 6: ARs for unbounded shift in Zigula.

This is A-ISL, as it simply requires identifying the penultimate vowel and associating that with a H tone. First, we define a predicate $\texttt{penult}(x)$, which is true for the second-to-last item in a string:

(8) $$\texttt{penult}(x) \overset{\text{def}}{=} \neg\texttt{last}(x) \wedge \texttt{last}(s(x))$$

The relevant formula for this map is again $A'(x, y)$; we omit the other formulas. We define this to be when y is the penultimate vowel:[3]

(9) $$A'(x, y) \overset{\text{def}}{=} \text{H}(x) \wedge \texttt{penult}(y)$$

Unbounded shift provides an example of a map that is A-ISL but not ISL. We thus have one

[3]In general, it is the last H tone that shifts to the end of the word (Kenstowicz and Kisseberth, 1990). To accommodate multiple H tones, this condition can be added to the definition in Equation (9) with the $\texttt{last}(x)$ predicate.

example where a pattern that is “non-local” in terms of strings is “local” when viewed in terms of ARs.

6 Analyses

In this section we apply our definitions of ISL and A-ISL to analyze a range of tone phenomena, including examples that (1) are ISL and A-ISL, (2) A-ISL but not ISL, (3) ISL but not A-ISL, and (4) neither ISL nor A-ISL.

As before, to analyze the processes in isolation, in some cases we assume restrictions on the underlying forms (i.e., the domain of the maps). We note in these cases any such assumptions.

6.1 Bounded Tone Spread

In bounded tone spread a tone on one TBU associates to some fixed number of TBUs. In Northern Bemba (Bickmore and Kula, 2013, henceforth Bemba), an underlying H tone spreads to the next vowel, but does not spread any further.[4]

(10) Bemba (Bickmore and Kula, 2013)
- a. /bá-ka-fik-a/ ↦ [bá-ká-fik-a] ‘they will arrive’
- b. /bá-ka-bil-a/ ↦ [bá-ká-bil-a] ‘they will sew’

In terms of strings, this map is as follows ($n > 0$):

$$f(w) \overset{\text{def}}{=} \begin{cases} w & \text{if } w = \varnothing^m\text{H} \\ \varnothing^m\text{HH}\varnothing^{n-1} & \text{if } w = \varnothing^m\text{H}\varnothing^n \end{cases}$$

This map is ISL, as witnessed by the transduction defined by the following formulas.

(11)
$$\text{H}'(x) \overset{\text{def}}{=} \text{H}(x) \vee \text{H}(p(x))$$
$$\varnothing'(x) \overset{\text{def}}{=} \varnothing(x) \wedge \neg\text{H}(p(x))$$

The first of these formulas states that a position is H in the output if either it was H in the input or its predecessor was H in the input. The second states the opposite: A position is $\varnothing$ in the output if it was a $\varnothing$ in the input and its predecessor was not H.

This map is also A-ISL. In terms of an autosegmental map, this process can be represented as a H tone associating to the TBU following the TBU to which it is associated in the input:

We can focus on the definition of the output association relation, as nothing else changes.

(12) $$A'(x, y) \overset{\text{def}}{=} A(x, y) \vee A(p(x), y)$$

Informally, (12) says that positions x and y are associated in the output if either (1) they are associated in the input, or (2) the predecessor of x is associated to y in the input.This allows a H to spread to exactly one following TBU (if it exists). Thus, bounded spread is both ISL and A-ISL.

6.2 Unbounded Deletion

We now turn to an example that is A-ISL but not ISL. Recall that in Arusa (Odden, 1994; Levergood, 1987) a phrase-final H is deleted following (any number of syllables after) another H:

(13) Arusa (Odden, 1994)
- a. /enkér kití/ ↦ [enkér kiti] ‘small ewe’
- b. /olórika sídáy/ ↦ [olórika siday] ‘good chair’

In this map, a stretch of Hs are converted to $\varnothing$ following some distinct stretch of H symbols—distinct here meaning that they are separated by some stretch of $\varnothing$s. However, a stretch of Hs is *not* converted if it does *not* follow some other distinct stretch of Hs.

(14)
- a. HH ↦ HH
- b. $\varnothing$H ↦ $\varnothing$H
- c. $\varnothing$H$\varnothing$H ↦ $\varnothing$H$\varnothing\varnothing$
- d. $\varnothing$H$\varnothing\varnothing$HH ↦ $\varnothing$H$\varnothing\varnothing\varnothing\varnothing$

The map can be generalized as

$$f(w) \overset{\text{def}}{=} \begin{cases} w & \text{if } w = \varnothing^\ell\text{H}^m\varnothing^n \\ v\varnothing^\ell\varnothing^m\varnothing^n & \text{if } w = v\varnothing^\ell\text{H}^m\varnothing^n, \\ & \quad v \in \Sigma^*\text{H}\Sigma^*, \ell > 0 \end{cases}$$

Lemma 2 *The Arusa map is not ISL.*

Proof: Let f be the Arusa map. For any k, consider the two strings H$\varnothing^{k-1}$ and $\varnothing^{k-1}$. These strings have the same $(k-1)$-suffix, but their

[4] This is also referred to as “binary tone spread” or “tone doubling” (Bickmore and Kula, 2013). Copperbelt Bemba has “ternary spread,” which is also considered bounded. Bounded spread in Bemba is blocked by the OCP, but we abstract away from that here (though including this constraint would not change the fact that the pattern is ISL and A-ISL).

tails clearly differ: $(\mathrm{H}, \varnothing) \in \texttt{tails}_f(\mathrm{H}\varnothing^{k-1})$, whereas $(\mathrm{H}, \mathrm{H}) \in \texttt{tails}_f(\varnothing^{k-1})$. $\square$

As we saw in Figure 1, given ARs the Arusa map deletes a final H that follows another H. The full map is thus

$$f(w) \stackrel{\text{def}}{=} \begin{cases} v \cdot \mathrm{V}^\ell\mathrm{V}^m\mathrm{V}^n \text{ if } w = v \cdot \begin{array}{c}\mathrm{H}^m\\ | \\ \mathrm{V}^\ell\mathrm{V}^m\ \mathrm{V}^n\end{array}, \\ \quad v \in \Gamma^* \cdot \begin{array}{c}\mathrm{H}\\ |\diagdown \\ \mathrm{V}\mathrm{V}^k\end{array} \cdot \Gamma^*, \ell > 0; \end{cases}$$

$f(w) \stackrel{\text{def}}{=} w$ otherwise.

As defined in §3.3, an output copy of an input position x is present if x satisfies a unary predicate $\varphi_D(x)$ establishing the domain of the output. Deletion is the failure of an input position to satisfy this formula. For Arusa, then, $\varphi_D(x)$ is true only for vowels and all but the last H.

(15) $\varphi_D(x) \stackrel{\text{def}}{=} \mathrm{V}(x) \vee \big(\mathrm{H}(x) \wedge (\texttt{first}(x) \vee \neg\texttt{last}(x))\big)$

Any final H node in the AR that is not first will fail $\varphi_D(x)$ and delete, regardless of the number of vowels between it and the preceding H. Thus unbounded deletion is A-ISL but not ISL.

6.3 Bounded Meussen's Rule

Meussen's rule is an alternation in which a H tone adjacent to a H tone either deletes or lowers to L (the deletion in Arusa in the previous section can be considered a form of Meussen's). An example from Luganda is shown in (16). Luganda distinguishes between H vowels ([á]), L vowels ([à]), and unspecified vowels ([a]).[5]

(16) Luganda (Hyman and Katamba, 2010)

a. /bálába/ → [bálàba] 'they see'

b. /bálílába/ → [bálìlàba], 'they will see'

c. /abátálíllábilila/ → [abátàlìlàbilila] 'they who will not look after'

d. /bákilába/ → [bákilába] 'they see it'

As seen clearly in Example (16), a series of H vowels of length n will surface as a HL^{n-1} sequence. We call this *bounded* Meussen's rule because it only applies to H tones that appear on adjacent vowels. Thus, Example (16d) surfaces as [bákilába], not *[bákilàba], because the two H tones are separated by one vowel. The map is thus as below $(n > 0)$.

$$f(w) \stackrel{\text{def}}{=} \begin{cases} w & \text{if } w = \{\varnothing, \mathrm{L}\}^m \\ f(v)\mathrm{HL}^{n-1} & \text{if } w = v\mathrm{H}^n, v \notin \Sigma^*\mathrm{H} \end{cases}$$

Although this applies to a sequence of Hs of any length, it is in fact ISL, because each H vowel only needs to "look" one vowel to its left in the input. Thus, the map can be defined with the following QF formulas.

(17)
$$\mathrm{H}'(x) \stackrel{\text{def}}{=} \mathrm{H}(x) \wedge \neg\mathrm{H}(p(x))$$
$$\mathrm{L}'(x) \stackrel{\text{def}}{=} \mathrm{H}(x) \wedge \mathrm{H}(p(x))$$
$$\varnothing'(x) \stackrel{\text{def}}{=} \varnothing(x)$$

Bounded Meussen's is therefore ISL, but interestingly, it is *not* A-ISL, at least given the representations of Hyman and Katamba (2010). If each vowel in a sequence of H tones is associated to its own H tone (as opposed to a single, spreading H tone), the AR map for Luganda is as follows.

$$f(w) \stackrel{\text{def}}{=} \begin{cases} f(v) \cdot \begin{array}{cc}\mathrm{H} & \mathrm{L}^{n-1}\\ | & | \\ \mathrm{V} & \mathrm{V}^{n-1}\end{array} \quad \text{if } w = v \cdot \begin{array}{c}\mathrm{H}^n\\ | \\ \mathrm{V}^n\end{array}, \\ \qquad\qquad v \notin \Gamma^* \cdot \begin{array}{c}\mathrm{H}\\ | \\ \mathrm{V}\end{array}; \end{cases}$$

$f(w) \stackrel{\text{def}}{=} w$ otherwise.

Recall that there is a distinction between a sequence of H tones that are associated to adjacent TBUs and H tones that are not associated to adjacent TBUs—the latter do not change to L.

(18)

a. $\begin{array}{cccc}\mathrm{H} & \mathrm{H} & & \\ | & | & & \\ \mathrm{V} & \mathrm{V} & \mathrm{V} & \mathrm{V}\end{array} \mapsto \begin{array}{cccc}\mathrm{H} & \mathrm{L} & & \\ | & | & & \\ \mathrm{V} & \mathrm{V} & \mathrm{V} & \mathrm{V}\end{array}$

b. $\begin{array}{cccc}\mathrm{H} & & \mathrm{H} & \\ | & & | & \\ \mathrm{V} & \mathrm{V} & \mathrm{V} & \mathrm{V}\end{array} \mapsto \begin{array}{cccc}\mathrm{H} & & \mathrm{H} & \\ | & & | & \\ \mathrm{V} & \mathrm{V} & \mathrm{V} & \mathrm{V}\end{array}$

c. ✗ $\begin{array}{cccc}\mathrm{H} & & \mathrm{H} & \\ | & & | & \\ \mathrm{V} & \mathrm{V} & \mathrm{V} & \mathrm{V}\end{array} \mapsto \begin{array}{cccc}\mathrm{H} & & \mathrm{L} & \\ | & & | & \\ \mathrm{V} & \mathrm{V} & \mathrm{V} & \mathrm{V}\end{array}$

The map is thus not A-ISL.

[5] Following Hyman and Katamba (2010), these output forms are intermediate and subject to other processes.

Lemma 3 *The Bounded Meussen's rule map is not A-ISL.*

Proof: As is clear from the above example, the string map on the tonal tier is not a function: HH may be mapped to either HH or HL. Thus, it is not ISL, and so the map is not A-ISL by Theorem 1. □

Intuitively, this distinction cannot be made only by referring to the predecessor function on the tonal tier: in each of (18), (18b), and (18c), the first H is the predecessor of the second. Thus, we cannot use predicates as in (17). Instead, the distinction is based on associations to TBUs. In order to refer to these, we must refer to the associations to TBUs. However, as A is a relation, this requires introducing new variables, which is impossible to do without introducing quantifiers.

6.4 Alternating Meussen's Rule

We now turn to a case in Shona, in which repeated application of Meussen's rule leads to surface strings of alternating H and L tones.

(19) Shona (Odden, 1986)
- a. /né-hóvé/ ↦ [né-hòvè] 'with-fish'
- b. /né-é-hóvé/ ↦ [né-è-hóvé] 'with-of-fish'
- c. /né-é-é-hóvé/ ↦ [né-è-é-hòvè] 'like-with-of-fish'

The examples in (19) present an interesting challenge in how to view this pattern in terms of strings. Under the assumption that each morpheme gets a single tone, an input like HHH is actually ambiguous without the morpheme boundaries. If HHH corresponds to H+HH, as in Example (19a), then the output is LHH. But if HHH corresponds to HH+H the output is HHL. So without morpheme boundaries the map isn't even a function and therefore is trivially not ISL.

With morpheme boundaries we see the alternating H/L pattern across the boundary. The tone sequences for the examples in (19) are as follows:

(20)
- a. H+HH ↦ H+LL
- b. H+H+HH ↦ H+L+HH
- c. H+H+H+HH ↦ H+L+H+LL

Even with the boundaries represented, however, we can show that the map is still not ISL. To simplify, in lieu of boundaries we will just assume one tone per morpheme, as follows:

(21)
- a. HH ↦ HL
- b. HHH ↦ HLH
- c. HHHH ↦ HLHL

In contrast to the bounded Meussen's rule case we saw in Luganda in §6.3, in which *any* H following another H in the input surfaces as L, here an H only surfaces as L if its predecessor was an H in the input that *also* surfaces as L *in the output*. Thus, whereas in Luganda a HHHH sequence surfaces as HLLL, in Shona it surfaces as HLHL.

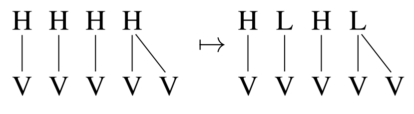
Abstracting away from unspecified TBUs, this function is as follows.

$$f(w) \stackrel{\text{def}}{=} \begin{cases} f(v)\text{L} & \text{if } w = v\text{L} \\ f(v)(\text{HL})^n & \text{if } w = v(\text{HH})^n, v \notin \Sigma^*\text{H} \\ f(v)(\text{HL})^n\text{H} & \text{if } w = v(\text{HH})^n\text{H}, \\ & \qquad v \notin \Sigma^*\text{H} \end{cases}$$

Lemma 4 *The Shona map is not ISL.*

Proof: Let f be the Shona map. For any even k, $(\text{H},\text{L}) \in \texttt{tails}_f(\text{HH}^{k-1})$, but $(\text{H},\text{H}) \in \texttt{tails}_f(\text{H}^{k-1})$, even though the two strings share the $(k-1)$-suffix H^{k-1}. Thus, f is not ISL for any even k. The case in which k is odd is similar. □

Intuitively, an H surfaces as L if it is an even-numbered position in a string of Hs. Determining odd and even positions in a string is not even FO-definable (McNaughton and Papert, 1971). Adopting ARs does not change this fact, as we still need to determine even-numbered H tones on the tonal tier, as in the following example.

(22)


Lemma 5 *The Shona map is not A-ISL.*

Proof: The map on the tonal tier is identical to the above string map, which is not ISL. By Theorem 1, then, the AR map is not A-ISL. □

This is thus an important illustration of the core concept of this paper: From an intuitive sense of ''local'' one might assume that this pattern is A-ISL, but given our rigorous definition it is not.

Pattern	Language	ISL	A-ISL
Bounded shift (§4.1, 5.2)	Rimi	✓	✓
Bounded spread (§6.1)	Bemba	✓	✓
Bounded Meussen's Rule (§6.3)	Luganda	✓	✗
Unbounded shift (§4.2,5.3)	Zigula	✗	✓
Unbounded deletion (§6.2)	Arusa	✗	✓
Alternating Meussen's Rule (§6.4)	Shona	✗	✗
Unbounded spread (§6.5)	Ndebele	✗	✗

Table 1: Summary of analyses.

6.5 Unbounded Tone Spread

Next, we examine unbounded tone spread, in which a tone spreads any number of TBUs until reaching a particular position in the word. For example, in Ndebele (Sibanda, 2004; Hyman, 2011), a high tone spreads up to the antepenult.

(23) Ndebele (Sibanda, 2004; Hyman, 2011)
- a. /ú-ku-hlek-a/ ↦ [ú-kú-hlek-a] 'to laugh'
- b. /ú-ku-hlek-is-a/ ↦ [ú-kú-hlék-is-a] 'to amuse (make laugh)'
- c. /ú-ku-hlek-is-an-a/ ↦ [ú-kú-hlék-ís-an-a] 'to amuse each other'

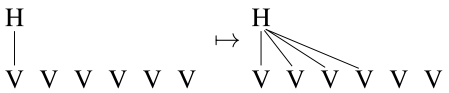
Assuming inputs with at most a single H, this function changes any number of ∅s (minus the final two) to H following an input H.

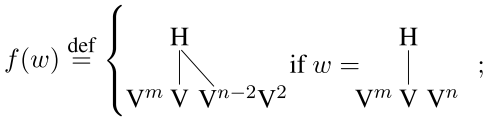
$$f(w) \stackrel{\text{def}}{=} \begin{cases} \varnothing^m \mathrm{H}\mathrm{H}^{n-2}\varnothing^2 & \text{if } w = \varnothing^m \mathrm{H} \varnothing^n, n > 1; \\ w & \text{otherwise.} \end{cases}$$

Lemma 6 *The Ndebele map is not ISL.*

Proof: Let f be the Ndebele map. For any k, the strings $\mathrm{H}\varnothing^{k-1}$ and $\varnothing^{k-1}$ have the same $(k-1)$-suffix, but $(\mathrm{H}, \varnothing) \in \mathtt{tails}_f(\mathrm{H}\varnothing^{k-1})$, whereas $(\mathrm{H}, \mathrm{H}) \in \mathtt{tails}_f(\varnothing^{k-1})$. □

Interestingly, the use of ARs instead of strings makes no difference. In an AR map, we have a single underlying H tone associating to all TBUs to its right, with the exception of the final two TBUs. The following is an AR for Example (23c), with the general map below.





$f(w) \stackrel{\text{def}}{=} w$ otherwise.

Lemma 7 *The Ndebele map is not A-ISL.*

Proof: In the case that w is an AR with m Vs, followed by a single V associated to a H, followed then by $n > 1$ Vs, the map must associate the first $n-2$ of this latter set of Vs to the H. This requires determining the set of Vs that are preceded ($\leq$) by the originally associated V. As it is well known that precedence cannot be defined in FO logic from successor (see, e.g., Libkin 2004), this predicate cannot be QF-definable. □

Interestingly, the Ndebele map is not A-ISL not because its composite tier maps are not ISL—no changes are made to the strings on either tier. Rather, it is not A-ISL strictly due to the output association relation not being QF-definable.

6.6 Summary

This section has presented analyses of a range of tone patterns, including ones that are ISL and A-ISL, patterns that are ISL but not A-ISL, patterns that are A-ISL but not ISL, and patterns that are neither. These results are summarized in Table 1.

7 Discussion

The goal of this paper has been to apply a rigorous, independently motivated notion of locality to investigate the notion that ARs "make non-local patterns local." The fact that in Table 1 common tone patterns fill out the logical possibilities of ISL and A-ISL shows that this statement is not automatically true. In fact, we found some patterns that look intuitively 'local' with ARs but, under a rigorous definition, are not. This opens up a rich line of investigation into what definitions of locality allow ARs to "make non-local patterns local," and under what conditions these definitions hold.

For example, we have taken the approach of analyzing tone processes in isolation, but it is also worth considering what happens to complexity when multiple processes are combined and interact in some way. Mathematically, this involves investigating whether or not the A-ISL class is closed under composition.

Another important question is what notion of locality might include the non-A-ISL patterns in Shona and Ndebele in §§6.4 and 6.5. One potential solution is the use of *implicit definitions* (Rogers, 1997). Briefly, implicit definitions allow the recursive definition of a predicate. For example, for unbounded spreading, as in Ndebele, we could define the output association relation as follows.

$$A'(x, y) \stackrel{\text{def}}{=} A(x, y) \vee A'(p(x), y)$$

This is a recursive definition in that any pair x, y that are associated is the base case ($A(x, y)$), and in the recursive case the predecessor of x is associated in the output to y ($A'(p(x), y)$). This predicate thus iteratively adds associations following an initial input association, much in the way that phonologists have characterized unbounded spread as the iterative application of local spread. A full study of such definitions, and whether or not they are decidable for ARs, is left for future work.

Finally, a full exploration of the A-ISL class should situate it in the sub-regular (i.e., describable with finite-state transducers) hierarchy of computational complexity as has been done in previous work on phonological complexity (e.g., Gainor et al., 2012; Heinz and Lai, 2013; Jardine, 2016, among others). We conjecture that A-ISL falls between the regular and ISL functions.

As some tone patterns were neither ISL or A-ISL, this raises the question of whether or not QF transductions are appropriate as a computational notion of locality. Clearly, on their own, QF transductions with a successor do not capture all phonological patterns, even over ARs. However, they do implement a fundamental notion of locality, which is that computations can only occur within some fixed window. We thus believe that a final computational theory of phonological processes will somehow incorporate QF definitions—one example is the implicit definitions mentioned above.

8 Conclusion

We have introduced the A-ISL class of maps, which are defined through QF transductions over ARs, and compared them with ISL string functions with respect to common tone processes. Our results point to a further division among ''unbounded'' or ''non-local'' phenomena into those that are A-ISL and those that cannot be modeled locally even with the use of ARs. This adds to our understanding of the computational nature of tonal processes and how they are represented.

References

Lee S. Bickmore and Nancy C. Kula. 2013. Ternary spreading and the OCP in Copperbelt Bemba. *Studies in African Linguistics*, 42.

Steven Bird and T. Mark Ellison. 1994. One-level phonology: Autosegmental representations and rules as finite automata. *Computational Linguistics*, 20.

Jane Chandlee. 2014. Strictly Local Phonological Processes. Ph.D. thesis, University of Delaware.

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

Jane Chandlee and Jeffrey Heinz. 2018. Strictly locality and phonological maps. *LI*, 49:23–60.

Jane Chandlee and Steven Lindell. in prep. A logical characterization of strictly local functions. In Jeffrey Heinz, editor, *Doing Computational Phonology*. OUP.

G. N. Clements. 1976. *Vowel Harmony in Nonlinear Generative Phonology: An Autosegmental Model*. Bloomington: Indiana University Linguistics Club Publications.

Bruno Courcelle. 1994. Monadic second-order definable graph transductions: a survey. *Theoretical Computer Science*, 126:53–75.

Joost Engelfriet and Hendrik Jan Hoogeboom. 2001. MSO definable string transductions and two-way finite-state transducers. *ACM Transations on Computational Logic*, 2:216–254.

Emmanuel Filiot and Pierre-Alain Reynier. 2016. Transducers, logic, and algebra for functions of finite words. *ACM SIGLOG News*, 3(3):4–19.

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

John Goldsmith. 1976. Autosegmental Phonology. Ph.D. thesis, Massachussets Institute of Technology.

Jeffrey Heinz and Regine Lai. 2013. Vowel harmony and subsequentiality. In *Proceedings of the 13th Meeting on Mathematics of Language*. Sofia, Bulgaria.

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

Larry Hyman and Francis X. Katamba. 2010. Tone, syntax and prosodic domains in Luganda. In Laura Downing, Annie Rialland, Jean-Marc Beltzung, Sophie Manus, Cédric Patin, and Kristina Riedel, editors, *Papers from the Workshop on Bantu Relative Clauses*, volume 53 of ZAS Papers in Linguistics, pages 69–98. ZAS Berlin.

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

Adam Jardine. 2017. On the logical complexity of autosegmental representations. In *Proceedings of the 15th Meeting on the Mathematics of Language*, pages 22–35. London, UK. Association for Computational Linguistics.

Martin Kay. 1987. Nonconcatenative finite-state morphology. In *Proceedings, Third Meeting of the European Chapter of the Association for Computational Linguistics*, pages 2–10.

Michael Kenstowicz and Charles Kisseberth. 1990. Chizigula tonology: the word and beyond. In Sharon Inkelas and Draga Zec, editors, *The Phonology–Syntax Connection*, pages 163–194. Chicago: the University of Chicago Press.

András Kornai. 1995. *Formal Phonology*. Garland Publication.

Barbara Levergood. 1987. Topics in Arusa phonology and morphology. Ph.D. thesis, University of Texas.

Leonid Libkin. 2004. *Elements of Finite Model Theory*. Berlin: Springer-Verlag.

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

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

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

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

James Rogers. 1997. Strict lt2 : Regular :: Local : Recognizable. In Christian Retoré, editor, *Logical Aspects of Computational Linguistics: First International Conference, LACL '96 Nancy, France, September 23–25, 1996 Selected Papers*, pages 366–385. Springer Berlin Heidelberg, Berlin, Heidelberg.

Thilo Schadeberg. 1979. Über die töne der verbalen formen im rimi. *Afrika und Übersee*, 57:288–313.

Galen Sibanda. 2004. Verbal phonology and morphology of Ndebele. Ph.D. thesis, UC Berkeley.

Bruce Wiebe. 1992. Modelling autosegmental phonology with multi-tape finite state transducers. Master's thesis, Simon Fraser University.

Moira Yip. 2002. *Tone*, Cambridge University Press.