Autosegmental ISL functions

First Author
Affiliation / Address line 1
Affiliation / Address line 2
Affiliation / Address line 3
email@domain

Second Author
Affiliation / Address line 1
Affiliation / Address line 2
Affiliation / Address line 3
email@domain

Abstract

1 Introduction

It has long been claimed that autosegmental representations (ARs) can capture non-local processes in a local way (Odden, 1994). Focusing on the domain of tone patterns, we explore this claim from a computational perspective by applying a precisely defined notion of locality following (Chandlee, 2014; Chandlee and Lindell, in prep). Our exploration reveals a three-way division among tone patterns: those that can be modeled locally without ARs (i.e., with just strings), those that can be modeled locally with ARs, and those that can't be modeled locally at all. We explicitly define the local processes (both with and without ARs) and offer a potential path for defining the non-local ones. We argue that the result is a more thorough understanding of how ARs interact with phonological locality.

The formalism we use to precisely define what it means to be local is first order (FO) logic. As we will show, the use of logical characterizations of phonological patterns is motivated by the straightforward means of extending the previous work in this line of research from strings to ARs.

The remainder of this paper is organized as follows. In §2 we briefly review autosegmental representations, and in §3 we briefly review the computational property of 'input strict locality' (ISL) that will serve as the distinction between local and non-local in the analyses to follow. §4 sketches Chandlee and Lindell (in prep)'s logical characterization of ISL functions through an illustrative example of a tone pattern that can be analyzed locally using strings. §5 extends that characterization to maps whose input is an autosegmental graph, as defined by Jardine (2016b), and demonstrates the range of tone processes that can be classified as ISL using ARs. §6 reviews a selection of tone processes that cannot be described in this way, highlighting the distinction between 'local' and 'non-local' maps in the computational sense advocated in this paper. §7 discusses the significance of these results and notes a few important directions for future research. §8 concludes.

2 Autosegmental representations

Autosegmental representations (Goldsmith, 1976; Clements, 1976) are phonological representations in which phonological primitives are arranged into distinct strings or *tiers*, with an *association* relation relating units on different tiers. ARs are often used to model tone patterns; an example word from Arusa (Odden, 1994; Levergood, 1987) is represented in Figure 1 both as a string and an AR. Here and throughout the paper, an acute accent [á] indicates a high (H)-toned vowel; unmarked vowels are pronounced with low (L) tones.

Figure 1 shows that the Arusa word [sídáy] 'good', in which both vowels are pronounced with a H tone, can be represented with an AR with two vowels on an vowel tier and a single H tone on a separate tonal tier that is associated to both vowels (indicated with straight lines). While tones are often considered to associate to moras or syllables

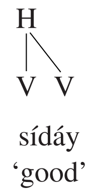

Figure 1: AR example from Arusa

(Yip, 2002), we use vowels here to stand in for any tone-bearing unit.

ARs have been argued to provide natural accounts of many tone and segmental processes. In particular, they have been claimed to model the long-distance interactions that are common in tone. To illustrate, when preceded by the word [olórika] 'chair' in the phrase [olórika siday] 'good chair', [síďáy] 'good' is pronounced instead with two low tones. This is explained through a *mapping* in which the underlying H tone depicted in Figure 1 is deleted following another H tone in the AR. This is shown schematically in Figure 2.

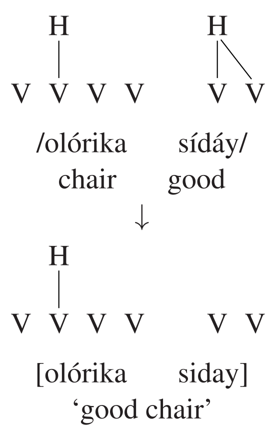

Figure 2: AR mapping from Arusa

In the underlying form (top graph), the vowels of /síďáy/ 'good' are associated to a single H tone (as in Figure 1), but this H tone is deleted following the H of /olórika/ 'chair'. As Odden (1994) points out, in terms of the string, this is a 'long-distance' process: the tone specification of the second vowel in /olórika/ 'chair' affects the tone specification of the vowels in /síďáy/, which are two vowels away. However, viewed as an AR, this becomes a 'local' process, in which H tones that are adjacent on the tonal tier interact.

The computational properties of ARs have been studied before (Kay, 1987; Wiebe, 1992; Bird and Ellison, 1994; Kornai, 1995; Jardine, 2017). In particular, (Jardine, to appear) shows that local grammars over sets of ARs can capture the range of long-distance well-formedness conditions in tone. However, tone *mappings*, such as the one exemplified in Figure 2, have not yet been studied under a computational notion of locality. This paper defines such a notion by extending input strict locality (ISL)—a previously defined locality property for string mappings—to autosegmental mappings. The next section presents the ISL property defined in previous work.

3 Input strict locality

The ISL property comes from the ISL functions, which are a proper subset of the regular relations (Chandlee, 2014; Chandlee et al., 2014). These functions are characterized as ones that determine an output string for a given input string based only on contiguous substrings of bounded length. As a subset of regular relations, they have a reduced computational complexity and expressivity. Despite their restrictive nature, however, they have been shown to be sufficient to model a significance range of local segmental phonological maps (Chandlee, 2014; Chandlee et al., 2014). The property of ISL thus serves as a precise, computationally defined notion of phonological locality. Here we provide a brief, informal explanation of what it means to be ISL; readers are referred to the work cited above for the technical details and for formal language-theoretic and automata-theoretic characterizations.

The defining trait of an ISL map is essentially this. Given an input string, at any given point the determination of what each input segment contributes to the output string can be made based only on that segment and a bounded number of its surrounding segments. This is illustrated in the example place assimilation map shown in Figure 3. Given an input like /ɪnpɹabʌbl̩/, the function needs to determine the corresponding output string. The 'input strictly local' designation means the function can perform this task by only focusing on a bounded substring in the input at any given time during the computation. Thus, as highlighted in Figure 3, when it reads the

/np/ sequence in the input, it can append [mp] to the output.

Input:	ɪ	**n**	**p**	ɹ	a	b	ʌ	b	l̩
Output:	ɪ		*mp*	ɹ	a	b	ʌ	b	l̩

Figure 3: ISL nasal place assimilation map for the input 'improbable'

It doesn't matter what came before this /np/ and it doesn't matter what might come after it: the output at this stage must always include [mp]. ISL functions are parameterized by the length of the substring it needs to look for; in this example the length of that substring (np) is 2, so the function is a 2-ISL function.

The idea that local phonology pays attention to contiguous substrings is certainly not novel, but ISL provides a precise notion of what it means for a phonological map to be local in a computational sense. Our goal in what follows is to see how well the notion of locality provided by ISL aligns with the sense in which autosegmental representations enable a local treatment of otherwise non-local phenomena.

In the next section we present the framework in which we will be conducting our investigation: FO logic formulae defined over graphs. We begin with graphs for string representations, extending them to ARs in §5.

4 ISL graph interpretations

Chandlee and Lindell (in prep) provide a logical characterization of ISL functions and show that they are equivalent to quantifier-free (QF) first order (FO) graph interpretations. The significance of the QF designation is that this class of functions can be modeled without the full expressivity of FO logic (i.e., they are computationally very simple). We will present this characterization through an example of a tone pattern that can be modeled without ASRs (i.e., just with strings). Readers are referred to Chandlee and Lindell (in prep) for the technical details and the proof of equivalence between this logical characterization and the ISL functions as originally defined.

We will use the example of bounded tone shift in Rimi. Bounded tone shift refers to the process by which a tone appears some fixed number of vowels away from its underlying position. In Rimi (Meyers, 1997), a tone shifts one vowel to the right. In the examples in (1), the affixes −a, −mu, and −i̧ have a low tone underlyingly but surface with a high tone when attached to a stem that ends in a high tone.

(1) Rimi (Schadeberg, 1978; 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'

A string graph representation of the underlying form in (1-a) is shown in Figure 4.

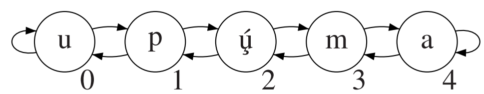

Figure 4: String graph representation of the UR in (1-a)

Figure 4 represents a string model with domain $\{0, 1, 2, 3, 4\}$. Position 0 is labeled 'u', position 1 is labeled 'p', etc. Both the successor (s) and predecessor (p) functions are represented with arrows (e.g., $s(0) = 1$, $s(3) = 4$, $p(1) = 0$, $p(4) = 3$, etc.). The start and end of the string are not represented explicitly (e.g., with symbols like #); instead, the word boundaries can be detected via the s and p functions. The first position in the word is its own predecessor, and likewise the final position in the word is its own successor. These facts are represented in the figure with looping arrows on positions 0 and 4.

Using this model of the underlying form, we define FO formulas that construct the corresponding surface form graph according to the bounded tone shift map. The formulas are all QF due to the existence of an upper bound on the substring that participates in the map. In other words, the fact that the map is ISL enables us to model it by focusing only on a subgraph of the input graph of fixed size.

We need a formula that identifies (i.e., evaluates to True for) position 4 as one that has a high tone in the output, and position 1 as one that does not have a high tone in the output. We will build up this formula in pieces. First we define a formula $H(x)$ that is True if position x has a high tone in the input:

(2) $H(x) \overset{\text{def}}{=}$ ú$(x) \vee$ á$(x) \vee$ é$(x) \vee$ į́(x) ...

This formula simply checks whether position x is labeled with any of the vowels with a high tone. Note that using strings instead of ARs means each vowel with a high tone is considered a separate label (as opposed to a single high tone label).

For simplicity let's assume only one consonant can intervene between two vowels (we will show how to relax this assumption in a moment). Then the following formula will identify a position that is labeled with 'a' and for which the preceding vowel has a high tone.

(3) $a(x) \wedge H(p(p(x)))$

This formula evaluates to True for exactly one position in the graph in Figure 4. Position 4 is labeled with 'a' and the predecessor of its predecessor—position 2—has a high tone.

Adopting the notations of Engelfriet and Hoogeboom (2001), we designate the formula in (3) as $\varphi_{á}(x)$, which has the effect of labeling the output correspondent of any x that satisfies it with 'á'. We define comparable formulas for the other vowels, a couple of which are shown in (5):

(4) a. $\varphi_{ú}(x) \overset{\text{def}}{=} u(x) \wedge H(p(p(x)))$
b. $\varphi_{é}(x) \overset{\text{def}}{=} a(x) \wedge H(p(p(x)))$

Vowels that do not follow a high tone vowel—as well as the high tone vowel itself—must bear a low tone in the output. The following formula ensures that position 2 is labeled as ų and not ų́ in the output. Again we would define comparable formulas for the other vowels.

(5) $\varphi_{ų}(x) \overset{\text{def}}{=}$ ų$(x) \vee$ ų́(x)

The output correspondents of all consonants (abbreviated with C) are labeled identically as their input correspondents, due to the formula shown in (6).

(6) $\varphi_C \overset{\text{def}}{=} C(x)$

Collectively these formulas, along with ones that state that the predecessor and successor relations are identical in the input and output, give us the output graph in Figure 5.

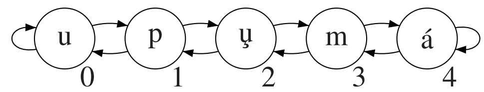

Figure 5: String graph representation of the SR in (1-a)

Through this example we can see that the potential for defining QF formulas comes from the upper bound on how many times we will need to call on p or s to find the crucial information that determines whether or not an alternation will take place. In other words, we don't need a formula to ask whether there exists a high tone two positions away; we can instead just ask a question about the predecessor of the predecessor, since we know that position coincides with the preceding vowel. If more than one consonant could intervene, we could simply expand this as need (e.g., to $p(p(p(x)))$ or $p(p(p(p(x))))$, etc.). The important thing is that due to constraints on syllable structure, an upper bound does exist.

The example of Rimi bounded tone shift is a case where we can model the map as ISL even without using ARs. Our goal in this paper is to extend this method for describing local maps using string representations to also explore maps that do require ARs. We will see that in certain cases the use of ARs enables modeling a map as ISL that would otherwise not be ISL-describable.

5 ISL autosegmental maps

In this section we will apply the characterization of ISL explained above to a variety of tone patterns to show that they are 'local' in this formal sense. First, however, we must extend the string graph representations used in the previous section to accommodate autosegmental representations.

5.1 Autosegmental graphs

Autosegmental graphs include two strings, a string of tone bearing units (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 relations, just as in the previous example. In addition, the association relation relates positions from the tone string to positions of the tone bearing unit string. We thus have the signature in (7), where D is the domain, p and s are the successor functions, and a is the association relation.

(7) $\langle D; p, s, a \rangle$

An example is shown in Figure 6, with an explicit graph model of the autosegmental representation for the underlying Arusa phrase /olórika sídáý/ 'good chair' from Figure 1 (for the sake of simplicity, word boundaries have been omitted).

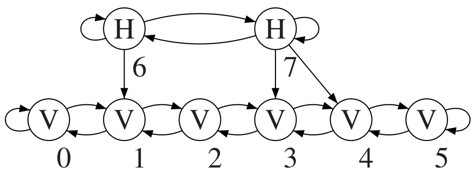

Figure 6: Example of an autosegmental graph

The representation in Figure 6 has the domain $\{0, 1, 2, 3, 4, 5, 6, 7\}$ bifurcated into two strings, HH and VVVVVV, each with its own predecessor and successor function as described in the previous section. In addition, there is an association relation $a = \{(6, 1), (7, 3), (7, 4)\}$ relating tones to their TBUs. While some authors define association as symmetric (Kornai, 1995), for the sake of the simplicity of the formulas in the QF graph interpretations, we define it as antisymmetric, although this assumption has no bearing on QF-definability.

In the following subsections we will use these AR models to show how various tone patterns can be modeled as ISL using QF FO formulas. These patterns will thereby be classified as ISL, or more specifically as ISL with ARs. We will use the designation A-ISL for such patterns. Specifically, we will apply the definition shown below.

Definition 1 *An input-output map is A-ISL if it can be modeled as a QF graph interpretation where the graph is an autosegmental graph.*

In each case we will also show whether or not the pattern can also be modeled with string graphs (i.e., whether in fact the ISL designation *depends on* the use of ARs). This will lead to the three-way classification scheme mentioned above, among patterns that are QF-describable with strings, patterns that are QF-describable with ARs, and patterns that are not QF-describable with either representation.

5.2 Bounded tone shift

The first pattern we will consider is bounded tone shift. We already showed one example of this patterns, with the string-based analysis of Rimi above. The Rimi pattern can also be modeled with ARs. Figure 7 shows an AR graph for example (1-a). Note that for ease of reading we will represent AR graphs in the remainder of the paper in this way (i.e., without the nodes, position numbers, successor relation, and predecessor relation pictured).

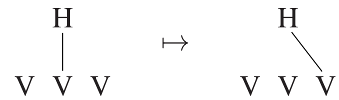

Figure 7: AR representation of bounded shift in Rimi

Note that between the input and output graphs, nothing changes with respect to position labeling or the successor and predecessor functions for either string (TBUs and tones). So the complete analysis would include formulas that maintain all of these things from input to output. We omit these formulas and instead focus on what *does* change in this map, which is the association relation.

The formula for the output re-association is given below in (8). Informally, it states that positions x and y are associated in the output graph if the predecessor of x is associated to y in the input graph.

(8)

$$a_O(x, y) \stackrel{\text{def}}{=} \quad a_I(p(x), y) \wedge H(y)$$

Another example of bounded shift is found in Kuki-Thaadow, where a string of tones each associate to the following vowel, as in (9) below.

(9) Kuki-Thaadow (Hyman, 2011)
/ka zóoŋ lien thúm/ $\mapsto$ [ka zooŋ líen thǔm],
'my three big monkeys'

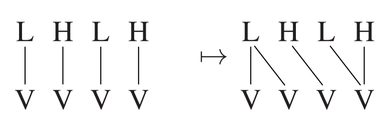

Figure 8: AR representation of bounded shift in (9)

In Kuki Thaadow, the first and last tones also reamin associated to their input vowels as well. This requires a slight variation on the above equation:

(10)

$$a_O(x,y) \overset{\text{def}}{=} \begin{array}{l} a_I(p(x),y) \vee \\ (\texttt{first}(x) \wedge a_I(x,y)) \\ (\texttt{last}(x) \wedge a_I(x,y)) \end{array}$$

Informally, this formula says that x and y should be associated in the output if the predecessor of x is associated to y in the input.

On the surface the Kuki Thaadow patterns appears more complicated than the one found in Rimi, but like the Rimi case, Kuki Thaadow tone shift could also be modeled with a string representation. This is again because all of the interactions involve adjacent vowels that will be separated by a string of consonants with an upper bound.

5.3 Unbounded tone shift

We can compare the examples from Rimi and Kuki Thaadow with the tone shift attested in Ziqula, which we will show is A-ISL but not ISL. In this *unbounded* tone shift, tones shift a longer distance to some fixed position in a word. In Zigula (Kenstowicz and Kisseberth, 1990) we find both toneless (11-a) and toned (11-b)-(d) examples.

(11) 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 each other'

In the toned examples, the underlying tone appears not on the root itself but on the penultimate vowel. This mapping can be represented autosegmentally as in Figure 9.

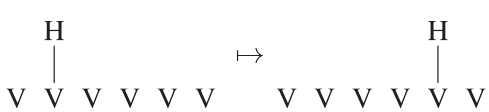

Figure 9: AR representation of unbounded tone shift in Zigula

This is A-ISL, as it simply requires identifying the penultimate vowel and associating that with a H tone. For convenience, we can define a predicate $\texttt{penult}(x)$ which is true for the penultimate vowel:

(12)

$$\texttt{penult}(x) \overset{\text{def}}{=} V(x) \wedge (s(s(x)) = s(s(s(x)))$$

Recall that for an x that is the last member of the string, $s(x) = x$. So for the second-to-last member of the string $s(s(x)) = s(x)$.

The relevant formula for this mapping is thus $a_O(x,y)$, which determines when the output copies of x and y are associated in the output. We thus define this to be when x is the last H tone and y is the penultimate vowel:[1]

(13)

$$a_O(x,y) \overset{\text{def}}{=} H(x) \wedge s(x) = x \wedge \texttt{penult}(y)$$

Thus, unbounded shift is A-ISL: associating the final H tone and the penultimate vowel can be accomplished with a QF formula. However, the corresponding string mapping is not ISL. To see this, consider the distinction between the mapping for toneless verb roots (a) and toned verb roots (b) shown in Figure 10.

a. V V V V V V ↦ V V V V V V
b. V V́ V V V V ↦ V V V V V́ V

Figure 10: String representation of unbounded tone shift in Zigula

The contrasting string mappings in Figure 10 highlight what is important about the mapping: a penultimate V in the underlying form *may* be changed to V́ in the output, but if and only if a V́ precedes in the underlying form. Like in Arusa, this cannot be checked with a substring of a fixed size, since the preceding V́ in the underlying form may be any number of positions to the right. Thus, unbounded shift provides an example of a mapping that is A-ISL but is not ISL. In other words, the potential to classify this map as local depends on allowing ARs.

[1]We specify that it is the last H tone; what happens to previous H tones is rather complex, and we abstract away from this here. Interested readers are referred to (Kenstowicz and Kisseberth, 1990).

5.4 Bounded tone spread

In bounded tone spread a tone on one vowel also associates to the next vowel, but does not spread any further. An example from Bemba is shown in (14).

(14) Bemba (Bickmore and Kula, 2013)
   a. /bá-la-kak-a/ $\mapsto$ [bá-lá-kak-a] ‘they tie up’
   b. /tu-la-bá-kak-a/ $\mapsto$ [bá-lá-kak-a] ‘we tie them up’

This map can be represented as in Figure 11.

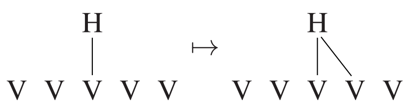

Figure 11: AR representation of bounded spread in (14-b)

Once again, we focus on defining the output association relation, since everything else remains the same. The output association relation can be defined as follows:

(15)

$$a_O(x,y) \stackrel{\text{def}}{=} \quad a_I(x,y) \vee a_I(p(x),y)$$

Informally, this says that positions x and x are associated in the output graph if either 1) they are associated in the input graph, or 2) the predecessor of x is associated to y in the input graph. In Figure 11, the first condition is met when x=2 and y=5 and the second condition is met with x=3 and y=5. Thus in the output graph, a includes (2,5) and (3,5).

Bounded shift is also ISL when viewed as a string mapping. We can consider strings of low-toned vowels (V) and high-toned vowels (V́), as in Figure 12.

V V́ V V V $\mapsto$ V V́ V́ V V

Figure 12: String representation of bounded spread in (14-b)

That this mapping is ISL is witnessed by the following formula, which states that a vowel is high toned if either it was high-toned in the input or its predecessor was high-toned in the input.

$$\acute{\mathrm{V}}_O(x) \stackrel{\text{def}}{=} \acute{\mathrm{V}}_I(x) \vee \acute{\mathrm{V}}_I(p(x))$$

Bounded spread is thus both ISL and A-ISL.

5.5 Bounded Meussen’s rule

Meussen’s rule refers to a type of tone alternation in which a tone adjacent to a high tone lowers. An example from Luganda is shown in (16) and Figure 13:

(16) Luganda (Hyman and Katamba, 1993; Hyman and Katamba, 2010)
   /bálílába/ $\rightarrow$ [bálìlàba], ‘they will see’

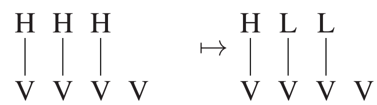

Figure 13: AR representation of Meussen’s Rule in Luganda

This time the key part of the interpretation is the formulas that determine the output position labels, rather than the association relation. As can be seen in Figure 13, the association lines do not change from the input graph to the output graph. The formulas that define which output positions are labeled H and which are labeled L are given in (17) and (18), respectively.

(17) $H_O(x) \stackrel{\text{def}}{=} H(x) \wedge \neg H(p(x))$

(18) $L_O(x) \stackrel{\text{def}}{=} L(x) \vee (H(x) \wedge H(p(x)))$

Informally, (17) says that an output position is labeled H if its input correspondent is labeled H and the predecessor of its input position is NOT labeled H. Likewise, (18) says an output position is labeled L if its input correspondent is labeled L, or else if its input correspondent is labeled H and the predecessor of its input correspondent is also labeled H.

This pattern is also ISL when viewed as a string mapping, as shown in Figure 14. Here, V́ indicates a high-toned vowel, V̀ a low-toned vowel, and V an unspecified vowel (as in (16)).

This string mapping is ISL because it can be defined by formula identical to those in (17) and (18) but replacing H with V́ and L with V̀.

$$\acute{\text{V}}\ \acute{\text{V}}\ \acute{\text{V}}\ \text{V}\ \text{V} \mapsto \acute{\text{V}}\ \grave{\text{V}}\ \grave{\text{V}}\ \text{V}\ \text{V}$$

Figure 14: String representation of local Meussen's rule in (16)

(19) $\acute{\text{V}}_O(x) \stackrel{\text{def}}{=} \acute{\text{V}}(x) \wedge \neg\acute{\text{V}}(p(x))$

(20) $\grave{\text{V}}_O(x) \stackrel{\text{def}}{=} \grave{\text{V}}(x) \vee (\acute{\text{V}}(x) \wedge \acute{\text{V}}(p(x)))$

Applications of Meussen's Rule like in Luganda is both A-ISL and ISL. The next section presents an example of *unbounded* Meussen's Rule, which we will see merits a different classification.

5.6 Unbounded Meussen's rule

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

(21) Arusa (Odden, 1994)

a. sídáy 'good'
b. enkér siday 'good ewe'
c. olórika siday 'good chair'

As we saw in Figure 2, this is represented as an AR mapping in which a H autosegment following another is deleted. This is repeated below in Figure 15.

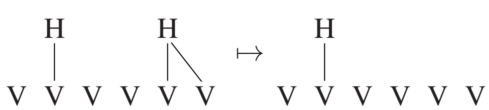

Figure 15: AR representation of long-distance Meussen's rule in Arusa

Deletion is implemented in logical interpretations through the failure of an input node to receive any label in its output (Engelfriet and Hoogeboom, 2001). Thus the following formula deletes all but the first H autosegment.

(22) $H_O(x) \stackrel{\text{def}}{=} H(x) \wedge \neg H(p(x))$

The formula in (22) is identical to that in (17) for output H nodes in Luganda. However, in the Arusa pattern, we assume that no nodes receive a L label in the output. Thus, any input H node that follows another H node receives no label, and is thus deleted. This achieves the deletion mapping in Figure 15. Thus, this mapping is A-ISL.

However, when viewed as a string mapping, it is not ISL. The equivalent string mapping is exemplified by the mapping in Figure 16.

$$\text{V}\ \acute{\text{V}}\ \text{V}\ \text{V}\ \acute{\text{V}}\ \acute{\text{V}} \mapsto \text{V}\ \acute{\text{V}}\ \text{V}\ \text{V}\ \text{V}\ \text{V}$$

Figure 16: String representation of the unbounded Meussen's rule in (16)

In this mapping, a stretch of V́ symbols are converted to V following some distinct stretch of V́ symbols—distinct here meaning that they are separated by some stretch of V symbols. However, a stretch of V́ symbols is *not* converted if it does *not* follow some other distinct stretch of V́ symbols (thus the initial V́ is not altered in Figure 16). It is easy to show that checking for a preceding stretch of V́ symbols is not ISL: the span of intervening Vs can be of any length, and so there is no finite substring one can check preceding a V́ to see whether or not there is a preceding V́.

This problem is obviated in the AR: the H to be deleted immediately follows another on the tonal tier. Thus, long-distance Meussen's rule in Arusa is another example of a mapping that is local in terms of ARs but is not local in terms of strings.

5.7 Tone split

Tone split is when a contour tone—either rising (LH) or falling (HL)—that is associated to a single TBU is split such that the first tone stays on the original TBU while the second associates to a following TBU. An example is found in Mende, with examples given in (23).

(23) Mende (Leben, 1978; Hyman, 2011)

a. /mbû-hu/ $\mapsto$ [mbú-hù] 'in owl'
b. /mbǎ-hu/ $\mapsto$ [mbà-hú] 'in rice'

Figure 17 shows the AR for tone split.

To begin, we define the formula in (24) that identifies an input position that is associated to a falling contour (HL). In other words, this is a position associated to both a H and an L and the H position is the predecessor the L position.

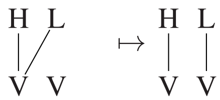

Figure 17: AR representation of Mende tone split as in (23-a)

(24) $HL(x) \overset{\text{def}}{=} a(x,y) \wedge a(x,z) \wedge H(y) \wedge L(z) \wedge p(y) = z$

Likewise, the formula in (25) identifies an input LH.

(25) $LH(x) \overset{\text{def}}{=} a(x,y) \wedge a(x,z) \wedge L(y) \wedge H(z) \wedge p(y) = z$

Now we define the output association formula. We will build this up in pieces for each scenario. First, an output position x is associated to an output position y that is labeled H if x has an HL contour in the input OR if its predecessor has an LH contour in the input.

(26) $H(y) \wedge [HL(x) \vee LH(p(x))]$

Similarly, an output position x is associated to an output position y that is labeled L if x has an LH contour in the input or its predecessor has an HL contour.

(27) $L(y) \wedge [LH(x) \vee HL(p(x)]$

The output association formula is then the disjunction of (26) and (27).

(28) $a_O(x,y) \overset{\text{def}}{=} [H(y) \wedge [HL(x) \vee LH(p(x))]] \vee [L(y) \wedge [LH(x) \vee HL(p(x)]]$

Once again, this map is also ISL with string representations, as the vowels involved are adjacent (or separated by a bounded number of consonants). Figure 18 shows the string-based representation of the tone split example in (23-a).

V̂ V ↦ V́ V̀

Figure 18: String representation of tone split in Mende example (23-a)

6 Non-local maps

The previous section presented a range of tone mappings that are either ISL and/or A-ISL. The classifications showed how the use of ARs enabled Qf modeling of unbounded phenomena that are non-ISL if only string representations are permitted. This of course does not amount to a claim that all unbounded phenomena are ISL once we have ARs. In this section we present a few more tone patterns that cannot be modeled with QF FO logic even with ARs. These patterns are then considered 'truly non-local' in the computational sense advocated for in this paper.

First is a case from Shona, in which repeated application of Meussen's Rule to underlying sequences of H tones leads to surface strings of alternating tones. In the examples in (29), the tone sequences are shown in parentheses.

(29) Shona (Odden, 1986; Meyers, 1987; Meyers, 1997)
  a. /né-hóvé/ ↦ [né-hòvè] (HH↦HL)
  b. /né-é-hóvé/ ↦ [né-è-hóvé] (HHH↦HLH)
  c. /né-é-é-hóvé/ ↦ [né-è-é-hòvè] (HHHH↦HLHL)

This pattern is not QF-definable; it's well-known that Monadic Second Order (MSO) logic is required to pick out even and odd elements (McNaughton and Papert, 1971).

Next is unbounded tone spread, such as in Ndebele (Sibanda, 2004; Hyman, 2011). An underlying high tone spreads up until the penultimate syllable.

(30) Ndebele
  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'

In our analysis of unbounded tone shift above, we used the following formula to identify the penultimate vowel of the word:

(31)

$$\texttt{penult}(x) \stackrel{\text{def}}{=} V(x) \wedge (s(s(x)) = s(x)$$

Likewise with unbounded tone spread, we can use this `penult` formula in the formula that identifies which output positions are associated to a high tone.

$$a_O(x,y) \stackrel{\text{def}}{=} \begin{array}{l} H(y) \wedge \neg\texttt{penult}(x) \wedge \\ \exists z[a_I(z,y) \wedge z \triangleleft x] \end{array} \quad (1)$$

But note crucially that this formula contains a quantifier, which is used to ensure that a H tone is associated to another position that precedes x. The need for a quantifier of course means this formula is not QF, which puts the pattern outside of our definition of A-ISL.

Lastly, Unbounded Tone Plateauing (UTP) as in the Luganda example in (32) (Hyman, 2011; Jardine, 2016a) is also not QF-definable. In UTP, a high tone spreads up to another high tone which may be an unbounded number of vowels away, creating a 'plateau' of high tones.

(32) Luganda /bikópo byaa-walúsiimbi/ $\mapsto$ [bikópóbyáá-wálúsiimbi] 'the cups of Walusimbi

The conditions under which an output vowel has a high tone are that it either 1) has a high tone in the input, or else 2) it both precedes and follows vowels that have a high tone in the input. This second condition requires a quantifier: for an output position z to hold a high tone there must *exist* two input positions x and y associated to high tones such that x precedes z and z precedes y. Furthermore, the sense of 'precedes' here must be general precedence $<$ rather than immediate predecessor $\triangleleft$. Such a formula thus falls outside the reach of the restrictive QF formulas we're using to show something is A-ISL.

This section has presented three examples of tone patterns that are neither ISL nor A-ISL, meaning they do not have the computational property of input strict locality regardless of whether autosegmental representations are employed. This list is not meant to be exhaustive. Our goal here is to show through examples what kinds of patterns are and are not ISL/A-ISL.

Pattern	**Language**	**ISL**	**A-ISL**	**¬ISL**
Bounded shift (§5.2)	Rimi	✓	✓	
Bounded shift (§5.2)	Kuki Thaadow	✓	✓	
Unbounded shift (§5.3)	Zigula	✗	✓	
Bounded spread (§5.4)	Bemba	✓	✓	
Bounded Meussen's Rule (§5.5)	Luganda	✓	✓	
Unbounded Meussen's Rule (§5.6)	Arusa	✗	✓	
Split (§5.7)	Mende	✓	✓	
(§6)	Shona	✗	✗	✓
Unbounded spread (§6)	Ndebele	✗	✗	✓
UTP (§6)	Luganda	✗	✗	✓

Table 1: Summary of analyses

In the next section we summarize the results from this and the previous section and discuss their implications.

7 Discussion

The findings of the previous two sections are summarized in Table 1.

Classifying tone patterns as bounded/unbounded is not a new observation, as the commonly used terms in the table show. Our goal for this paper is to apply a rigorous, mathematically grounded way to think about this distinction. The very fact that in the table unbounded patterns are further divided into A-ISL and non-ISL indeed suggests that something more nuanced is going on in the way that ARs enable 'local' analyses.

This initial investigation into the computational complexity of tone patterns points to several important areas for future work. First, 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. For example, in §5.7 tone split in Mende was analyzed as both ISL and A-ISL. The full paradigm for this pattern also includes tone spread in the case when the final tone is not a contour tone. This is shown in the examples in (33-a) and (33-b), with (33-c) and (33-d) repeated from (23) above.

(33) Mende (Leben, 1978; Hyman, 2011)

a.	/kɔ́-hu/ ↦ [kɔ́-hú]	'in war'
b.	/bɛ̀lɛ̀-hu ↦ [bɛ̀lɛ̀-hù]	'in trousers'
c.	/mbû-hu/ ↦ [mbú-hù]	'in owl'
d.	/mbǎ-hu/ ↦ [mbà-hú]	'in rice'

The complete analysis for this pattern would then have to not only perform tone spread or tone split as needed, but would also have to determine which process to apply to a given word. The formulas would therefore be more complex (in the sense of longer) than those presented in §5.7, but still QF. But this example raises the question of in what other ways can distinct tone processes interact, and will that interaction ever require a non-QF analysis to combine otherwise QF phenomena.

More generally, an exhaustive study of possible tone phenomena that classifies each in terms of the scheme we present in Table 1—as well as a set of possible interactions once multiple tone processes co-occur—would provide a more thorough understanding of the computational nature of tone maps.

Lastly, we have shown how the use of ARs instead of strings allows certain (but not all) 'non-local' phenomena to be modeled as QF, in the very spirit in which ARs were originally introduced. But this approach raises the important question of whether in the case of A-ISL the classification is in fact negated by the increased complexity of the representation. In other words, are we simply passing the buck of the complexity off to the representation instead of the map itself?

The results presented in this paper do indicate that the ISL maps are a proper subset of the A-ISL maps. The larger question is thus one of situated the class of A-ISL in a larger hierarchy of computational complexity as has been done in a lot of previous work (e.g., (Gainor et al., 2012; Heinz and Lai, 2013), among others). This hierarchy typically starts with the regular relations at the top, citing the relative learnability advantages of various regions of subregular mappings. By hypothesis, A-ISL then falls somewhere between the regular relations and the ISL functions. Proving the exact relationship among these classes is then an important future direction of this work.

8 Conclusion

We have introduced the computational property of A-ISL, which is a map that can be modeled as a graph interpretation using QF FO formulas over an autosegmental graph. We used this notion along with the previously defined property of ISL—describable with QF FO formuals over string graphs—to categorize a range of tone maps. 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 autosegmental representations. Collectively these results add to our understanding of the overall computational nature of phonological processes as well as the particular way in which ARs render non-local processes local.

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

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

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

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, April.

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 Andras Kornai and Marco Kuhlmann, editors, *Proceedings of the 13th Meeting on Mathematics of Language*, Sofia, Bulgaria.

Larry Hyman and Francis X. Katamba. 1993. A new approach to tone in luganda. *Language*, 69:34–67.

Larry Hyman and Francis X. Katamba. 2010. Tone, syntax and prosodic domains in Luganda. In Laura Downing, Annie Rialland, Jean-Marc Beltzung, Sophie Manus, Cdric 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.

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.

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

Adam Jardine. 2016b. *Locality and non-linear representations in tonal phonology*. Ph.D. thesis, University of Delaware.

Adam Jardine. 2017. On the logical complexity of autosegmental representations. In Makoto Kanazawa, Philippe de Groote, and Mehrnoosh Sadrzadeh, editors, *Proceedings of the 15th Meeting on the Mathematics of Language*, pages 22–35, London, UK, July. Association for Computational Linguistics.

Adam Jardine. to appear. The expressivity of autosegmental grammars.

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.

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

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

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

Scott Meyers. 1987. *Tone and the structure of words in Shona*. Ph.D. thesis, University of Massachusetts, Amherst.

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.

Thilo Schadeberg. 1978. Über die töne der nominalen und pronominalen formen im rimi. *Afrika und Übersee*, 56:189–209.

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.