On the Logical Complexity of Autosegmental Representations

Adam Jardine
Rutgers University
adam.jardine@rutgers.edu

Abstract

Autosegmental mapping from disjoint strings of tones and tone-bearing units, a commonly used mechanism in phonological analyses of tone patterns, is shown to not be definable in monadic second-order logic. This is abnormally complex in comparison to other phonological mappings, which have been shown to be monadic second-order definable. In contrast, generation of autosegmental structures from strings is demonstrated to be first-order definable.

1 Introduction

This paper applies *logical transductions* as introduced by Courcelle (1994) to study the cognitive complexity of non-string representations and transformations in phonology. Generative phonology studies phonological patterns both in terms of *transformations*, or relations between input *underlying representations* (URs) and output *surface representations* (SRs), and *phonotactics*, or generalizations about the well-formedness of SRs. Studies of the computational complexity of these patterns have established clear bounds on the expressivity needed to describe them. For example, Johnson (1972) and Kaplan and Kay (1994) showed that the ordered rewrite-rule grammars of Chomsky and Halle (1968) describe exactly Regular string relations, and more recent cross-linguistic studies have shown that phonological transformations fall into more restrictive subclasses of the Regular class (Chandlee and Heinz, 2012; Chandlee, 2014; Heinz and Lai, 2013; Payne, 2014; Jardine, 2016a). Similarly, phonotactic patterns have been shown to fall into sub-Regular classes of formal languages (Heinz, 2009, 2010; Rogers et al., 2013). This has led to a hypothesis that there is a sub-Regular bound on phonology (see, e.g., Heinz and Idsardi, 2013), which has clear connections to cognitive complexity (Rogers and Pullum, 2011; Rogers et al., 2013) and how humans learn sound patterns (Heinz, 2009, 2010; Lai, 2015).

However, these complexity classes are defined in terms of strings, and since the advent of autosegmental phonology (Goldsmith, 1976), generative phonology has commonly employed the use of non-string structures. Perhaps the most commonly used of these has been *autosegmental representations* (ARs), which represent words with graph structures in which disjoint strings are associated to one another in some fashion. For example, Fig. 1 shows an autosegmental derivation for the Mende word [félàmà] 'junction', which is comprised of a high-toned syllable followed by two low-toned syllables (following convention syllables are represented with σ).

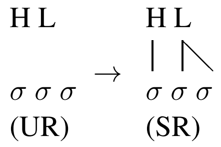


Figure 1: AR derivation for [félàmà] 'junction'

In the AR, the tone pattern of [félàmà] is represented as a HL (high-low) string associated to three syllables as depicted on the right-hand side of Fig. 1 (with association depicted by straight lines). While finite-state models of ARs and AR transformations exist, much of this work has found the need to use enriched automata with additional tapes (Kay, 1987; Wiebe, 1992; Kornai, 1995) or synchronized states (Bird and Ellison, 1994).

Instead, this paper takes a *logical* approach to studying the complexity of autosegmental representations (thus following the work of Bird and Klein, 1990; Jardine, 2014), as it allows

Proceedings of the 15th Meeting on the Mathematics of Language, pages 22–35,
London, UK, July 13–14, 2017. ©2017 Association for Computational Linguistics

for more flexibility with respect to the structures we can describe. This builds on a few key results. First, the Regular sets of strings are exactly those definable by *monadic second-order* (MSO) logic (Büchi, 1960; Elgot, 1961; Trakhtenbrot, 1961). Second, Courcelle (1994) introduced MSO *transductions* for graph structures, in which the output structure is determined as an MSO interpretation of the input structure. As MSO-definable string transductions subsume Regular functions (Filiot and Reynier, 2016), we can then recast the Regular hypothesis for phonology in logical terms: phonology is at most MSO-definable.

This leads to the two results of this paper, one negative and one positive. First, *tone mapping* transformations, in which an unassociated AR is mapped to a fully associated one—exemplified in Fig. 1—are not MSO-definable. Second, (at least some) ARs are *first-order* (FO) definable from strings; i.e., we can write a FO transduction from a string representing a sequence of toned syllables to its corresponding AR. Because this transduction is defined in the terms of the FO language of the input, this means that any FO formula we write over these ARs can be translated into the FO logic of their corresponding strings. This means that any FO constraint written over these ARs still describes a Regular set of strings—i.e., that ARs are not significantly more expressive than strings.

This paper is structured as follows. §2 introduces string models and logic, and §3 details how this relates to the study of phonology. §4 discusses the non-definability of tone mapping in MSO, and §5 discusses the FO-definability of ARs from strings. §6 concludes.

2 Preliminaries

2.1 String models and logics

Let an *alphabet* Σ be a finite set of symbols and a *string* w be a sequence of symbols in Σ; let $|w|$ denote the length of w. Let Σ^* represent all possible strings over Σ, including the *empty string* λ ($|\lambda| = 0$). A *stringset* (or *formal language*) is some subset of Σ^*. For some $\sigma \in \Sigma$, σ^n denotes the string consisting of n repetitions of σ.

A *relational model* $\langle U, R_1, R_2, ..., R_n\rangle$ is a representation of some structure with a universe U of elements and n relations $R_i \subseteq U^k$ for some finite k. We can represent a string $w \in \Sigma^*$ with a *finite* relational model $\mathcal{M}_w = \langle U, \prec, (P_\sigma)_{\sigma\in\Sigma}\rangle$ where $U = \{1, 2, ..., |w|\}$ is an initial segment of the natural numbers representing the positions in the string, $\prec$ is a binary relation representing the natural order over the positions in the string, and each P_σ is a unary relation representing the set of positions containing the symbol σ. For example, for $\Sigma = \{a, b\}$ the model for the string aba is

$$\mathcal{M}_{aba} = \big\langle \{1,2,3\}_U, \{(1,2),(1,3),(2,3)\}_\prec, \\ \{1,3\}_{P_a}, \{2\}_{P_b} \big\rangle.$$

We can use these models of this form to define a *first order* (FO) predicate logic over strings in Σ^*. Let $x, y, ...$ denote *variables* that range over positions in a string. For variables x and y, we can then use $x \prec y$ and $\sigma(x)$ for each $\sigma \in \Sigma$ as *atomic predicates* which are true when x and y are interpreted as positions related by $\prec$ in a string model and when x is interpreted as a position in the unary relation P_σ of a model, respectively. We also assume an additional atomic predicate $x = y$ which is true when x and y are interpreted as the same position. A FO logic is then the set of *formulas* built recursively out of these atomic predicates and the logical connectives $\neg, \wedge, \vee, \rightarrow$ and the quantifiers $\exists, \forall$ in the usual way. A *free variable* is a variable not bound by a quantifier; we write $\varphi(x_1, x_2, ..., x_n)$ to indicate that $x_1, x_2, ..., x_n$ is the exhaustive set of free variables in a FO formula φ. For example, we can define the following useful formulas with one free variable:

$$\texttt{first}(x) \overset{\text{def}}{=} (\forall y)[\neg y \prec x]$$

$$\texttt{last}(x) \overset{\text{def}}{=} (\forall y)[\neg x \prec y]$$

We also define a two-variable formula for the successor relation (using infix notation).

$$x \triangleleft y \overset{\text{def}}{=} x \prec y \wedge (\forall z)[\neg(x \prec z \wedge z \prec y)]$$

A formula φ with no free variables is called a *sentence*. Let satisfaction of a model $\mathcal{M}$ of φ, written $\mathcal{M} \models \varphi$, be defined in the usual way. The set of strings $L(\varphi)$ described by φ is the set of strings $\{w \in \Sigma^* | \mathcal{M}_w \models \varphi\}$. For example, if

$$\texttt{last}_a \overset{\text{def}}{=} (\forall x)[\texttt{last(x)} \rightarrow a(x)]$$

then $L(\texttt{last}_a)$ is the set of strings that end in a.

A *monadic second order* (MSO) logic is a FO logic extended with the ability to quantify over arbitrary sets in the string. Let *set variables* $X, Y, ...$ which range over sets of positions in a string. A MSO logic is thus FO logic to which we add the atomic formulas $X(x)$, $Y(x)$, etc., which are true when x is interpreted as a position in the set assigned to X, Y, etc., and in which $\exists$ and $\forall$ can also bind set variables.

It is well-known that FO sentences over string models with $\prec$ describe exactly the Star-Free stringsets (McNaughton and Papert, 1971) whereas MSO sentences describe exactly the Regular stringsets (Büchi, 1960; Elgot, 1961; Trakhtenbrot, 1961).

2.2 Logically definable transductions

We can also use logic to define a *transduction* from an input structure to an output structure, as first introduced by Courcelle (1994) for graphs and later related to string transductions and their automata-theoretic characterizations (Engelfriet and Hoogeboom, 2001; Filiot, 2015). (For an overview of related work see Filiot and Reynier 2016.)

In such a logical transduction, the output structure is defined by an *interpretation* over a finite number of copies of the input structure (where 'interpretation' is used in the sense of a translation from the logical language of one structure into that of another; see, e.g. Hodges 1997). MSO and FO transductions are defined as follows.

Definition 1 (MSO/FO transduction) *Given some natural number k, an input alphabet Σ and an output alphabet Γ, an* MSO *(resp.* FO*)* transduction *is defined by*

- $\varphi_{\texttt{dom}}$*, a* domain formula*, or sentence in the MSO (FO) logic of the input that defines the domain of the transduction,*

- *For each $1 \leq n \leq k$ and $\gamma \in \Gamma$, a formula $\varphi^n_\gamma(x)$ in the MSO (FO) logic of the input with exactly one free variable, and*

- *For each $1 \leq n, m \leq k$, a formula $\varphi^{n,m}_{\prec_\Gamma}(x, y)$ in the MSO (FO) logic of the input with exactly two free variables*

To restrict our domain to strings in Σ^*, we include in our domain formula the sentence $\texttt{string}_\Sigma$ as defined by

$$\begin{aligned}\texttt{string}_\Sigma \overset{\text{def}}{=} & (\forall x)[\textstyle\bigvee_{\sigma \in \Sigma} \sigma(x)] \wedge \\ & (\forall x)[\textstyle\bigwedge_{\sigma \neq \sigma' \in \Sigma} \sigma(x) \rightarrow \neg\sigma'(x)] \wedge \\ & (\forall x, y, z)[x \triangleleft y \wedge x \triangleleft z \rightarrow y = z] \wedge \\ & (\forall x, y)[\texttt{first}(x) \wedge \texttt{first}(y) \rightarrow \\ & \qquad\qquad\qquad\qquad x = y]\end{aligned}$$

The output of such a transduction is defined as follows. For each position x in the input and for each n for which exactly one $\varphi^n_\gamma(x)$ is true, a copy of x labeled γ appears in the output. For each pair of positions x, y and for each pair n, m for which $\varphi^{n,m}_{\prec_\Gamma}$ is true, the nth copy of x precedes the mth copy of y with respect to the output ordering $\prec_\Gamma$ on positions in the output. For example, to rewrite strings over the alphabet $\Sigma = \Gamma = \{a, b\}$ such that each b immediately following another b in the input is written out as an a then we set $k = 1$ and

$$\begin{aligned}\varphi_{\texttt{dom}} & \overset{\text{def}}{=} \texttt{string}_\Sigma \\ \varphi^1_a(x) & \overset{\text{def}}{=} a(x) \vee (b(x) \wedge (\exists y)[y \triangleleft x \wedge b(y)]) \\ \varphi^1_b(x) & \overset{\text{def}}{=} b(x) \wedge \neg(\exists y)[y \triangleleft x \wedge b(y)] \\ \varphi^{1,1}_{\prec_\Gamma}(x, y) & \overset{\text{def}}{=} x \prec y\end{aligned}$$

This transduction is illustrated in Fig. 2 for an input string $abba$.

First, $\varphi^1_a(x)$ is defined to be true in the output for all positions x in the input that are either labeled a or are labeled b but also succeed some input b. Thus, the copies corresponding to each a in the input are labeled as in the output, as well as the copy for the second input b in the input. Likewise, $\varphi^1_b(x)$ is defined to be true in the output for all positions x in the input that are labeled b and do not succeed another b. In Fig. 2, this is true for the first b in the input, so its copy is also labeled b. As the output order $\varphi^{1,1}_{\prec_\Gamma}(x, y)$ is defined to be true when the input order $x \prec y$ is true, the order is preserved exactly in the output.

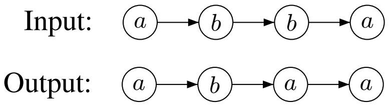


Figure 2: Replacing b with a following an input b

Note as there is only one interpretation per input, these transductions are functional. (For non-functional MSO transductions, see Engelfriet and Hoogeboom 2001.)

To give one more example, we can define a transduction that 'doubles' a string, i.e. given an input $w \in \Sigma^*$ outputs ww. We set $k = 2$, $\varphi_{\texttt{dom}} \overset{\text{def}}{=} \texttt{string}_\Sigma$, and

$$\begin{aligned}\varphi^1_a(x) \overset{\text{def}}{=} \varphi^2_a(x) \overset{\text{def}}{=} a(x) \quad & \varphi^{1,2}_{\prec_\Gamma}(x, y) \overset{\text{def}}{=} x < y\texttt{True} \\ \varphi^1_b(x) \overset{\text{def}}{=} \varphi^2_b(x) \overset{\text{def}}{=} b(x) \quad & \varphi^{2,1}_{\prec_\Gamma}(x, y) \overset{\text{def}}{=} \texttt{False} \\ \varphi^{1,1}_{\prec_\Gamma}(x, y) \overset{\text{def}}{=} \varphi^{2,2}_{\prec_\Gamma} & (x, y) \overset{\text{def}}{=} x \prec y\end{aligned}$$

(`True` and `False` indicate a formula is evaluated to true or false for any input positions.)

This is interpreted as follows. Each a and b in the input is given two identical copies in the output. As both $\varphi^{1,1}_{\prec_\Gamma}(x,y)$ and $\varphi^{2,2}_{\prec_\Gamma}(x,y)$ are set equal to $x \prec y$, the first set of copies has the same order as the input, as does the second. That $\varphi^{1,2}_{\prec_\Gamma}(x,y)$ is set to `True` states that all copies in the first set precede all copies in the second set; this establishes an order between the two sets of copies. Finally, setting $\varphi^{2,1}_{\prec_\Gamma}(x,y)$ to `False` ensures that second copies never precede first copies. An example with *abba* in the input is depicted in Fig. 3.

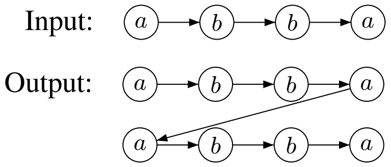


Figure 3: Doubling *abba*

This second example shows that we can freely manipulate the order of elements in the output; indeed, the output need not be a string. In fact, we can define the output structure to have a new binary relation R not present in the input structure by definining a predicate $\varphi^{m,n}_{R}(x,y)$ in terms of the MSO logic of the input structure. We make use of this in §4.2. Importantly, as they are defined in terms of interpretations, both MSO and FO transductions are closed under composition for graph structures in general (Courcelle, 1994).

3 Logic and phonology

Because of its well-known connections to computational complexity, we can apply logic to the study of the complexity of phonological patterns. This section reviews relevant results from the study of phonotactic (phonological surface well-formedness) patterns as stringsets and phonological transformations (mappings from URs to SRs) as transductions. Both show that MSO-definability is a clear, if loose, bound on the complexity of phonology.

3.1 Stringsets

Phonotactics are language-specific well-formedness constraints on how sounds can be combined to create words. An example from Kagoshima Japanese is given in Table 1: all words have a high tone either on the final or penultimate mora (Kubozono, 2012). (The high tone is marked with an acute accent [á] on the vowel.)

hána	'nose'	HL
sakúra	'cherry blossom'	LHL
kagaríbi	'watch fire'	LLHL
...		...
haná	'flower'	LH
usagí	'rabbit'	LLH
kakimonó	'document'	LLLH
...		...

Table 1: Kagoshima Japanese tone well-formedness

Such constraints can be modeled as stringsets. For example, given the alphabet $\{\mathrm{H}, \mathrm{L}\}$, the Kagoshima pattern can be modeled as the set of strings $\{\mathrm{HL}, \mathrm{LHL}, \mathrm{LLHL}, ..., \mathrm{LH}, \mathrm{LLH}, \mathrm{LLLH}, ...\}$, where Ls and Hs represent low- and high-toned moras, respectively.

All previous work on natural language phonotactics as stringsets has found these patterns to be at most Regular stringsets, with all but a few exceptions being sub-Star Free (Heinz, 2007, 2009, 2010; Heinz et al., 2011; Rogers et al., 2013). In logical terms, this means that definability in MSO is a clear bound on the complexity of phonotactics, with most patterns being FO-definable. To illustrate, the stringset representing the Kagoshima tone pattern can be defined with the FO sentence

$$(\mathtt{last}_{\mathrm{H}} \vee \mathtt{penult}_{\mathrm{H}}) \wedge \\ (\forall x, y)[(\mathrm{H}(x) \wedge \mathrm{H}(y)) \rightarrow x = y],$$

where $\mathtt{last}_{\mathrm{H}}$ is defined as $\mathtt{last}_{a}$ above and $\mathtt{penult}_{\mathrm{H}}$ is defined as $(\forall x, \exists y)[(x \triangleleft y \wedge \mathtt{last}(y)) \rightarrow \mathrm{H}(x)]$. This sentence describes the set of strings that has exactly one H either in final or penultimate position.

These results are important for a theory of phonology because they allow for the hypothesis that phonotactics are at most MSO-definable, a hypothesis which can be interpreted in terms of cognitive complexity (Rogers and Pullum, 2011; Rogers et al., 2013) and how humans learn phonotactics (Heinz, 2010; Lai, 2015; McMullin and Hansson, 2015). More restrictive characterizations exist, based on subclasses of the Star-Free stringsets (see, e.g., Heinz, 2010), but for the present purposes it is enough to consider FO- and MSO-definability.

3.2 Transductions

We can also fruitfully apply logical transductions to phonological theory (Heinz, forthcoming), as mainstream theories of generative phonology aim to explain linguistic sound patterns through a transformation from an input UR to an output SR (Chomsky and Halle, 1968; Prince and Smolensky, 2004). Indeed, these transformations have been studied from an automata-theoretic perspective, leading to restrictive characterizations of phonology. Johnson (1972) and Kaplan and Kay (1994) show that the phonological rewrite rules of Chomsky and Halle (1968) are describable with finite state machines; that is, that they describe Regular relations. Subsequent work on phonological transformations has demonstrated for a wide variety of processes—including local assimilation, deletion, and epenthesis (Chandlee, 2014), dissimilation (Payne, 2014), metathesis (Chandlee and Heinz, 2012), and vowel harmony (Heinz and Lai, 2013)—to lie in even more restrictive subclasses of Regular string transductions. The single known possible exception to this is full reduplication—i.e. the copying over of an entire input form, as in the Indonesian *buku-buku* 'books', lit. 'book-book' (Sneddon et al., 2010). This is not a Regular relation, although it can be argued that this process is morphological and not phonology proper (for discussion see Chandlee and Heinz, 2012).

From the logical perspective, all of these results place phonological transformations squarely within the class of MSO-definable transductions. Any functional Regular relation is MSO-definable (Filiot and Reynier, 2016), so any phonological transformation describable with a (functional) rewrite rule is MSO-definable. Even full reduplication is FO-definable, as witnessed by the string doubling transduction defined in §2.2. Thus, MSO-definability appears to be a loose, yet clear, bound on the computational complexity of phonological transformations.

3.3 Interim summary

The above has reviewed the evidence for MSO-definability as a complexity bound on phonology. The advantage of viewing such a complexity bound in logical terms is that we are able to view the complexity of both phonotactics and transformations in unified terms.

A further advantage of the logical perspective is that it allows us to study the complexity of non-string representations in the same terms. The remainder of the paper studies *autosegmental representations* (ARs) in the same terms.

4 The complexity of tone mapping

This section motivates *tone mappings* and ARs using a well-known empirical case, then it is shown that tone mapping is not MSO-definable.

4.1 Tone mapping in Mende

Mende (Leben, 1973, 1978) is a classic example of a tone pattern which has been argued to be best analyzed in terms of autosegmental mapping of tones to syllables. Mende nouns fall in to one of five categories: 1) words for which all syllables are pronounced with a high tone (e.g. [háwámá] 'waist'), 2) words for which all syllables are pronounced with a low tone (e.g. [kpàkàlì] 'three-legged chair'), 3) words which begin high but end low (e.g. [mbû] 'owl' and [félàmà] 'junction'), 4) words which begin low but end high (e.g. [mbǎ] 'rice' and [ndàvúlá] 'sling'), and 5) words which show a rising-falling pattern (e.g. [mbǎ̂] 'companion' and [nìkílì] 'peanut'). These five categories are exemplified by the forms in Table 2, where tones are indicated as diacritics on the vowels as follows: [á] = high tone, [à] = low tone, [â] = falling tone, [ǎ] = rising tone, and [ǎ̂] = rising-falling tone. The columns are of 1-, 2-, and 3-syllable words.

1. H	kɔ́ 'war'	pɛ́lɛ́ 'house'	háwámá 'waist'
2. L	kpà 'debt'	bɛ̀lɛ̀ 'pants'	kpàkàlì 'chair'
3. HL	mbû 'owl'	ngílà 'dog'	félàmà 'junction'
4. LH	mbǎ 'rice'	fàndé 'cotton'	ndàvúlá 'sling'
5. LHL	mbǎ̂ 'companion'	nyàhâ 'woman'	nìkílì 'peanut'

Table 2: Mende noun tone (Leben, 1978)

Of interest is the fact that *contour tones*—that is, the rising and falling toned-syllables—and *plateaus*, or sequences of like-toned syllables, only occur on the right edge of the word. For example, [nyáhâ] 'woman' is attested, but a word like *[nyâhá], with a rising toned-syllable on the left edge, is not attested. Likewise, words like [félàmà] 'junction', with a sequence of two low-toned syllables on the right edge, are commonly attested, whereas words like *[fèlàmá], are rare.[1] Furthermore, Mende words conform to one of the five tonal shapes exemplified in Table 2; words showing a falling-rising pattern, for example, are unattested.

Furthermore, these tonal shapes are maintained when toneless suffixes are affixed to the noun, resulting in the tone of the suffix varying depending on the tone pattern of its root. The following data illustrate this with the toneless suffix /-ma/ 'on'.

	Isolation	Suffixed	
H	pɛ́lɛ́	pɛ́lɛ́-má	'war'
HL	mbû	mbú-mà	'owl'
	ngílà	ngílà-mà	'dog'
LHL	nyàhâ	nyàhá-mà	'companion'

Table 3: Mende /-ma/ suffix tone (Leben, 1978)

Note that the suffixed forms also preserve the generalizations noted above restricting contours and plateaus to the right edge, and so the tone pattern 'stretches' to accomodate the new syllable. Thus [mbû] 'owl', which has a contour falling tone in isolation, is realized with a sequence of pure high and low-toned syllables as [mbú-mà] 'on owl' when suffixed.

Following a proposal by Leben (1973), and subsequent work in autosegmental phonology (e.g., Goldsmith, 1976; Pulleyblank, 1986; Yip, 2002) tone patterns like Mende's have been explained by a left-to-right mapping of a *melody*, or string of tonal units, to a string of syllables. These disjoint strings are referred to as *tiers*, and the representation as a whole is an AR. For example, the words in Table 2, Row 3 share a HL (high-low) melody, which is then mapped to the syllables in the words as depicted in Fig. 4. Following convention, syllables are denoted with σ and the *association* relation depicted as lines drawn between units on distinct tiers.

Thus, the contour falling tone of [mbû] 'owl is the result of an HL sequence associating to a single syllable; likewise the plateau of low-toned syllables in [félàm̀a] 'junction' is the result of an L tone associating to multiple syllables. The question then is how to restrict association such that this multiple association occurs only on the right edge of the word. Because this association pattern holds for all lexical items, including the suffixed forms in Table 3, it is thus entirely predictable and taken not to be present in the UR of a word. Thus, it must be created by some phonological transformation that associates tones to syllables. This transformation has been analyzed as proceeding according to laws often referred to as the *well-formedness conditions* (WFCs). The following definition is due to Yip (2002).

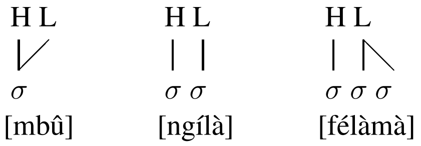


Figure 4: ARs illustrating mapping of HL melody to words of various syllable length

Definition 2 *The well-formedness conditions*

a. Every syllable must have a tone.

b. Every tone must be associated to some syllable.

c. Association proceeds one-to-one, left-to-right.

d. Association lines must not cross.

Intuitively, the WFCs ensure that in the SR, every tone is associated to some syllable, and vice-versa, by a step-by-step process in which first tone and first syllable are associated, then the second tone and second syllable, and so on. (It bears mentioning that 'one-to-one' here is used not as it is to describe mathematical functions, but in terms of how pairs of tones and syllables are associated one after another.) If there are remaining tones or syllables on the right edge of a tier that have not been paired off, WFCs (a), (b), and (d) associate them to the rightmost unit on the opposite tier: (a) and (b) require all units to be associated, but (d) forbids the crossing of any existing associations to do so. Fig. 5 shows how this process works for [nyàhâ] 'woman', [nyàhá-mà] 'on woman', both of which have an underlying LHL melody, and [félàmà] 'junction', which has an underlying HL melody. This figure demonstrates that the WFCs explain the generalization in Mende that contours and plateaus only occur on the right edge of the word through a transformation from a UR with no association to a fully associated SR.

The WFCs have been shown not to be strictly universal; whether or not tones are as-

[1] See Dwyer (1978) and rebuttal by Leben (1978) for discussion about exceptions to the generalizations stated here.

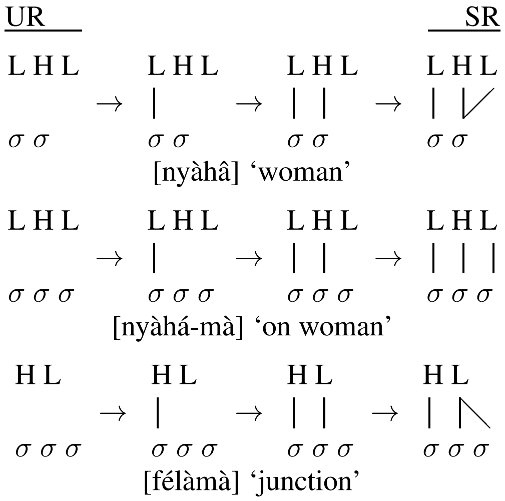


Figure 5: Step-by-step breakdown of the association transformation

sociated from left-to-right or right-to-left, or whether or not contours are built out of left-over tones or are simply left unpronounced, have been shown to vary from language to language (Goldsmith, 1976; Newman, 1986; Pulleyblank, 1986; Hewitt and Prince, 1989; Archangeli and Pulleyblank, 1994; Yip, 2002). Also, there have since been non-derivational approaches to tone mapping (Zoll, 2003). However, all generative explanations of tone mapping patterns that use ARs rely on the idea of an unassociated UR being transformed into a SR with associations, with one-to-one association forming the basis of the transformation. The following shows that such a transformation is not a MSO-definable transduction.

4.2 Tone mapping as a logical transduction

We can characterize this transformation, as depicted in Fig. 6, as a transduction that takes two strings of length n and m, respectively ($n, m > 0$), as input and adds an association relation between the positions in the strings that follows the WFCs outlined in Def. 2. As long as we can assume there is some property that distinguishes between units on each tier, we can abstract away from distinctions between units on a particular tier and instead focus on predicates $a(x)$ and $b(x)$ which are true if and only if x is on the 'upper' and 'lower' tier, respectively. (For example, $a(x)$ can mean 'x is a tone' and $b(x)$ can mean 'x is a syllable'.)

In terms of relational models, the transduction takes models of the form

$$\langle U, \prec, P_a, P_b\rangle$$

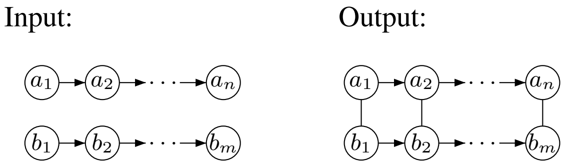


Figure 6: The association transduction

and describes a model of the form

$$\langle U', \prec', \circ, R_a, R_b\rangle,$$

where $\circ$ denotes a new relation representing association. This relation must conform to the WFCs in Def. 2; this is formalized in Def. 3 of the association relation.

Definition 3 *For an AR whose tiers are a pair of disjoint strings $a_1a_2...a_n$ and $b_1b_2...b_m$, and for ℓ being the lesser of n and m, the association relation $\circ$ is the unique relation comprised of the symmetric closure of all pairs (a_i, b_i) for $1 \leq i \leq \ell$ unioned with $(a_n, b_{n+1}), ..., (a_n, b_m)$ if $n = \ell$ or $(a_{m+1}, b_m), ..., (a_n, b_m)$ if $m = \ell$.*

To see why this definition matches that in Def. 2, note that WFC (c) in Def. 2, which stipulates one-to-one, left-to-right association, requires that $(a_1, b_1), (a_2, b_2), ..., (a_\ell, b_\ell) \in \circ$. If $n \neq m$, this leaves either a final segment $a_{\ell+1}a_{\ell+2}...a_n$ (if $m = \ell$) or $b_{\ell+1}b_{\ell+2}...b_m$ (if $n = \ell$) that must also be associated, per WFCs (a) and (b). WFC (d), which bans line crossing, stipulates that these final elements cannot 'reach back' and associate to anything except for the final element on the opposite tier. So if $m = \ell$ then $(a_{m+1}, b_m), (a_{m+2}, b_m)..., (a_n, b_m) \in \circ$ and if $n = \ell$ then $(a_n, b_{n+1}), (a_n, b_{n+2})..., (a_n, b_m) \in \circ$. We take the symmetric closure as association is usually regarded as symmetric (Kornai, 1995).

No MSO definition of this relation is possible. To show how, we can rely on two facts established above: 1) MSO transductions are essentially interpretations where relations in the output structure are defined in terms of the logical language of the input structure; and 2) MSO transductions are closed under composition.

First, we can instead consider as input strings of the form a^nb^m (again, with $n, m > 0$). This set is MSO definable, as witnessed by the formula $\texttt{string}_{a^nb^m}$ defined below:

$$\texttt{string}_{a^nb^m} \overset{\text{def}}{=} (\forall x, y)[\neg(b(x) \wedge a(y) \wedge x \prec y)] \wedge (\exists x, y)[a(x) \wedge b(y)]$$

These strings are equivalent, in terms of MSO, to pairs of disjoint strings of shape a^n and b^m. The reason is that we can write a MSO transduction from one to another. If $\prec$ is the ordering in the input $a^n b^m$ string, we simply define the order $\prec_\Gamma$ for the output structure such that it omits all precedence between a positions and b positions.

$$\begin{aligned}
\varphi_{\texttt{dom}} &\overset{\text{def}}{=} \texttt{string}_\Sigma \wedge \texttt{string}_{a^n b^m} \\
\varphi^1_a(x) &\overset{\text{def}}{=} a(x) \\
\varphi^1_b(x) &\overset{\text{def}}{=} b(x) \\
\varphi^{1,1}_{\prec_\Gamma}(x,y) &\overset{\text{def}}{=} x \prec y \wedge \neg(a(x) \wedge b(y))
\end{aligned}$$

An example of this is given in Fig. 7.

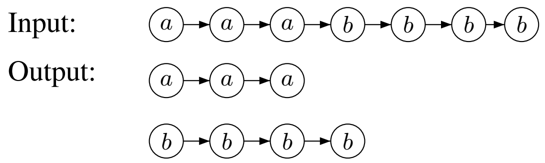


Figure 7: Relating $a^n b^m$ strings to disjoint tiers a^n and b^m.

Thus, we know that MSO statements over pairs of disjoint strings a^n and b^m are equivalent to statements over strings in $a^n b^m$. We can then use this to prove Theorem 1.

Theorem 1 *Association between two disjoint strings according to the WFCs in Definition 2 is not MSO definable.*

Proof: The proof is by contradiction. Assume the converse, that we can define using MSO a transduction that takes disjoint pairs of strings a^n and b^m and outputs them as associated autosegmental representations with some association relation $\circ$ that obeys the WFCs as defined in Def. 3.

With this relation we can write a sentence $\varphi_{\texttt{eq}} \overset{\text{def}}{=} (\forall x, \exists y, \forall z)[x \circ y \wedge (x \circ z \rightarrow z = y)]$, which holds that every position is associated to exactly one position. If $\circ$ obeys the WFCs, then per the discussion of the structure of $\circ$ in Def. 3, $\varphi_{\texttt{eq}}$ is only true for structures whose association relation is the set of pairs of the form (a_i, b_i) (and their converse). That is, the only structures for which $\varphi_{\texttt{eq}}$ are true are pairs of disjoint strings a^n and b^m for which $n = m$.

Now consider strings of the form $a^n b^m$. As shown above, there is a MSO transduction from these strings to disjoint pairs of a^n and b^m. By assumption, there is then a MSO transduction that adds an association relation $\circ$ to these pairs. Because MSO transductions are closed under composition, then $\varphi_{\texttt{eq}}$ can be written as the MSO language of strings $\{a,b\}^*$. Thus the sentence $\texttt{string}_{a^n b^m} \wedge \varphi_{\texttt{eq}}$ restricts us exactly the set of strings $a^n b^m$ for which $n = m$. It is well-known that this set is not regular (see, e.g., Hopcroft et al., 2006), and thus not MSO definable. Thus we have a contradiction, and so the assumption must be false. □

Importantly, because MSO transductions are closed under composition, there is no breakdown of this process into a finite set of composite steps (such as those illustrated in Fig. 5) that are themselves MSO-definable. Note also that the proof highlights that in particular it is the *one-to-one* requirement on association that makes it not MSO-definable: it is this property that introduces the ability to check the parity of the a and b tiers.

4.3 Interpreting the result

We have thus shown that, as a transduction from an unassociated pair of tiers to an associated one following the WFCs in Def. 2, tone mapping is not MSO-definable. This puts it in sharp contrast to all other phonological UR-SR transductions for whom complexity results exist: as discussed in Sec. 3.2, these processes have been shown to be well within MSO-definability. This makes tone mapping aberrant in terms of its computational complexity. How do we interpret this result?

One answer is to take this as evidence that tone has access to more computational power than other parts of phonology. In fact, this has been argued by Jardine (2016a) on the basis of comparisons between the complexity of tonal phenomena and segmental phenomena when viewed as string transductions. However, an issue with this interpretation is that the tonal phenomena that Jardine cites are still Regular relations and thus MSO-definable. Thus tone mapping is still highly complex, even compared to other tonal phenomena.

Another possible interpretation is that tone mapping is simply an incorrect characterization of the data available. For example, both Dwyer (1978) and Shih and Inkelas (2015) take issue with Leben (1973)'s tone mapping characterization of Mende, and offer alternative explanations using representational assumptions that do not require tone mapping (or an analogue thereof). However, other tone patterns that have been successfully accounted for using tone mapping include those of Hausa (Newman, 1986, 2000), Kukuya (Hyman, 1987), and the wide variety of languages analyzed in Goldsmith (1976), Pulleyblank (1986), and Zoll (2003). The alternative explanations mentioned above have yet to be shown to enjoy the same broad empirical coverage (though future work may indeed show this).

Another explanation is that it is wrong to assume that the mapping generalization holds for tiers of unbounded length. The proof above relies on the fact that the input strings are of the form $a^n b^m$ for any n, m—if either n or m had some bound, the proof would no longer hold. Indeed, Yli-Jyrä (2013) gives a finite-state (and thus MSO-definable) implementation of tone mapping given the assumption that the tonal tier is bounded. However, it is not clear that this can be assumed for all cases. For example, in Kikuyu (Clements and Ford, 1979), morphological concatenation can extend the tonal tier before mapping occurs. Regardless, "tonal tiers must be bounded" is a hypothesis worth further testing, as the result here shows it has consequences for the complexity of phonology.

A final interpretation of the result is to posit that the one-to-one property of tone mapping as a phonological *universal* and thus is not relevant to the study of the complexity of *language-specific* phonological phenomena. As noted in §4, there are languages whose patterns have been shown to violate the WFCs in Def. 2 with respect to directionality and whether or not all tones or syllables are associated. However, the one-to-one property appears to be shared by all such patterns. Jardine (2016b, to appear) demonstrates for many of these patterns that, if one-to-one association is assumed in the representation, these language-specific constraints on association can be described with a restricted propositional logic, well within the complexity of MSO. Thus, if we isolate the one-to-one property of association, shown in the proof of Thm. 1 to be responsible for its non-definability in MSO, from the other aspects of tone mapping, then we can maintain MSO-definability as a cohesive bound on the complexity of language-specific phonological phenomena. How this separation might be implemented in a theory of tone will be left for future work.

5 Deriving autosegmental representations from single strings

The result in the previous section raises an important question: How powerful are ARs? Specifically, does invoking ARs allow for grammars that are too expressive to provide a reasonable theory of phonological patterns? To put it in more concrete terms, we can represent the tone pattern of a word either as a string of toned syllables or as an AR.[2] Table 4 gives some examples, two from Mende and one hypothetical, where strings are over an alphabet $\{\mathrm{H}, \mathrm{L}, \mathrm{F}, \mathrm{R}\}$ whose symbols represent high-, low-, falling-, and rising-toned syllables, respectively.

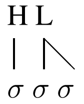
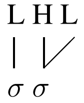
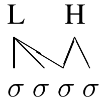
Form	String	AR
[félàmà] 'junction'	HLL	
[nyàhâ] 'woman'	LF	
(hypothetical)	LLRH	

Table 4: Strings and ARs

Thus, for any string over $\{\mathrm{H}, \mathrm{L}, \mathrm{F}, \mathrm{R}\}$, there is a corresponding autosegmental representation. Note that these ARs obey WFCs (a), (b), and (d) from Def. 2, in that each tone is associated to a syllable and vice versa, and these association lines do not cross. However, the AR for LLRH violates WFC (c), as the tones have not associated in a left-to-right manner. We can thus talk about ARs that obey (a), (b), and (d), but will ignore (c), as the latter would restrict us to a subset $\{\mathrm{H}, \mathrm{L}, \mathrm{F}, \mathrm{R}\}^*$.

We can then talk about strings and their corresponding ARs. Jardine and Heinz (2015) define such a relationship in terms of concatenation, but they do not address how this relationship connects to complexity. As discussed in §3.1, natural language phonotactics are largely describable with FO-definable stringsets. The question then is, given, for example, an FO logic over ARs, can we describe sets of strings that are not FO-definable? This is a valid question as, for example, even restrictions on FO over trees can generate Context

[2] We abstract away from the issue of whether tones are a property of moras, syllables, or some other unit. See, e.g., Yip (2002) for more on this issue.

Free stringsets (Rogers, 1997).

The following demonstrates otherwise: ARs are FO-definable from strings, and thus any FO formula over ARs is translatable into a FO formula in strings (given $\prec$). It should be noted that autosegmental phonology is not a monolithic theory, and in practice various definitions of ARs have been proposed (one formal overview can be found in Coleman and Local, 1991). A full survey of these and how they might be defined is beyond the scope of this paper; instead, this section focuses on demonstrating that basic ARs obeying the WFCs (a), (b), and (d) in Def. 2 are FO-definable from strings over $\{\mathrm{H}, \mathrm{L}, \mathrm{F}, \mathrm{R}\}$. In other words, we formalize the relationship between strings and ARs exemplified in Table 4. This illustrates that the fundamental ideas of autosegmental structure—distinct tiers associated to one another according to some well-formedness conditions—is FO-definable from strings.

5.1 Definition of transduction

We define a transduction from string models of the form

$$\langle U, \prec, P_{\mathrm{H}}, P_{\mathrm{L}}, P_{\mathrm{F}}, P_{\mathrm{R}} \rangle$$

to autosegmental models of the form

$$\left\langle U', \triangleleft', \circ, R_{\mathrm{H}}, R_{\mathrm{L}}, R_{\sigma} \right\rangle .$$

Essentially, we define the transformation from the second column of Table 4 to the third, for all strings in $\{\mathrm{H}, \mathrm{L}, \mathrm{F}, \mathrm{R}\}^*$.

We do this by defining the notion of a *span*, or a series of consecutive positions in U that share the same tone and thus will be associated to the same tone on the melody tier in the output AR. We then create extra copies of each element in U that represents a change in spans. These extra copies become the tones in the melody tier. Note that the order in the output is a successor relation $\triangleleft'$; this is not essential, but was chosen for two reasons. One, it is more straightforward to depict in the examples below. Two, its definition gives formal weight to an idea long noted by phonologists: local relationships (i.e. those over $\triangleleft'$) in ARs correspond to long-distance relationships (i.e. those over $\prec$) in strings (see, e.g., Odden, 1994).

First, we define some useful formula and notational shortcuts. The first denotes when y lies between some x and z.

$$x \prec y \prec z \overset{\text{def}}{=} x \prec y \wedge y \prec z$$

We then define formulas in the logic of the input string that represent the tonal relationships between units in the string. The following formula $\texttt{sametone}(x, y)$ is true when x ends with the same tone that y begins with (thus it is true for an H and an F pair as F starts high).

$$\begin{aligned} &\texttt{sametone}(x, y) \overset{\text{def}}{=} \\ &\quad (\mathrm{H}(x) \wedge \mathrm{H}(y)) \vee (\mathrm{L}(x) \wedge \mathrm{L}(y)) \vee \\ &\quad (\mathrm{H}(x) \wedge \mathrm{F}(y)) \vee (\mathrm{L}(x) \wedge \mathrm{R}(y)) \vee \\ &\qquad (\mathrm{R}(x) \wedge \mathrm{H}(y)) \vee (\mathrm{F}(x) \wedge \mathrm{L}(y)) \end{aligned}$$

We do this because we will create tones on the melody tier exactly at syllables where there is a change in tone. This marks the beginning of a *span* of one or more like-toned syllables.

$$\begin{aligned} &\texttt{spanfirst}(x) \overset{\text{def}}{=} \\ &\qquad (\forall y)[y \triangleleft x \rightarrow \neg\texttt{sametone}(x, y)] \end{aligned}$$

$$\begin{aligned} &\texttt{span}(x, y) \overset{\text{def}}{=} x \prec y \wedge \texttt{sametone}(x, y) \wedge \\ &\qquad (\forall z)[x \prec z \prec y \rightarrow \texttt{sametone}(x, z)] \end{aligned}$$

For example, the positions in the following strings that satisfy $\texttt{spanfirst}(x)$ are underlined: $\underline{\mathrm{HL}}\mathrm{L}$, $\underline{\mathrm{LF}}$, and $\underline{\mathrm{L}}\mathrm{LRH}$. Note that neither the R nor the H in LLRH satisfy this formula because $\texttt{sametone}(x, y)$ returns true when x is L and y is R and likewise when x is R and y is H.

We can then define the transduction from strings to autosegmental representations by setting $k = 3$. One set of copies transfers over the syllables, the next initial tones. The third set of copies is for creating the additional tones in the F and R contours. (In general, for strings whose symbols represent contours of at most length n, $k = n + 1$.)

We define the unary labeling relations in the autosegmental representation as follows.

$$\begin{aligned} \varphi^1_\sigma(x) &\overset{\text{def}}{=} \texttt{True} \\ \varphi^2_{\mathrm{H}}(x) &\overset{\text{def}}{=} (\mathrm{H}(x) \vee \mathrm{F}(x)) \wedge \texttt{spanfirst}(x) \\ \varphi^2_{\mathrm{L}}(x) &\overset{\text{def}}{=} (\mathrm{L}(x) \vee \mathrm{R}(x)) \wedge \texttt{spanfirst}(x) \\ \varphi^3_{\mathrm{H}}(x) &\overset{\text{def}}{=} \mathrm{R}(x) \qquad\qquad \varphi^3_{\mathrm{L}}(x) \overset{\text{def}}{=} \mathrm{F}(x) \end{aligned}$$

All other unary formulas are set to false; that is, $\varphi^1_{\mathrm{H}}(x) \overset{\text{def}}{=} \varphi^1_{\mathrm{L}}(x) \overset{\text{def}}{=} \varphi^2_\sigma(x) \overset{\text{def}}{=} \varphi^3_\sigma(x) \overset{\text{def}}{=} \texttt{False}$. This works as depicted in Fig. 8 for the strings HLL, LF, and LLRH. Each set of copies in the output is organized into a labeled row.

As $\varphi^1_\sigma(x)$ is set to $\texttt{True}$, every element in the input has a copy in set 1 labeled σ. This corresponds to the intuition that each position in the input string

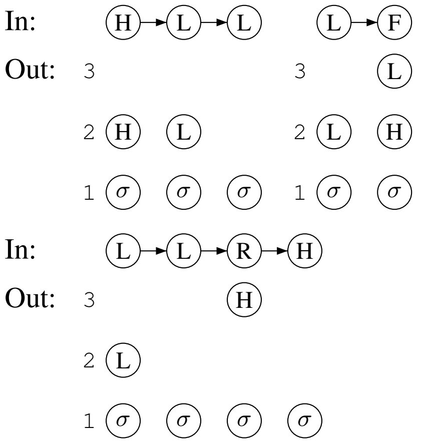


Figure 8: Creating elements in autosegmental representations.

represents a syllable. In copy set 2, Hs and Ls are copied over only at the positions in a string representing a change in tone, as $\varphi^2_{\text{H}}(x)$ and $\varphi^2_{\text{L}}(x)$ are defined to be true only when $\mathtt{spanfirst}(x)$ is true. (Note again that for the string LLRH, this is false for the positions labeled R and H.) Finally, $\varphi^3_{\text{H}}(x)$ and $\varphi^3_{\text{L}}(x)$ ensure that the contour-toned syllables F and R in the input are given a third copy L and H, respectively, in the output.

The next step is to define an order on the elements of the output. Again, we define the successor relation $\triangleleft'$. Like the definition of the unary relations, this definition will make use of the $\mathtt{spanfirst}(x)$ and $\mathtt{span}(x, y)$ formulas.

$$
\begin{aligned}
&\varphi^{1,1}_{\triangleleft'}(x,y) \stackrel{\text{def}}{=} x \triangleleft y\\
&\varphi^{2,2}_{\triangleleft'}(x,y) \stackrel{\text{def}}{=} x \prec y \wedge\\
&\qquad\qquad \mathtt{spanfirst}(x) \wedge \mathtt{spanfirst}(y) \wedge\\
&\qquad\qquad\qquad (\forall z)[x \prec z \prec y \rightarrow \mathtt{span}(x,z)]\\
&\varphi^{2,3}_{\triangleleft'}(x,y) \stackrel{\text{def}}{=}\\
&\quad \big(\mathtt{spanfirst}(x) \wedge (\text{F}(y) \vee \text{R}(y)) \wedge x = y\big) \vee\\
&\quad \big(\mathtt{spanfirst}(x) \wedge (\text{F}(x) \vee \text{R}(x)) \wedge \mathtt{span}(x,y)\big)\\
&\varphi^{3,2}_{\triangleleft'}(x,y) \stackrel{\text{def}}{=} (\text{F}(x) \vee \text{R}(x)) \wedge \mathtt{spanfirst}(y) \wedge\\
&\qquad\qquad\qquad\qquad \mathtt{span}(x,y)\\
&\varphi^{3,3}_{\triangleleft'}(x,y) \stackrel{\text{def}}{=} (\text{F}(x) \vee \text{R}(x)) \wedge (\text{F}(y) \vee \text{R}(y)) \wedge\\
&\qquad\qquad\qquad\qquad \mathtt{span}(x,y)
\end{aligned}
$$

We set $\varphi^{m,n}_{\triangleleft'}(x,y)$ for all other values of m and n to false. These definitions work, as illustrated in Fig. 9, as follows. The formula $\varphi^{1,1}_{\triangleleft'}(x,y)$ copies the successor relation from the input faithfully for the initial copies (i.e. those labeled σ). Next, $\varphi^{2,2}_{\triangleleft'}(x,y)$ draws a successor relation between the second copies of the initial positions for adjacent spans. Finally, $\varphi^{2,3}_{\triangleleft'}(x,y)$ and $\varphi^{3,2}_{\triangleleft'}(x,y)$ deal with the extra elements in a contour. The formula $\varphi^{2,3}_{\triangleleft'}(x,y)$ draws a successor relation from the second copy of a contour-toned syllable to its third (i.e., between the two tones in the contour) when the contour is first in a span (e.g. F in LF). In case the first part of a contour is part of a previous span (e.g. in the case of R in LLRH), it draws a successor relation from the first position in the previous span to the second part of the contour. The formulas $\varphi^{3,2}_{\triangleleft'}(x,y)$ and $\varphi^{3,3}_{\triangleleft'}(x,y)$ then similarly draw a successor relation from the second part of a contour that is the start of a span to the initial position of the next successive span (these latter two formulas are not used in the examples).

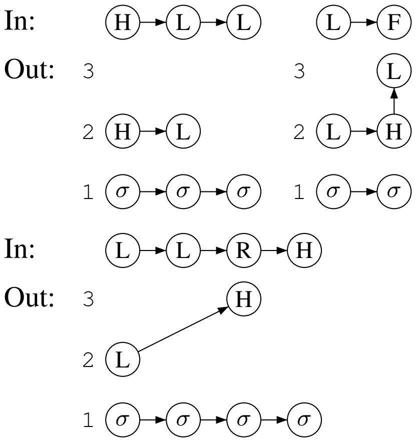


Figure 9: Creating the successor relation in autosegmental representations.

Finally, we draw the associations between the tiers. This is relatively simple: we define formulas that relate a second or third copy of a node with its own first copy as well as any first copies in its span (and vice versa, to obtain a symmetric relation).

$$
\begin{aligned}
\varphi^{2,1}_{\circ}(x,y) &\stackrel{\text{def}}{=} \mathtt{spanfirst}(x) \wedge\\
&\phantom{\stackrel{\text{def}}{=}} (x = y \vee \mathtt{span}(x,y))\\
\varphi^{1,2}_{\circ}(x,y) &\stackrel{\text{def}}{=} \mathtt{spanfirst}(y) \wedge\\
&\phantom{\stackrel{\text{def}}{=}} (x = y \vee \mathtt{span}(y,x))\\
\varphi^{3,1}_{\circ}(x,y) &\stackrel{\text{def}}{=} (\text{F}(x) \vee \text{R}(x)) \wedge\\
&\phantom{\stackrel{\text{def}}{=}} (x = y \vee \mathtt{span}(x,y))\\
\varphi^{1,3}_{\circ}(x,y) &\stackrel{\text{def}}{=} (\text{F}(y) \vee \text{R}(y)) \wedge\\
&\phantom{\stackrel{\text{def}}{=}} (x = y \vee \mathtt{span}(y,x))
\end{aligned}
$$

We set $\varphi^{m,n}_{\circ}(x,y)$ for all other values of m and n to false. This obtains the association relations as depicted in Fig. 10.

As the reader can confirm, we have thus obtained the relationship between strings and ARs as originally examplified in Table 4.

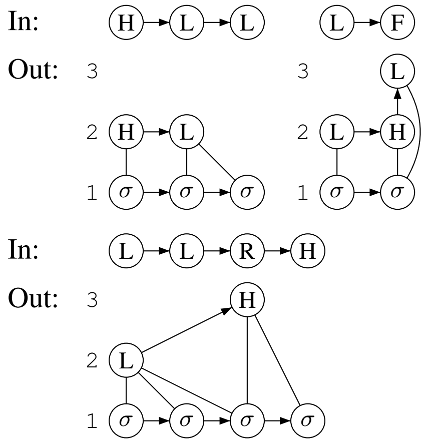


Figure 10: Creating the association relation in autosegmental representations.

Theorem 2 *The above FO transduction maps every string over* $\{\mathrm{H}, \mathrm{L}, \mathrm{R}, \mathrm{F}\}^*$ *to an AR following WFCs (a), (b), and (d).*

Proof: (Sketch.) WFCs (a) and (b) in Def. 2 stipulate that every syllable is associated to a tone and vice versa. Note that for any position x in the input string, either $\mathtt{spanfirst}(x)$ will be true of this position or $\mathtt{span}(y, x)$ is true of this position in some y. The formulas defining the $\circ$ relation ensures all such pairs are associated and each $\mathtt{spanfirst}(x)$ is associated with its copy on the melody tier, and that this relation is symmetric. For WFC (d), that the definitions depend on both $\mathtt{spanfirst}(x)$ and $\mathtt{span}(x, y)$ means that no association line will 'cross' into a new span. □

5.2 Discussion

We have thus demonstrated a set of ARs that are FO-definable from strings representing sequences of toned syllables. Because this transduction is defined as an *interpretation* of the input structure, the relations in the AR model are equivalent to FO-statements in the string model. For example, the atomic formula $x \triangleleft' y$ in the AR model is true when either of the formulas $\varphi^{n,m}_{\triangleleft'}$ is true. In other words,

$$x \triangleleft' y \equiv \varphi^{1,1}_{\triangleleft'}(x,y) \vee \varphi^{2,2}_{\triangleleft'}(x,y) \vee \varphi^{2,3}_{\triangleleft'}(x,y) \vee \varphi^{3,2}_{\triangleleft'}(x,y) \vee \varphi^{3,3}_{\triangleleft'}(x,y).$$

The same is true for the other atomic formulas $x \circ y$, $\mathrm{H}(x)$, $\mathrm{L}(x)$, and $\sigma(x)$ in the FO logic of the AR model. This means that any FO formula in the logic of the AR model can be translated into the FO logic of the string model. Thus, FO over these ARs is equivalent to FO in the string model.

One caveat is that in the definition for the AR successor order, $\varphi^{2,2}_{\triangleleft'}(x,y)$, $\varphi^{2,3}_{\triangleleft'}(x,y)$, $\varphi^{3,2}_{\triangleleft'}(x,y)$, and $\varphi^{3,3}_{\triangleleft'}(x,y)$ all used the string precedence predicate $x \prec y$, either directly in the definition or through the use of the predicate $\mathtt{span}(x, y)$. While concerns for space preclude a full proof, it is easy to see that these same predicates could not be defined using the string successor $x \triangleleft y$ and still account for spans of arbitrary length. This means that including the precedence relation $\prec$ in the string model is crucial for the definition of the AR successor $\triangleleft'$ (note that $x \triangleleft y$ is FO-definable from $x \prec y$ but the reverse is not true). As mentioned above, this means that successor in the AR, specifically successor on the melody tier, corresponds to precedence in the string model.

To summarize, this section has introduced a method for defining ARs in FO from strings representing sequences of toned syllables. Thus, FO statements over ARs are no more powerful than FO statements over strings (with $\prec$). Note again that this definition is categorically different from the tone-mapping transformation discussed in the previous section, which was shown to not be MSO-definable.

6 Conclusion

This paper has presented two new results, one negative and one positive, regarding complexity and autosegmental representations in phonology. The first result is that tone mapping transformations assigning units on one tier to units on another tier in a one-to-one fashion are not MSO-definable. This is in sharp contrast to other phonological patterns, which have been shown to be at least MSO definable and, in most cases, FO-definable. The second, positive, result is that ARs *are* FO-definable from strings, showing that they do not significantly increase the expressive power of phonotactic grammars. It is thus also likely that they do not significantly increase the expressive power of string mappings, although the logical study of phonological transformations is still ongoing (see, e.g., Heinz, forthcoming). This work thus represents one of many steps towards an understanding of phonological computation and representation.

Acknowledgments

The author thanks Jeffrey Heinz, Thomas Graf, Jane Chandlee, and three anonymous reviewers for their thoughtful comments and discussion.

References

Diana Archangeli and Douglas Pulleyblank. 1994. *Grounded Phonology*. Cambridge: MIT Press.

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

Steven Bird and E. Klein. 1990. Phonological events. *Journal of Linguistics* 26:33–56.

J. Richard Büchi. 1960. Weak second-order arithmetic and finite automata. *Zeitschrift für Mathematische Logik und Grundlagen der Mathmatik* 6:66—92.

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

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

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

George N. Clements and Kevin C. Ford. 1979. Kikuyu tone shift and its synchronic consequences. *Linguistic Inquiry* 10:179–210.

John Coleman and John Local. 1991. The "No Crossing Constraint" in autosegmental phonology. *Linguistics and Philosophy* 14:295–338.

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

David Dwyer. 1978. What sort of tone language is Mende? *Studies in African Linguistics* 9:167–208.

C. C. Elgot. 1961. Decision problems of finite automata design and related arithmetics. *Transactions of the American Mathematical Society* 98(1):21–51.

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.

Emmanual Filiot. 2015. Logic-automata connections for transformations. In *Logic and Its Applications (ICLA)*, Springer, pages 30–57.

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

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

Jeffrey Heinz. 2007. *The Inductive Learning of Phonotactic Patterns*. Ph.D. thesis, UCLA.

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

Jeffrey Heinz. 2010. Learning long-distance phonotactics. *LI* 41:623–661.

Jeffrey Heinz, editor. forthcoming. *Doing Computational Phonology*. Oxford: Oxford University Press.

Jeffrey Heinz and William Idsardi. 2013. What complexity differences reveal about domains in language. *Topics in Cognitive Science* 5(1):111–131.

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.

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

Mark Hewitt and Alan Prince. 1989. OCP, locality, and linking: the N. Karanga verb. In E. Jane Fee and Katherine Hunt, editors, *WCCFL 8*. Stanford: CSLI Publications, pages 176–191.

Wilfred Hodges. 1997. *A Shorter Model Theory*. Cambridge: Cambridge University Press.

John Hopcroft, Rajeev Motwani, and Jeffrey Ullman. 2006. *Introduction to Automata Theory, Languages, and Computation*. Addison-Wesley, third edition.

Larry Hyman. 1987. Prosodic domains in Kukuya. *NLLT* 5:311–333.

Adam Jardine. 2014. Logic and the generative power of Autosegmental Phonology. In John Kingston, Claire Moore-Cantwell, Joe Pater, and Robert Staubs, editors, *Supplemental proceedings of the 2013 Meeting on Phonology (UMass Amherst)*. LSA, Proceedings of the Annual Meetings on Phonology.

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. to appear. The local nature of tone association patterns. *Phonology* .

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

C. Douglas Johnson. 1972. *Formal aspects of phonological description*. Mouton.

Ronald Kaplan and Martin Kay. 1994. Regular models of phonological rule systems. *Computational Linguistics* 20:331–78.

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

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

Haruo Kubozono. 2012. Varieties of pitch accent systems in Japanese. *Lingua* 122:1395–1414.

Regine Lai. 2015. Learnable versus unlearnable harmony patterns. *LI* 46:425–451.

W. R. Leben. 1973. *Suprasegmental phonology*. Ph.D. thesis, Massachussets Institute of Technology.

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

Kevin McMullin and Gunnar Ólafur Hansson. 2015. Locality in long-distance phonotactics: evidence for modular learning. In *NELS 44, Amherst, MA*. In press.

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

Paul Newman. 1986. Tone and affixation in Hausa. *Studies in African Linguistics* 17(3).

Paul Newman. 2000. *The Hausa Language: An encyclopedic reference grammar*. New Haven: Yale University Press.

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

Amanda Payne. 2014. Dissimilation as a subsequential process. In *Proceedings of the 44th Meeting of the North East Linguistic Society (NELS)*.

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

Douglas Pulleyblank. 1986. *Tone in Lexical Phonology*. Dordrecht: D. Reidel.

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*, Springer Berlin Heidelberg, Berlin, Heidelberg, pages 366–385.

James Rogers, Jeffrey Heinz, Margaret Fero, Jeremy Hurst, Dakotah Lambert, and Sean Wibel. 2013. Cognitive and sub-regular complexity. In *Formal Grammar*, Springer, volume 8036 of *Lecture Notes in Computer Science*, pages 90–108.

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

Stephanie Shih and Sharon Inkelas. 2015. Autosegmental aims in surface optimizing phonology. Ms., available at lingbuzz/002520.

James Neil Sneddon, K. Alexnader Adelaar, Dwi N. Djenar, and Michael Ewing. 2010. *Indonesian: A comprehensive grammar*. New York: Routledge.

Boris Avraamovich Trakhtenbrot. 1961. Finite automata and logic of monadic predicates. *Doklady Akademii Nauk SSSR* 140:326–329.

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.

Anssi Yli-Jyrä. 2013. On finite-state tonology with autosegmental representations. In *Proceedings of the 11th International Conference on Finite State Methods and Natural Language Processing*. Association for Computational Linguistics, pages 90–98.

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