

# Learning Tonotactic Patterns over Autosegmental Representations

Han Li

January 10, 2025

## Abstract

Autosegmental theory (Goldsmith, 1976) proposes that tone should be represented in a tier separate from segmental tier, and both tiers are linked through associations governed by wellformedness conditions. Tonotactic patterns can be easily expressed over autosegmental representations (ARs). This paper is the first to show how tonotactic patterns can be learned over ARs. It uses the bottom-up learning algorithm proposed by Chandlee et al. (2019), which captures the most general constraints that adequately cover a given finite set of positive data with model-theoretic representations. An experimental case study was conducted using tonal data from Hausa, a Chadic language mainly spoken in West Africa. The experiment revealed 26 distinct ARs among 664 surface-true Hausa words, and identified 7 AR structures (syllable number  $\leq 3$ , tone number  $\leq 3$ ) missing from the data, which constitutes the constraints in the tonal grammar. The computationally discovered constraints either match ones previously proposed or better provide more explicit contexts. Additionally, patterns not previously reported were also identified by the algorithm. The paper demonstrates the feasibility and efficiency of ARs as a possible representation for grammar inference.

## 1 Introduction

Tonal phonology differs from segmental phonology due to the nonlinear nature of tonotactic patterns. For wellformedness of tonotactic patterns, tonal units (such as H and L) and their associations with timing units need to be taken into consideration. This characteristic is captured in Autosegmental Theory (Goldsmith, 1976), which proposes that tone should be represented on a tier separate from the segmental tier, and both tiers are linked through associations regulated by a set of wellformedness conditions. Subsequently, autosegmental theory has demonstrated distinct advantages in representing prevalent tonotactic patterns (Hyman, 2011), for which linear theories struggle to formalize (Odden, 1994).

Despite the importance of ARs in tonal phonology, there is little work on using them in learning. This paper offers the first attempt at learning tonotactic patterns over ARs instead of string representations. Distinct ARs are represented by model-theoretic representations which can fully capture the unique internal AR structures. The paper focuses on three major questions concerning tonotactic learning: why tonotactic patterns are preferably learned with ARs; whether any algorithm can learn the tonotactic grammar of the language, and what tonotactic patterns can be found by the algorithm and what they reveal to us. Using a case study from Hausa, I show that a bottom-up learning algorithm (Chandlee et al., 2019) successfully

captures the tonotactic constraints over autosegmental representations in the language. Further analysis reveals that among these constraints, not only are there those previously attested in linguistic analysis but also there are those never reported before which have been discovered.

To be more specific, consider one commonly discussed example of tonal mapping from the HL and LH melody in Hausa and Mende (Jardine, 2017; Zoll, 2003). The two languages form a “mirror image” of each other. They first differ in their contour formation: in Mende, rising tones are permitted on monosyllabic words but they are forbidden in Hausa. Secondly, when it comes to the 2-tone melody mapping onto a trisyllabic word, Mende and Hausa have distinct surface forms: a HL becomes HLL in Mende trisyllabic words but HHL in Hausa, and LH becomes LLH in Mende but LHH in Hausa.<sup>1</sup> Furthermore, monosyllabic  $\widehat{\text{LH}}$  are acceptable in Mende yet forbidden in Hausa. These patterns can be represented easily with ARs as shown in Table 1.

- (1) Mende  
 HL mbû ( $\widehat{\text{HL}}$ ) pèle (HL) fèlàrà (HLL)  
 LH mbă ( $\widehat{\text{LH}}$ ) níká (LH) lèlèrà (LLH)
- (2) Hausa  
 bût ( $\widehat{\text{HL}}$ ) wàtà:(HL) láhádì (HHL)  
 – ìná (LH) àshá:ná: (LHH)

Wellformed language/pattern	LH	HL	contours
Mende	$\begin{array}{c} \text{L} \quad \text{H} \\   \quad   \\ \sigma \quad \sigma \end{array}$	$\begin{array}{c} \text{H} \quad \text{L} \\   \quad   \\ \sigma \quad \sigma \end{array}$	$\begin{array}{c} \text{H} \quad \text{L} \quad \text{L} \quad \text{H} \quad \text{L} \quad \text{H} \quad \text{L} \\ \diagdown \quad \diagup \quad \diagdown \quad \diagup \quad \diagdown \quad \diagup \quad \diagdown \quad \diagup \\ \sigma \quad \sigma \quad \sigma \end{array}$
Hausa	$\begin{array}{c} \text{L} \quad \text{H} \\   \quad   \\ \sigma \quad \sigma \end{array}$	$\begin{array}{c} \text{H} \quad \text{L} \\   \quad   \\ \sigma \quad \sigma \end{array}$	$\begin{array}{c} \text{H} \quad \text{L} \\ \diagdown \quad \diagup \\ \sigma \end{array}$

Table 1: 2-tone trisyllabic forms and contours in Hausa and Mende

Jardine (2017) explains how these tonotactic patterns over ARs can be expressed as graphs (see also Coleman and Local (1991); Jardine and Heinz (2015)). First, there is a set of labeled nodes representing elements from the two tiers of information in ARs: one from the tonal tier and the other from the TBU tier. On each tier, the fixed order of the elements is depicted with directed edges. Between two tiers, undirected edges link elements corresponding to association lines in ARs. As shown in Figure 1, labeled nodes are represented by circles with labels inside them; directed edges are arrows with heads; undirected edges are represented with lines between two tiers. One can see each AR and its graph is identical in terms of their tonal sequence, TBU numbers, and association lines. Hence, we can define a wellformed AR as one does not contain any graph or connected pieces of a graph (known as subfactors) that is considered a constraint in the grammar. For example, in Hausa, the absence of a monosyllabic rising tone can be understood as the language forbidding the structure where two tonal nodes, a L followed by a H, connect to the same

<sup>1</sup>In this paper, when two identical tones occur adjacent to each other, I will follow the tonal transcription method commonly used by Africanists, where “HH” represents two adjacent syllables both having a high tone (Leben, 1971; Newman, 2002; Zoll, 2003). This shorthand notation does not necessarily indicate a violation of the Obligatory Contour Principle in Autosegmental Theory (Goldsmith, 1976). The same rationale applies to the “HHL” (two H-toned syllables followed by a L-toned syllable), “LHH” (a L-toned syllable followed by two H-toned syllables) and similar notations used elsewhere. Concerning ARs, Jardine and Heinz (2015) showed how to convert such representations into ARs that obey the OCP.

TBU node on the timing tier as in Figure 1a. If any AR contains (1a), for instance as (1b) or (1c) does, it will be identified illicit as well.

More formally, if Grammar  $G$  is the set of constraints in the language, such that  $G = \{r_1, r_2, \dots, r_n\}$ , a wellformed structure is one that contains no forbidden subfactors, or equivalently, the one satisfies the conjunction of negative literals:  $\neg r_1 \wedge \neg r_2 \wedge \neg r_3 \wedge \dots \wedge \neg r_n$  (Jardine, 2017; Rogers and Lambert, 2019).

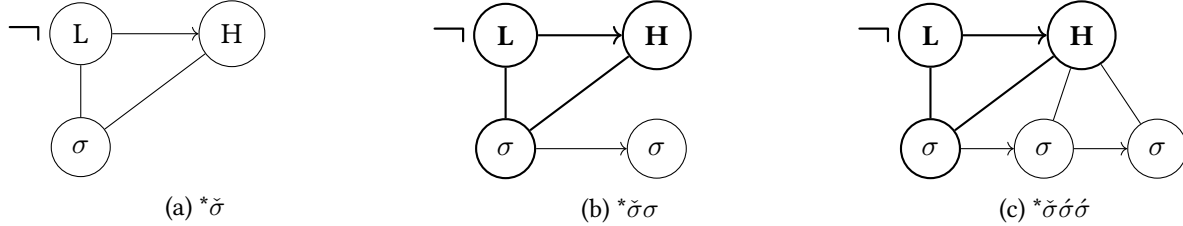


Figure 1: A forbidden structure  $*LH$

Naturally, the next major question would be how these constraints can be learned computationally. In other words, is it possible to learn the set of constraints from example words, and thus capture the Grammar of the language? Chandlee et al. (2019) introduce a bottom-up learning algorithm which is capable of discovering the most general forbidden subfactors represented in model-theoretic forms in a language, given a finite set of examples (known as positive data). I tested this algorithm using tonotactic data from Hausa, a Chadic language spoken in West Africa, as a case study. The data were represented over ARs using a model-theoretic approach. The aim was to compare the matches between computationally detected constraints and linguistically attested constraints. It is hoped that this will not only provide more insights into tonotactic patterns under computational contexts but also show the efficacy of ARs for tonal language data.

This paper is structured as follows. In Section 2, I will introduce key concepts in autosegmental theory and emphasize the advantages of representing tones over ARs rather than linear phonology. Section 3 will present the preliminaries in formal language and model theory. Building on the foundations laid in Section 3, Section 4 will provide a detailed description of the algorithm implementation over AR models. In Section 5, I will introduce the Hausa data used in the case study and explain how string are converted into ARs to fit algorithm implementation. Section 6 will discuss the results of the experiment, followed by the conclusion and future questions in Section 7.

## 2 Autosegmental and String Representation

One important question before moving onto any learning problems is that why should autosegmental representations be favored over string representations or features in tonal phonology. In this section, I will review arguments supporting that ARs offer advantages over linear theories for tonal phonology. Three commonly discussed tonal phenomenon (reviewed in details along with more arguments in Odden (2013)) are relevant to tonotactic patterns as well as computational grammar inference:

1. Contour formation: complex tones emerge as a result of readjusting tonal units with timing units in autosegmental theory rather than changing feature values;
2. Melodic tonal restriction: languages have limited number of tonal sequences which are independent from their segmental content;
3. Non-local or “across-the-board” patterns: one tone unit can operate over multiple segments;

These arguments support AR as a better and more explicit representation of tones.

## 2.1 Complex tones over ARs

First, contours typically form when two contrastive tones occupy the same tone-bearing unit, resulting in a complex tone formed in the original sequential order. With linear representations, tones are either represented with tone labels (H, L, M, F, R) or with features (H as [+H], L as [-H], R as [-H, +contour] and F as [+H, +contour]), which are treated as properties of the segments. Derivational contours are typically formalized as changes in labels or feature values as in (3) and (4), where contour tones [+contour] emerge as a result of two opposing tone values are adjacent.

$$(3) \quad \begin{array}{l} H \rightarrow F / \_L \\ L \rightarrow R / \_H \end{array}$$

$$(4) \quad \begin{array}{l} [+High] \rightarrow [+contour] / \_[-High] \\ [-High] \rightarrow [+contour] / \_[+High] \end{array}$$

However, this formalization approach leads to undergeneralization and overgeneralization. The feature system struggles to generalize processes when H becomes R before L or F; or when L becomes F before H or R. In these two cases, natural classes are needed to incorporate {L, F} and {H, R}, which is impossible since no set of feature-values picks out the two tones in each group. On the other hand, within the feature system, patterns such as those in (6) are predicted, but they are unattested in any tonal languages discovered so far.

$$(5) \quad \begin{array}{l} \text{Undergeneralization} \\ ?[+H] \rightarrow [+contour] / \_ \{[-H], [+H, +contour]\} \\ ?[-H] \rightarrow [+contour] / \_ \{[+H], [-H, +contour]\} \end{array}$$

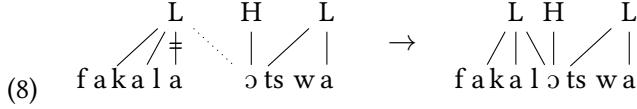
$$(6) \quad \begin{array}{l} \text{Overgeneralization} \\ *[+H] \rightarrow [+contour] / [-H]\_ \\ *[-H] \rightarrow [+contour] / [+H]\_ \end{array}$$

Furthermore, the linear theory shows more difficulty with convex (FR) and concave (RF) tones, which only increases the tonal inventory. Mongo (Odden, 1994) has the three-tone contours FR and RF. If these tones are viewed as inseparable and inherent tonal types in the inventory, the inventory will increase from 2 (H and L) to 6 (H, L, F, R, FR, RF). If so, one could expect 36 combinations for a disyllabic word, and 216 for a trisyllabic words.

Autosegmental phonology, on the other hand, views tonal processes not as arbitrary changes in feature values, but rather as adjustments or re-associations of tones and timing units. From this perspective, contour tones emerge as a result of two tones associating with a single tone-bearing unit. This significantly reduces the necessity for a large inventory of units which integrate tones with segmental content, and provides a more straightforward analysis of complex tone preservation processes, even as observed in

- (7) Mongo
- |               |   |             |            |
|---------------|---|-------------|------------|
| bàlóngá bákáé | → | bàlóngǎ káé | H + R → FR |
| fàkàlà ôtswà  | → | fàkàlô tswà | L + F → RF |
| bǎnkò bǎmǒ    | → | bǎnkǎmǒ     | L + R → R  |
| ǒmǒ èmbè      | → | ǒmǐ mbè     | R + F → RF |

Mongo. In (8), no new tonal types are introduced into the inventory nor is there any change in the melody (from L-HL to LHL). Instead, there are two changes: the deletion of vowel *a* in the segment tier and the dissociation of *a* from L, which then spreads onto the following segment. As a result, the vowel *ɔ* is associated to 3 tones LHL which corresponding to RF in Ex (8).



## 2.2 Melodic Tonal Restriction

Many languages have been found to have a fixed set of melodies that “must be abstracted away from the the TBUs on which they are realized” (Hyman, 2011).<sup>2</sup> In other words, there is a limited number of tonal combinations over phrases or words that a language can have. Mende is an often-cited example of this melodic tonal restriction. The language has 2 contrast tones: H and L while HL combines to F, LH to R, and LHL to RF. A trisyllabic word could therefore have  $5 * 5 * 5 = 125$  possible tonal combinations. However, irrespective of the syllable number, Mende only have five tonal melodies: H, L, HL, LH, and LHL. (Leben, 2017; Odden, 2013; Hyman, 2011).

No.	Melody Pattern	Monosyllabic	Disyllabic	Trisyllabic
1.	/H/	kó	kó-má	hávámá
2.	/L/	kpà	bèlè	bèlè-mà
3.	/HL/	mbû	mbú-mà	félàmà
4.	/LH/	mbǎ	mbà-má	ndàvúlá
5.	/LHL/	mbǎ	nyàhâ	nyàhâ-mà

Table 2: Mende Tone Patterns (Odden, 2013, p.294)

The same case also applies to Kukuya, where Paulian (1975, cited from Hyman (2011)) recognized same five tonal melodies that can be predicted to map onto stems of different lengths. Besides, in Kpelle, the fix tonal meldoies are /H,L,HL,M, MHL/.<sup>3</sup> These examples indicate that if tones are viewed as a segmental

<sup>2</sup>There are languages where tonal melodies are not highly restricted such as Cantonese and Mandarin Chinese. They can be viewed as “syllable-tone” languages where tones do not changes when morphemes are combined together (Yip, 2002). Even for these languages, Wang (1967) pointed out the domain of tones applies over the entire voiced portion of syllables and the tone quality is irrelevant to the nature of the segment bearing the tone, and thus tone features need to be differentiated from segmental features.

<sup>3</sup>Hyman (2011, p.492) mentioned the M melody has an underlying form of /LH/ which result in Kpelle having the same tonal patterns as Mende.

property, the constraints accounting for them will be more complex. Instead, the restrictions can be explained by stating that there are only five patterns of H and L combinations on the tonal tier of an AR, but they are linked to different lengths of TBUs.

### 2.3 Non-local tonal patterns

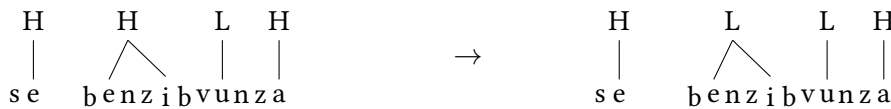
It has also been argued that ARs are favored in tonal phonology because they easily describe non-local tonal patterns which are very common among tonal languages (Odden, 2013). By non-local pattern it refers to the cases where a tone triggers a change that does not exclusively apply to its local neighbors but all elements in the target domain. One example is the Shona noun compound (Odden, 1994). If the initial tone of the noun stem (leftmost column in Example 9) is a H tone, the prefix *né*, *sé*, *ché* will trigger a series of L tone to change into H (9b and 9c) until another L tone is present to block the spreading (9d).

	N	with N	like N	of N	gloss of N
	a. mùrúmé	né-mùrúmé	sé-mùrúmé	ché-mùrúmé	“man”
(9)	b. mbùndùdzí	né-mbùndùdzì	sé-mbùndùdzì	ché-mbùndùdzì	“army worms”
	c. hákàtà	né-hákàtà	sé-hákàtà	ché-hákàtà	“bones”
	d. bènzìbvùnzá	né-bènzìbvùnzá	sé-bènzìbvùnzá	ché-bènzìbvùnzá	“fool”

A rule could be formalized to account for this change:  $[+H] \rightarrow [-H] / \{sé, né, shé\}_-$ . This will cause the wrong surface forms such as \***né-mbùndùdzí**, \***né-hákàtà** and \***né-bènzìbvùnzá** due to the fact that the rule will only apply to the first element right next to the H-toned prefix. To accurately capture the rule, a formula needs to capture a long-distance dependency that will apply to every element within the target domain as shown in (10).

- (10)  $H \rightarrow L / H\_ (L)$   
 $HH \rightarrow LL / H\_ (L)$   
 $\dots$   
 $H^n \rightarrow L^n / H\_ (L)$

The autosegmental analysis provides a more straightforward explanation, as it accounts for the lowering phenomenon exclusively affecting the initial H tone, which may be associated with multiple syllables simultaneously.



This example demonstrates two properties that are very common in tonal phonology but less common in segmental phonology. The first one is tonal polarity, which refers to the melodic alternation between H and L tones (Cahill, 2007; Hyman and Schuh, 1974; Odden, 1994). The second property, unboundedness, refers to the fact that the number of tone elements subject to change is unlimited as long as the contextual environment is satisfied (Jardine, 2016).

### 3 Preliminaries

#### 3.1 Strings and their Representations

Informally, a string can be understood as a sequence of characters. Let  $\Sigma$  be a finite alphabet of symbols and  $\Sigma^*$  the set of all strings over  $\Sigma$ .  $|w|$  indicates the length of  $w \in \Sigma^*$ . For two strings  $w$  and  $v$ ,  $wv$  refers to their concatenation of two strings, and for a set  $L \subseteq \Sigma^*$  of strings and a string  $w$ , by  $wL$  we denote  $\{wv \mid v \in L\}$ . A string  $u$  is a *substring* (or a *factor*) of string  $v$  if there are strings  $x, y \in \Sigma^*$  such that  $u = xvy$ . If  $u$  is of size  $k$  and is a factor of  $v$ , then  $u$  is a  $k$ -factor of  $v$ .

A string can be represented model-theoretically. Model theory provides a uniform framework for representing objects and their relations to one another. A model-theoretic representation for strings contains two parts: the Domain  $D$  and a finite set of  $n$ -ary relations  $\mathcal{R} = \{R_1, R_2, \dots, R_i\}$  (Heinz, forthcoming). The set of relations comprise the *signature* of the model. Distinct objects will have distinct model signatures to represent the unique internal structures. The signature  $\{R_1, R_2, \dots, R_i\}$  includes different types of relations: unary relations represent symbols in the string, and binary relations indicate either a successor relation (denoted by  $\triangleleft$ ) and a precedence relation (denoted by  $\prec$ ) to maintain the linear order of elements. In a simple example string of strings formed with from the alphabet  $\{a, b\}$ , the representation of  $abaa$  would be  $\mathcal{M} = \{D \mid R_a, R_b, \triangleleft\}$  where  $D = \{1, 2, 3, 4\}$  indicating there are 4 elements in the string;  $R_a$  is a unary relation for the symbol  $a$  such that  $R_a = \{1, 3, 4\}$ ;  $R_b$  is a unary relation for the symbol  $b$  such that  $R_b = \{2\}$ , and the binary successor relation  $\triangleleft = \{\langle 1, 2 \rangle, \langle 2, 3 \rangle, \langle 3, 4 \rangle\}$  indicates the order of elements. A diagram for the model  $abaa$  and its signature is illustrated in Figure 2.

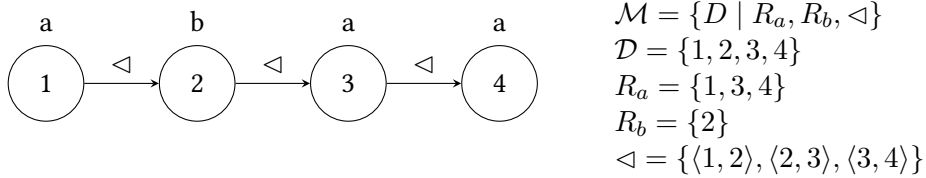


Figure 2: String model and its signature for  $abaa$

#### 3.2 ARs as Unconventional Models

A string model can be viewed as a one-dimensional model since elements are ordered linearly on one axis. String models with successor and precedence relations have been called “conventional” models (Chandlee et al., 2019; Strother-Garcia et al., 2017; Vu et al., 2018). Being conventional means that each element has a single and exclusive label. For example, in Figure 2, the node  $\{2\}$  is labeled  $b$  and not anything else. Unlike the string model, ARs are unconventional models consisting of two strings: a tonal string and a timing string. Elements on each string are first in a successor relation and also bidirectionally connected to elements from the other string. We can refer to an AR as a bistring or a two-dimensional string. Its model signature can be constructed by initially building two string models and then combining them with association relations.

To construct the tonal string of an AR model needs a sequence of H and L alternating with each other. We could construct a successor string model with only tonal sequences as shown in Figure 3. In this model, the domain  $D_t$  matches the number of tone units; two unary relations  $R_H$  and  $R_L$  refer to the H tone and L tone, respectively. Additionally, a binary relation  $\triangleleft_t$  represents a linear successor relation.

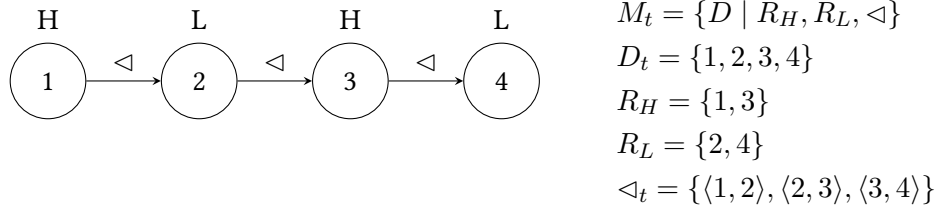


Figure 3: Tonal string model *HLHL* and its signature

Likewise, a string model for the timing string as in Figure 4 can be constructed in the same fashion. The domain  $D_s$  matches the number of TBUs (syllables or moras). We will use  $R_\sigma$  to refer to TBU unary relation, which is the only element on the string. Then, TBUs are connected in a linear order.

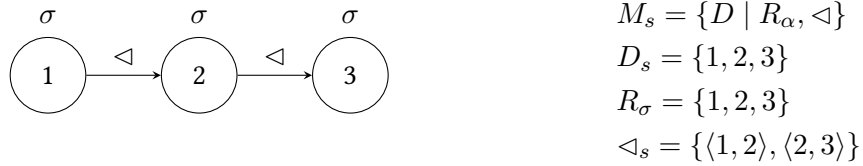


Figure 4: String model and its signature for the tonal sequence *HLHL*

Finally, elements from one string connect with those on the other one. Linguistically, this corresponds to the association lines in AR that indicate the arrangement of H and L with respect to the temporal order. To connect the two strings and build an AR model, a binary relation  $\alpha$  is used to represent the associations held between two strings. The signature of the combined model is the union of the signatures of the tonal string and timing string, along with the binary association relation. Specifically, the size of  $D$  is the sum of tonal units and TBUs such that  $D = \{1, 2, \dots, |D_t| + |D_s|\}$ . If we renumber each distinct node, for example, with the first syllable being node 1 and the first tone node 4, we will have a model as in Figure 5.

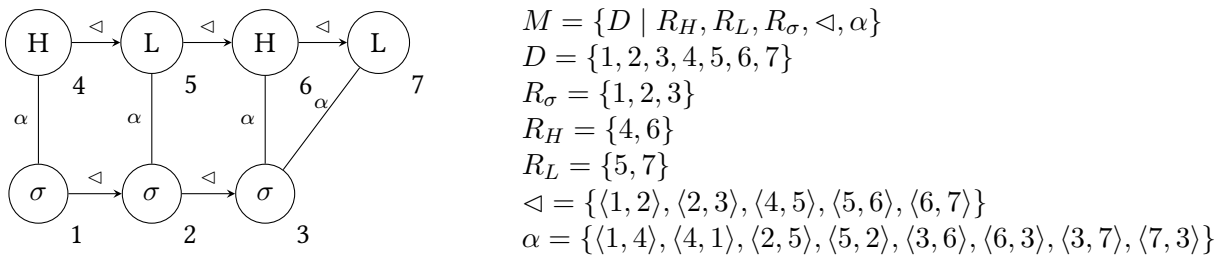
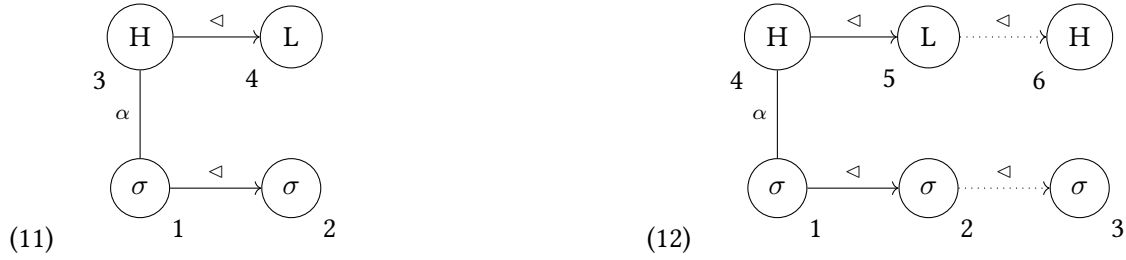


Figure 5: AR model for a word  $\sigma\sigma\sigma$



For the successor function, there are few points to mention. First, the elements in a successor relations are directed nodes, indicating a left-to-right temporal order. Secondly, the tonal string in successor relation indicates the alternation of H and L tone. No two identical tone units are allowed to be adjacent. The Obligatory Contour Principle forbids strings such as \*HHL and \*LLH. To expand a given AR from a successor relation, such as from (11) to (12), one can either add a tone unit which is opposite to the preceding tone on the tone tier, or add one TBU on the timing tier.



As for the association relation, the nodes are undirected, indicating that the binary relations are doubly connected. A list of association tuples of  $\langle 1, 4 \rangle, \langle 2, 5 \rangle, \langle 3, 5 \rangle$  is equivalent to  $\langle 4, 1 \rangle, \langle 5, 2 \rangle, \langle 5, 3 \rangle$ . Besides, the structure allows the existence of unlinked units. For example, in (11) the L tone labeled 4 is not linked to any element in the TBU tier. In the model signature, the association relation  $\alpha = \{ \langle 1, 3 \rangle \}$  and does not include any of  $\{ \langle 1, 4 \rangle, \langle 2, 4 \rangle, \langle 4, 1 \rangle, \langle 2, 4 \rangle \}$ . We refer the L at 4 and  $\sigma$  at 2 as *floating* units. Section 4 will give more formal definition for AR expanding and floating units.

## 4 Learning Autosegmental Representations

Following [Jardine \(2017\)](#), the previous section argues that ARs can be represented in an unconventional model consisting of two strings: one on the tonal tier and the other on the syllable tier. Elements of the strings on both tiers are connected by a successor relation, while an association relation holds between the two tiers. In this section, I'll introduce a bottom-up learning algorithm proposed by [Chandlee et al. \(2019\)](#) which was designed to learn constraints in a space of partially ordered structures. This algorithm can learn tonotactic constraints over ARs since the containment relation of ARs is also partially ordered in language.

### 4.1 A Bottom-Up Learning Algorithm

[Chandlee et al. \(2019\)](#) discuss two directions of grammatical inference, a top-down and a bottom-up direction, the latter one is more efficient and guaranteed to find the constraints in a language. In the bottom-up learning fashion, the learner starts with the most generalized structures and move upwards to the more specific ones. Once a structure is identified missing from the positive data, the learning path moving upwards is thus discontinued and the learner will first recognize the current structure as a constraint in the language and will no longer consider its superfactors. One advantage of the algorithm is it guarantees to find the most general grammar which covers the given data. A bottom-up learning algorithm (BUFIA) is shown in Algorithm 1.

---

**Algorithm 1** Bottom-Up Learning Algorithm

---

**Require:** positive data  $D$ , empty structure  $s_0$ , max constraint size  $k$

```
 $Q \leftarrow \{s_0\}$ 
 $G \leftarrow \emptyset$ 
 $V \leftarrow \emptyset$ 
while  $Q \neq \emptyset$  do
   $s \leftarrow Q.\text{DEQUEUE}()$ 
   $V \leftarrow V \cup \{s\}$ 
  if  $\exists x \in D$  such that  $s \sqsubseteq x$  then
     $S \leftarrow \text{NextSupFact}(s)$ 
     $S' \leftarrow \{s' \in S \mid |s'| \leq k \wedge (\neg \exists g \in G)[g \sqsubseteq s'] \wedge s' \notin V\}$ 
     $Q.\text{ENQUEUE}(S')$ 
  else
     $G \leftarrow G \cup \{s\}$ 
  end if
end while
return  $G$ 
```

---

The algorithm initially requires three inputs: a finite size of positive data  $D$ , an empty structure  $s_0$ , and a maximum constraint size  $k$ . The positive data  $D$  contains a finite set of surface-true structures in the language. The empty structure  $s_0$  is a structure that has a length of zero. The factor  $k$  limits the size of structures considered by the algorithm. Additionally, the algorithm initializes two empty sets:  $G$ , representing the grammar which is empty at first and will be returned with constraints discovered by the algorithm once the learning is complete; and a set  $V$  which stores checked structures to prevent the learner from revisiting those structures that have been examined.

A queue  $Q$  is used to store structures awaiting to be checked by the algorithm. As shown in the Algorithm 1, there are two operations related to  $Q$ :  $Q.\text{DEQUEUE}$  and  $Q.\text{ENQUEUE}$ . The former is to take a structure  $s$  out of the queue for later checking, and the latter is to add the next structure into the  $Q$ .

Once the learning starts, BUFIA initiates from the most generalized and least complex structure, the empty structure. It then checks whether the current structure is present in the dataset. If the structure is found, the 'NextSupFact' function generates its immediate superfactors which have a size  $\leq k$ , are not visited, and contain no structures that have been already forbidden by the grammar. The superfactors will be appended to the rear of the queue for later checking. If any structure is absent from the dataset, the algorithm adds it to the Grammar  $G$  and ceases to generate its superfactors. This process repeats until the size of the structure exceeds a predetermined threshold  $k$ .

The structures added into  $G$  are those missing from the positive data, indicating that the current structure is not considered as a wellformed structure but a constraint in the language. Once a structure is considered forbidden, none of its superfactors will be checked by the learner.

## 4.2 Implementing BUFIA over Autosegmental Representations

To have BUFIA learn tonotactic patterns over ARs, we need to convert the data representation into ARs and determine the measurement of a size of an AR. Here, the data in the positive sample  $D$  are represented in ARs and three factors measure its size: the length of tonal tier  $t$ , the length of the syllable tier  $s$ , and the number of associations formed between two tiers  $|\alpha|$ . A total size  $k$  would be the sum of three factors:  $k = t + s + |\alpha|$ . Following this principle, an empty AR has a length of 0 on the tonal tier, timing tier or associations in between hence its size  $k$  is 0. The partially ordered hypothesis space of ARs is shown in Figure 6.

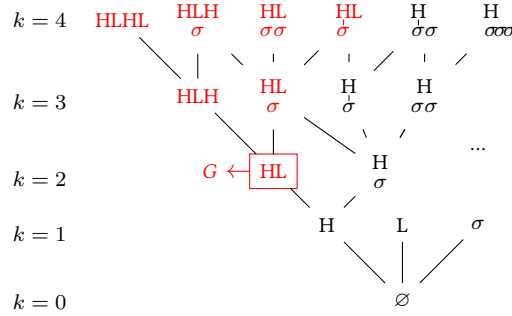


Figure 6: Bottom-Up Learning over ARs

As mentioned in Section 4.1, the algorithm starts its search from an empty structure. Two important functions are involved: the **Contain** function checks for the presence of a given structure, and the **NextSupFact** function will generate its immediate superfactors.

For each structure of size  $k$ , its immediate superfactors will have a size of  $k + 1$ , and rank one level up higher in Figure 6. The one-step increase in size results from three potential changes: addition of a tone, or a TBU, or an association between the two tiers. As can be seen from Figure 6, when  $k = 1$ , the structure can be a floating H, a floating L, or a floating syllable. Take the floating H for instance, its superfactors ( $k = 2$ ) include a floating HL sequence, and a floating H with a floating syllable. If the HL sequence is found missing from the data  $D$ , the algorithm will cease generating any superfactors and add HL into the grammar as a constraint forbidden by the language (labeled in red).

TG questioned the interpretation of “floating”. If tone-TBU association can be seen as a feature, does the floating unit has (a) unspecified association feature, or (b) a negative association feature. If it is (a), then its superfactors include: an associated  $[H-\sigma]$  and also an unassociated  $[Hx\sigma]$ . Linguistically, the unassociated one could mean a “delinking” structure

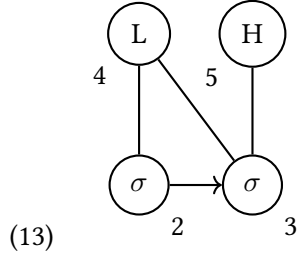
Two functions **NextSupFact** and **Contain** are essential parts the learner BUFIA infers grammars from. In the rest of the section, we will define these functions in terms of ARs.

#### 4.2.1 Contain

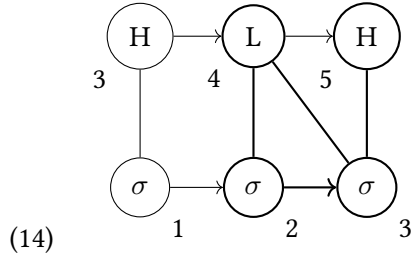
To illustrate Contain relation, I will follow the definition of *restriction* and *subfactor* given by [Chandlee et al. \(2019, p.94\)](#).

**Definition 1.**  $A = \langle D^A; \triangleleft, R_A^1, \dots, R_A^n \rangle$  is a restriction of  $B = \langle D^B; \triangleleft, R_B^1, \dots, R_B^n \rangle$  iff  $D^A \subseteq D^B$  and for each  $m$ -ary relation  $R_i$ , we have  $R_A^i = \{(x_1, \dots, x_m) \in R_B^i \mid x_1, \dots, x_m \in D^A\}$ .

Based on Definition 3, (13) is a restriction of (14) since  $D^A = \{2, 3, 4, 5\} \subseteq \{1, 2, 3, 4, 5\} = D^B$  and  $R_H^A \in R_H^B, R_L^A \in R_L^B, R_\sigma^A \in R_\sigma^B, \triangleleft^A \in \triangleleft^B, \alpha^A \in \alpha^B$ .



$$\begin{aligned} \mathcal{M}^A &= \{\mathcal{D} \mid R_H, R_L, R_\sigma, \triangleleft, \alpha\} \\ \mathcal{D}^A &= \{2, 3, 4, 5\} \\ R_\sigma^A &= \{2, 3\} \\ R_H^A &= \{5\} \\ R_L^A &= \{4\} \\ \triangleleft^A &= \{\langle 2, 3 \rangle\} \\ \alpha^A &= \{\langle 2, 4 \rangle, \langle 3, 5 \rangle, \langle 4, 2 \rangle, \langle 5, 3 \rangle\} \end{aligned}$$



$$\begin{aligned} \mathcal{M}^B &= \{\mathcal{D} \mid R_H, R_L, R_\sigma, \triangleleft, \alpha\} \\ \mathcal{D}^B &= \{1, 2, 3, 4, 5\} \\ R_\sigma^B &= \{1, 2, 3\} \\ R_H^B &= \{3, 5\} \\ R_L^B &= \{4\} \\ \triangleleft^B &= \{\langle 1, 2 \rangle, \langle 2, 3 \rangle, \langle 3, 4 \rangle, \langle 4, 5 \rangle\} \\ \alpha^B &= \{\langle 1, 3 \rangle, \langle 2, 4 \rangle, \langle 3, 5 \rangle, \langle 3, 1 \rangle, \langle 4, 2 \rangle, \langle 5, 3 \rangle\} \end{aligned}$$

With the concept of restriction, we can determine whether an AR  $A$  is contained in another one  $B$  by comparing  $A$  with a restriction of  $B$ , denotes as  $B'$ . If there exists a relation-preserving bijection between  $A$  and  $B'$ ,  $A$  is identified as a subfactor of  $B$ .

**Definition 2.** Structure  $A$  is a subfactor of structure  $B$  ( $A \sqsubseteq B$ ) if  $A$  is connected, there exists a restriction of  $B$  denoted  $B'$ , and there exists  $h : A \rightarrow B'$  such that for all  $a_1, \dots, a_m \in A$  and for all  $R_i$  in the model signature: if  $h(a_1), \dots, h(a_m) \in B'$  and  $R_i(a_1, \dots, a_m)$  holds in  $A$  then  $R_i(h(a_1), \dots, h(a_m))$  holds in  $B'$ . If  $A \sqsubseteq B$ , we also say that  $B$  is a superfactor of  $A$ .

In other words, if all relations held in  $A$  also hold in a related way in  $B$ ,  $A$  is considered as a subfactor of  $B$ , and  $B$  is considered as a superfactor of  $A$ . Figure (7) illustrates the process of determining the containment relation between (7a) and (7c). One can identify (7a) as a superfactor of (7c) because there exists a restriction of (7a) that has equivalent relations to those of (7c).

Lastly, If AR is the empty structure, it is contained by all ARs, but only contains itself.

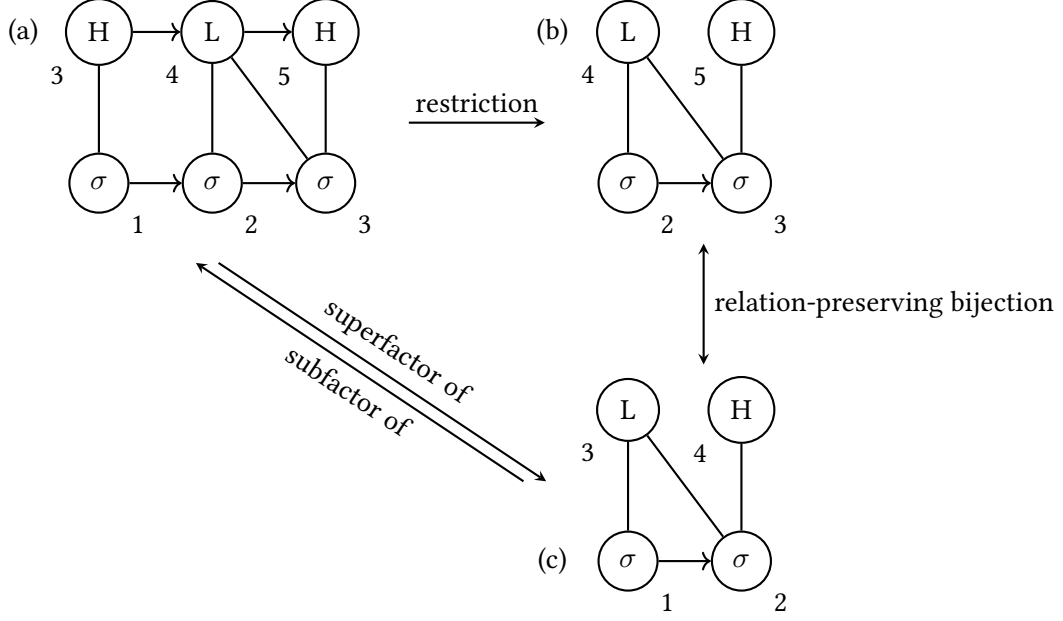


Figure 7: Superfactor, Restriction, and Subfactor AR

#### 4.2.2 Next Autosegmental Structure

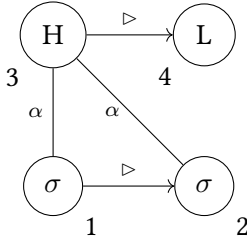
When the algorithm identifies the presence of a given structure, the **NextSupFac** function will generate its immediate superfactor. As mentioned in Section 4.2, the immediate superfactor of a structure of size  $k$  has a size of  $k + 1$ , which can result from an one-unit addition of tone number  $t$ , TBU numbers  $s$ , or the number of associations  $|\alpha|$ . In other words, an AR can be expanded using three operations: **addTone**, **addTBU**, or **addAssociation** depending on whether there are any floating units in the AR.<sup>4</sup>

$$\text{NextSupFact}(\text{AR}) = \begin{cases} \text{addTone}(\text{AR}) \cup \text{addTBU}(\text{AR}) & \text{if } \text{containFloat}(\text{AR}) = \text{False} \\ \text{addTone}(\text{AR}) \cup \text{addTBU}(\text{AR}) \cup \text{addAssociation}(\text{AR}) & \text{if } \text{containFloat}(\text{AR}) = \text{True} \end{cases}$$

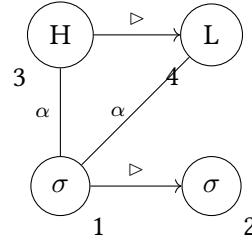
First, the function **containFloat** checks for the presence of a floating unit in the AR. For  $\text{AR} = \{D \mid R_H, R_L, R_\sigma, R_\triangleleft, R_\alpha\}$ , a unary relation  $R_i \in \{R_H, R_L, R_\sigma\}$  is considered *floating* if  $R_i \in R_\triangleleft \wedge R_i \notin R_\alpha$ . The two cases are shown in (15) and (16). AR (15) contains a floating tone L at position {4}. This element is in a successor relation of  $\{\langle 3, 4 \rangle\}$  on the tonal tier but missing from the association relation, which only contain one pair  $\{\langle 1, 2 \rangle\}$ . Similarly, AR (16) contains a floating syllable at position {2} for it is present in the successor relation  $\{\langle 1, 2 \rangle\}$  but absent from the association relation  $\{\langle 1, 3 \rangle, \langle 1, 4 \rangle\}$ .

<sup>4</sup>Theoretically, an association line can be formed between two non-floating units as long as it does not cross any other association lines, known as the No-Crossing Constraint (Goldsmith, 1976; Coleman and Local, 1991; Hammond, 1988). In the current implementation, a new association will occur when there is at least one floating unit in the structure available to form an association. Since floating units are located on the rightmost side of the AR, at least one end of a new association line is at the rightmost end of the tonal string or the TBU string. This also ensures that all constraints of size less than  $k$  will be considered.

(15) AR containing a floating tone



(16) AR containing a floating syllable

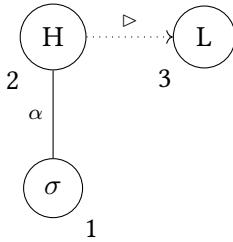


The **addTone** function is defined as follows: for any autosegmental representation  $AR$  with a size of  $k$  and a string  $w = t_1 t_2 \dots t_n, t \in \{H, L\}$  on the tonal tier, there exists one and only one  $AR'$ , which is  $AR$ 's  $k+1$  expansion on the tonal tier if  $AR'$  has a tonal string  $w' = wt_{n-1}$ . From a model-theoretic perspective, for an AR with a model signature of  $M = \{D \mid R_H, R_L, R_\sigma, R_{\triangleleft}, R_\alpha\}$ ,  $R_i$  is denoted as the final element in the tonal string,  $i \in D$ , and  $R_i \in \{R_H, R_L\}$ . Then  $AR$ 's  $k+1$  expansion on the tonal tier will only change its domain and unary relation  $D' = D \cup i+1, R_i \in R_H \Rightarrow R_{i+1} \in R_L \wedge R_i \in R_L \Rightarrow R_{i+1} \in R_H$ . In other words, a tonal string can only be expanded by adding one tone unit at a time, which is the opposite tone unit of the original final tone.

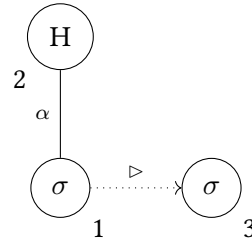
do i need  
to define  
"final"

The **addTBU** function is defined as for any autosegmental representation  $AR$  with a size of  $k$  and has a length of  $s$  on the TBU tier, there exists one and only one  $AR'$ , which is  $AR$ 's  $k+1$  expansion on the TBU tier if  $AR'$  has a TBU length of  $s+1$ . In terms of model signature,  $D' = D \cup i+1, R_{i+1} \in R_\sigma$ .

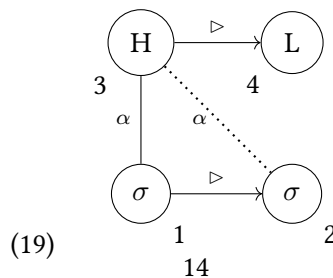
(17) Add a New tone

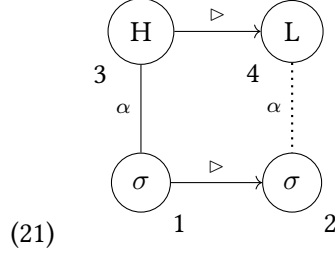
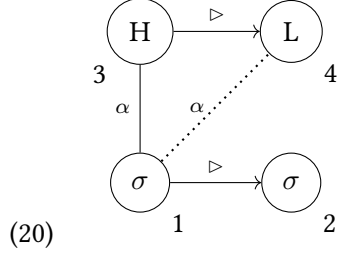


(18) Add new TBU



The **addAssociation** function will only apply to those ARs containing floating units. A new association  $(i, j)$  is only considered formed validly if and only if there does not exist an association  $(i', j') \in R_\alpha$  such that  $(i' > i \wedge j' < j) \vee (i' < i \wedge j' > j)$ . This rule is enforced in accordance to the No-Crossing Rule in Autosegmental Theory (Goldsmith, 1976; Coleman and Local, 1991; Hammond, 1988). In this case, there are 3 ways to form possible new associations: the floating tone connects to the syllable as in (19); the floating syllable connects to the tone as in (20); the floating tone connects to a floating syllable as in (21).





## 5 Case Study: Hausa

This section presents a case study of using BUFIA to identify tonotactic constraints over autosegmental representations in Hausa, a language with two contrast tones (H and L) spoken in West Africa. It demonstrates tonotactic patterns can be learned successfully over ARs given a certain length on the syllable tier  $s$  and length on the tonal tier  $t$ . Out of 664 words represented as strings extracted from the Hausa database, which include nonborrowed and unanalyzable words translated into ARs, the learning algorithm successfully identified 7 AR constraints ( $s \leq 3$  and  $t \leq 3$ ). This case study illustrates the feasibility and efficiency of learning over autosegmental representations. First, tonal learning over autosegmental representations significantly reduces the hypothesis space: of all 664 words, only 26 distinct AR patterns were found, effectively making the positive dataset smaller. Secondly, the algorithm exhibits three advantages compared to previous literature: it discovers previously identified constraints, provides more accurate and specific constraints than ones posited previously which are found to be too general, and discovers new constraints that have not been reported yet. This section first reviews the basic tonology of Hausa then introduces the three major steps of the experiment: data extraction, representation conversion, and algorithm implementation. The results will be discussed in the Section 6.<sup>5</sup>

### 5.1 Tonology of Hausa

Let us first review the basic tonology of the language in question. Hausa has a 2-tone system (H and L) and the tone-bearing unit is the syllable which consists of at most two moras. A falling contour (HL) can be formed when the two tones occupy a single syllable. There is no monosyllabic LH or any three-tone contours. According to Newman’s reference grammar of Hausa, the tonal patterns are summarized as a **right-to-left** spreading phenomenon (Newman, 2002, p.600).

“If a word has more tones than syllables, then two tones can dock on the same syllable if that syllable has the necessary weight (i.e. two moras) to carry the two tones. The tones of the melody are assigned to the syllables from right to left. If there are more syllables than tones, one keeps on spreading the tone in a leftward fashion...If there are more tones than syllables, then one assigns the tones from right to left until one has run out of syllables, whereupon one assigns the remaining tone to the initial syllable (assuming that it is a bimoraic syllable that can carry two tones).”

<sup>5</sup>Source codes can be accessed from [https://github.com/lihan829/AR\\_learning.git](https://github.com/lihan829/AR_learning.git)

Newman argues that the leftward spreading pattern explains that, for nonderived canonical words with the contour tone, the HL contour is typically formed on the last syllable. In disyllabic words, common patterns include HL, LH, and HH; whereas in trisyllabic words, common patterns are HLH, LHL, HHL, and LHH. The LL sequence is uncommon across all instances, whether LL, HLL, or LLL. [Zoll \(2003\)](#) argues the analysis of directional spreading leads to paradoxical outputs (e.g. \*HLL, \*LLH but LHH, HHL) and therefore proposes instead a quality-based perspective. According to Zoll, the constraint LAPSE (L-tone spreading to two syllables) overranks CLASH (H-tone spreading to two syllables). The absence of LL, particularly at the word final position, has also been explained as a Low Tone Raising rule  $L \rightarrow H / L\_ \#$  first proposed by [Leben \(1971\)](#). In summary, the tonotactic patterns attested in Hausa for canonical nonderived words are listed below in the format of constraints.

(22) Linguistically Attested Constraints in Hausa:

1. \*LH: No monosyllabic rising
2. \*(3 $\mu$ ): No three-tone contour
3. NF-Cont: No initial HL contour for di-/polysyllabic words
4. \*LL: Avoid LL sequence

The learning problem is whether these tonal constraints can be accurately captured by an algorithm. The hypothesis in the current paper is: if the hypothesis space for the learning algorithm comprises all possible ARs (under a certain size) that a language can generate, and these ARs are represented over model-theoretic representations, then the bottom-up learner will successfully capture the tonotactic grammar by searching from the most generalized AR to more specific ones against the positive data. If the algorithm can generate a set of constraints in Hausa, how do they correspond to previously identified linguistic constraints? Driven by these questions, a case study of Hausa tonotactic patterns was designed.

## 5.2 Data Preparation

The data used for the current experiment comes from Hausa dataset ([Awagana et al., 2009](#)) from The World Loanword Database (WOLD) ([Haspelmath and Tadmor, 2009](#)). The WOLD includes 41 languages. Each language dataset consist of around 1460 core words with meanings in the Loanword Typology (LWT) meaning list. For the Hausa dataset, the vocabulary contains 1668 core meaning-word pairs. The dataset includes information such as “Form”, “Meaning”, “Borrowed”, “Likelihood”, “Analyzability”, “Age”, and “Source”. The fields most relevant to the current study are listed in Table 3. To get the canonical tonal patterns in Hausa, only those words that were clearly not borrowed from other languages and were simple monomorphemic words were used. All others were filtered out. A total of 664 words transcribed in orthography remained and stored in a .csv file.



Field	Explanation	Example
Form	Transcription in orthography where á = H tone, à = L tone, âa = F tone, and double vowels indicate long vowels	kásáa
Meaning	The meaning of the form as listed in the Loanword Typology (LWT) list	“the land”
Borrowed	Four levels of borrowing certainty ranging from 0 (“no evidence of borrowing”) to 1 (“clearly borrowed”)	no evidence for borrowing
Analyzability	Describes the degree of analyzability of a word form, ranging from “Unanalyzable” (monomorphemic word) to “Analyzable phrasal” (verb-noun phrases or prep-noun phrases)	unanalyzable

Table 3: Description of fields relevant to the current study

### 5.3 Representation Conversion

Since the goal is to apply the algorithm to autosegmental representations, a Python code was created to convert all string representations into ARs. These ARs are stored as tuples, as illustrated in the *Coding* column of Table 4. Each tuple contains two pieces of information: the tonal melody of the word (e.g., 'H' or 'HLH') and a list of pairs indicating associations of tones and tone-bearing units (e.g., [(1, 1), (1, 2)]).

Within the association list, the first element  $i$  in each pair  $(i, j)$  specifies the index of the tone and the second element  $j$  specifies the syllable index. In other words,  $(i, j)$  indicates that tone  $i$  is associated with syllable  $j$ . The list orders the associations from the first tone to the last, and also the first syllable to the last syllable. Thus from the last pair in the list  $(i_n, j_n)$ , one can easily infer that the current AR structure contains  $i_n$  tone unit(s) and  $j_n$  syllable(s).

Take the word *kásáa* for instance, the AR is represented as ('H' [(1, 1), (1, 2)]) in the coding system. This reads as the melody the AR bears has one tone 'H', and has an association of [(1, 1), (1, 2)]. More specifically, (1, 1) indicates the first tone 'H' connects to the first syllable and (1, 2) means the first tone 'H' again connects to the second syllable. For the word containing contour tones such as *bâutáa*, the AR shows a tone of 'HLH' and an association of [(1, 1), (2, 1), (3, 2)] which means first two tone units HL both connects to the first syllable while the last third tone H connects to the second syllable. The entire AR contains 3 tones and 2 syllables.

After converting and eliminating identical ARs, 26 distinct ARs were identified from among 664 string representations (See Attachment A). The given data shows a maximum syllable number is 5, indicating the non-borrowed monomorphemic words in Hausa are likely within a 5-syllable length. Besides, there are 8 melodic patterns found in the data, which are H, L, LH, HL, LHL, HLH, HLHL, LHLH.

### 5.4 Implementing BUFIA over ARs

After converting all string representations into ARs, the next step is to run BUFIA on the dataset of 26 attested ARs. Since BUFIA initiates from the empty structure, it generates its immediate superfactors: a

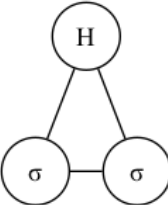
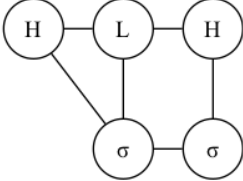
Transcription	AR	Coding
<i>kásáa</i>		( 'H' [ (1,1) , (1,2) ] )
<i>bâutáa</i>		( 'HLH' , [ (1,1) , (2,1) , (3,2) ] )

Table 4: Corresponding Strings, ARs, and Coding

floating H, a floating L, and a floating syllable. These floating units are indicated by **None** in the list. In Table 5, AR1 and AR2 contain a floating tone and a floating syllable respectively and both have **None** in their coding. AR (5a) has (2, **None**) in its association list, indicating that the second tone is not connected to any syllable; AR (5c) has (**None**, 2) indicating that the second syllable is not connected to any tone.

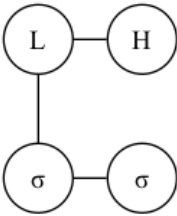
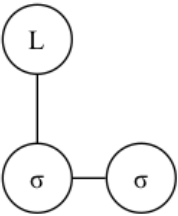
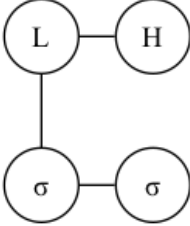
floating tone	floating syllable	float tone and syllable
		
(a) (LH, [ (1,1) , (2, <b>None</b> ) ] )	(b) (L, [ (1,1) , ( <b>None</b> , 2) ] )	(c) (LH, [ (1,1) , ( <b>None</b> , 2) , (2, <b>None</b> ) ] )

Table 5: Coding for ARs with float units

## 6 Results and Discussion

The algorithm BUFIA discovered 7 tonotactic constraints missing from the data when the syllable sizes  $s$  is no greater than 3 and the tone sizes  $t$  is no greater than 3. In Table 6, cells with 0 indicates that no constraints were found at current  $s$  and  $t$  sizes; cells with a dash indicates no new constraints were added on the basis of the previous sizes. In the following section, I will compare the linguistically attested constraints (will be referred as “rule constraints”) with the algorithm detected constraints (will be referred as “BUFIA constraints”) and discuss whether the two match and, if not, what would be the gap between the two.

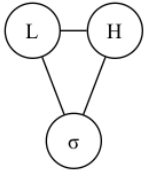
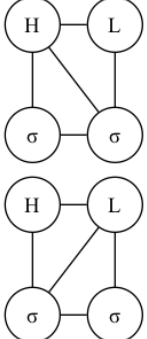
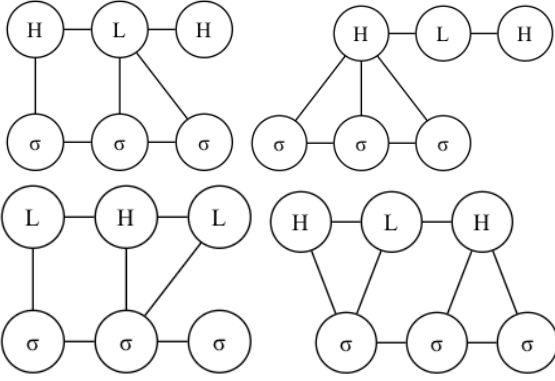
	$t = 1$	$t = 2$	$t = 3$
$s = 1$	0		-
$s = 2$	0		-
$s = 3$	0	-	

Table 6: Constraints over AR in Hausa when  $s \leq 3, t \leq 3$

Table 7 lists both Rule Constraint 1 and 2, and a BUFIA constraint concerning monosyllabic forms. Rule Constraint 1 states that the language forbids monosyllabic LH and Constraint 2 states no 3-tone contour, which are translated into an AR fashion where no LH, LHL, and HLH are linked to a single syllable. One can see that the algorithm effectively identified the absence of monosyllabic rising tone when  $s = 1$  and  $t = 2$  since (7a) and (7d) are identical. Moreover, since both (7b) and (7c) contain (7a/7d) as a subfactor, forbidding (7a/7d) automatically entails forbidding both three-tone superfactors. Consequently, the algorithm has already identified (7d) as a constraint, and therefore ceased further generating or checking its superfactors. In conclusion, we can consider the algorithm accurately captures the monosyllabic tonal patterns.

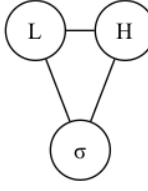
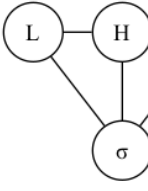
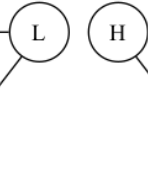
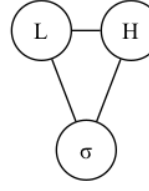
Rule Constraints			BUFIA Constraints
			
(a)	(b)	(c)	(d)

Table 7: Linguistically Attested and BUFIA Detected ARs for monosyllabic words

**Constraint 3** bans initial HL contours for di/polysyllabic nonderived words, which can be depicted as the Rule Constraint (8a) in Table 8. In this case, edge markers are added to indicate the word initial position. The graph indicates that when the word has  $\geq 2$  syllables, a structure which has HL on its initial syllable will be forbidden in Hausa. An extra syllable following the first one in (8a) is important since Hausa does allow a monosyllabic word with HL (e.g. *mâi* “oil”, *kâi* “head” ).

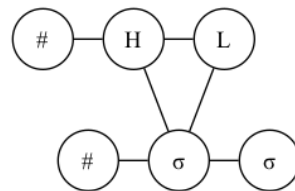
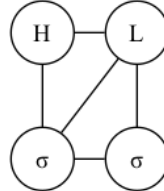
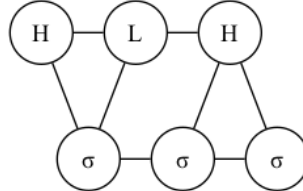
Rule Constraint	BUFIA Constraint	
		
(a)	(b)	(c)

Table 8: Rule constraints and BUFIA constraints concerning initial HL

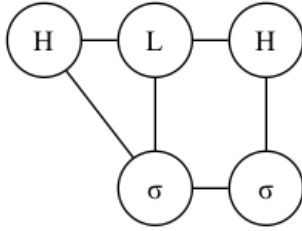
Two constraint ARs (8b) and (8c) are found by BUFIA. It should be mentioned the syllables in (8b) and (8c) are not position-specified: the leftmost syllable in the both graphs does not necessarily correspond to the first syllable in the word. In other words, not only these AR constraints themselves but also structures containing them will be banned. Now we will consider (8b) and (8c) in the most generalized case where they are viewed as word initially and comparable to (8a).<sup>6</sup>

The first one (8b) is in a disyllabic form and states that no AR has been found to contain at least two syllables where the first one is associated with a HL contour and the second one with a L tone. The second one (8c) reads that no AR has been found to have at least 3 syllables where the first one is associated with a HL contour and the rest two are linked with a H. Obviously, this indicates structures that contain (8a) but

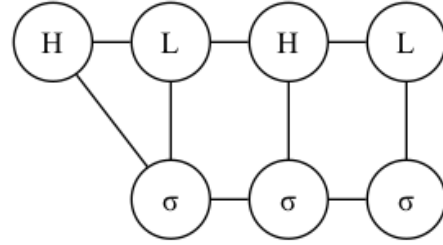
<sup>6</sup>This is admittedly a limitation of the current experiment. Thanks to Thomas Graf for pointing this out, where edge markers should have been considered to obtain a more generalized constraint without any assumption of possible positions where contour tones can be formed. The reasons that the current experiment does not consider the word-internal contours are: (1) in the traditional Autosegmental Theory, contour tones form as a result of one-to-one, left-to-right association conventions that govern the mapping of tonal melody, which leads to contour forms usually at word-final position (Goldsmith, 1976); (2) In African languages, contours demonstrate an “edge effect” and largely limited to the final tone-bearing unit of the domain (?) and Hausa contour distribution shows the same pattern (Newman, 2002); (3) even in languages where tonal melodies are not justified, contours still tend to prefer a prosodic final position, as argued by Zhang (2009), which is attributed to the longer duration of these syllables.

do not contain (8b-c) are present in the current data. We can list out the superfactors of (8a) and exclude those contain (8b-c) to find out the missing gaps between the Rule constraints and BUFIA constraints, which are listed as (23,24) below.

(23) Disyllabic form with initial HL:  $\acute{\sigma}\acute{\sigma}$



(24) Trisyllabic form with initial HL:  $\acute{\sigma}\acute{\sigma}\grave{\sigma}$



Via a reverse search, 7 words were found to match these two structures, as shown in Ex (25). This result show that the Rule Constraint of initial HL needs to be made more explicit and restricted if it aims to account for all data.

(25) Hausa words do contain an initial HL

kyânwáa	“the cat”
kûnnée	“the ear”
tsûmmáa	“the handkerchief or rag”
sâiwáa	“the root”
jînkái	“the pity”
bâutáa	“to worship, to obey”
yânyáawàa	“the fox (fennec of Sahara)”

**Constraint 4** states the avoidance of LL sequences. It is worth noting that this is not a strict ban. Newman (2002, p.606) pointed out LL exceptions are “relatively uncommon and mostly represent identifiable or presumed loanwords”. Correspondingly, the algorithm did not report any constraint when  $t = 1$  and  $s = 2$ , confirming there are cases (as in (26)) where disyllabic words containing LL sequences. By examining 4 words containing LL sequence in our dataset, it is easy to notice that many of them belong to the question words, which could contribute to the LL exceptional cases.

màcè	“female”
yàayàa	“how”
yàushè	“when”
wà nè	“which”

Additionally, although it has been commonly reported that the trisyllabic HLL and LLH are disfavored in Hausa (Zoll, 2003; Leben, 1971; Newman, 2002) the BUFIA constraint (9c) indicates that both patterns are actually present in the current data. In fact, the LLH pattern is not uncommon: 16 words were found to bear the same tonal melody with the same association (see Appendix ??). Moreover, from (9c), we

can revise the “no LL spreading” rule to: LL spreading is banned when two L-toned syllables are wedged between two H-toned syllables.<sup>7</sup>

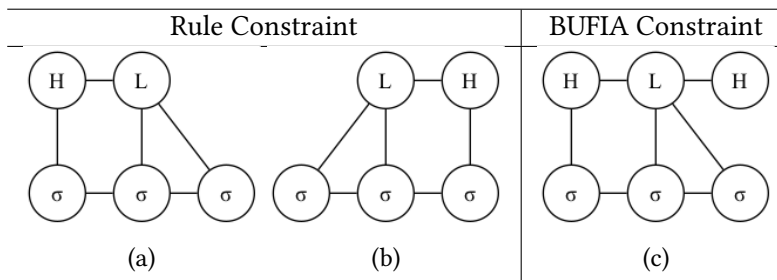


Table 9: Linguistically Predicted and BUFIA Detected ARs for HLL and LLH patterns

In summary, this section reports the results of implementing BUFIA to find tonotactic patterns over ARs and compares computationally attested constraints with linguistically reported constraints. It is found that a bottom-up learning algorithm can effectively learn tonotactic patterns over autosegmental representations. Additionally, three outcomes emerge regarding the matching between BUFIA constraints and Rule constraints. First, the two coincide, indicating that BUFIA accurately captures exactly what previous linguistic rules formalize. Second, BUFIA constraints are more restrictive and specific than linguistic constraints without overgeneralizing the patterns. Lastly, the algorithm also uncovers some patterns, usually of a larger size, that have not been reported in any other literature.

## 7 Conclusion

This paper shows the first attempt to learn tonotactic patterns over autosegmental representations using a bottom-up learning algorithm. Through a case study of Hausa, 26 distinct ARs among 654 surface-true Hausa words were found, and tonotactic constraint grammar consist of 14 ARs ( $\leq 4$  syllables and 3 tones). Furthermore, by comparing BUFIA constraints and Rule constraints, we found sometimes the two matches but more often BUFIA constraints were more explicit or specific than the Rule constraints. For example, we found patterns that were identified to be constraints occur in the positive data, which is due to the constraint only applies during certain phonological contexts. We also found patterns that haven not reported by previous linguistic analysis, which perhaps due to the structure’s growing size. These results show the feasibility of tonotactic grammar inference over autosegmental representations and extends the advantaged of using algorithm to enhance previous generalization.

However, this paper does not provide any comparison between learning results using string representations and ARs, which is a limitation and also a reasonable next step to take. Comparing the efficiency of the same algorithm learning over two types of representations would be more convincing in determining whether tonotactic patterns should be better learned over ARs.

<sup>7</sup>To be more precise, the BUFIA constraint (9c) states that no L tone can occupy two adjacent syllables and is simultaneously wedged between one H-toned syllable and another H tone whose association does not have to be specified. Theoretically, the second H tone could be linked to the rightmost syllable that L tone spreads onto or linked to a fourth syllable by itself. However, the first option is not available since LH cannot dock on the same syllable.

Questions  
below  
were  
asked by  
Thomas,  
Ellen

Additionally, the scope of this paper only reaches tonotactic patterns. There are studies on learning tonal processes using Autosegmental Input Strictly Local functions ([Chandlee and Jardine, 2019](#); [Zhu, 2020](#)), which delve deeper into the mapping between underlying forms to surface forms. It would be interesting to see how specific tonal processes could be captured by the algorithm implementation.

TG asked - any way to track the derivative and notderivative tones (eg F tone as an underlying tone vs. HL combination). I dont think i answered the question at all...

Lastly, one major point of this paper is to demonstrate the autonomous nature of tones can be learned as a bistring which connects a vocal folds configuration to the linear timing system. Building upon the findings of this study, one may ask whether and how autosegmental learning can benefit Asian language research such as Mandarin or Cantonese. Different from the African systems, tonal languages in Asia usually have a richer tonal inventory, including tones at different levels and several contour tones. Additionally, these languages usually do not have rich morphology like African languages. In particular, Cantonese and Mandarin are to a large extent seen as syllable-tone languages where each syllable has a specified underlying tone, and all syllables in a word retain the underlying tone when combined ([Yip, 2002](#)). In this specific case, learning over ARs specifically for tonotactic patterns may not significantly reduce the hypothesis space since there is no restriction on either tier of ARs. However, for languages rich in sandhi processes and showing interactions between tone and segments or other phonation features, representations that separate tone from other features may be worth investigating to see any better results.

## A Hausa Autosegmental Structures

No.	Tone Number	Syllable Number	Tonal Melody	Association
1	1	1	H	[(1, 1)]
2	1	1	L	[(1, 1)]
3	1	2	H	[(1, 1), (1, 2)]
4	1	2	L	[(1, 1), (1, 2)]
5	1	3	H	[(1, 1), (1, 2), (1, 3)]
6	1	4	H	[(1, 1), (1, 2), (1, 3), (1, 4)]
7	2	1	HL	[(1, 1), (2, 1)]
8	2	2	LH	[(1, 1), (2, 2)]
9	2	2	HL	[(1, 1), (2, 2)]
10	2	3	LH	[(1, 1), (2, 2), (2, 3)]
11	2	3	LH	[(1, 1), (1, 2), (2, 3)]
12	2	3	HL	[(1, 1), (1, 2), (2, 3)]
13	2	3	HL	[(1, 1), (2, 2), (2, 3)]
14	2	4	LH	[(1, 1), (1, 2), (1, 3), (2, 4)]
15	2	4	HL	[(1, 1), (1, 2), (1, 3), (2, 4)]
16	2	4	LH	[(1, 1), (2, 2), (2, 3), (2, 4)]
17	2	5	HL	[(1, 1), (1, 2), (1, 3), (1, 4), (2, 5)]
18	3	2	LHL	[(1, 1), (2, 2), (3, 2)]
19	3	2	HLH	[(1, 1), (2, 1), (3, 2)]
20	3	3	LHL	[(1, 1), (2, 2), (3, 3)]
21	3	3	HLH	[(1, 1), (2, 2), (3, 3)]
22	3	4	LHL	[(1, 1), (2, 2), (2, 3), (3, 4)]
23	3	4	HLH	[(1, 1), (1, 2), (2, 3), (3, 4)]
24	3	5	LHL	[(1, 1), (1, 2), (1, 3), (2, 4), (3, 5)]
25	4	3	HLHL	[(1, 1), (2, 1), (3, 2), (4, 3)]
26	4	4	LHLH	[(1, 1), (2, 2), (3, 3), (4, 4)]

Table 10: 26 distinct ARs among 655 Hausa core word list



## B Hausa Constraints

No	t	s	Melody	Association
1	1	4	L	(1, 1), (1, 2), (1, 3), (1, 4)
2	2	1	LH	(1, 1), (2, 1)
3	2	2	HL	(1, 1), (1, 2), (2, 2)
4	2	2	HL	(1, 1), (2, 1), (2, 2)
5	2	4	HL	(1, 1), (2, 1), (None, 2), (None, 3), (None, 4)
6	2	4	HL	(1, 1), (2, 2), (2, 3), (None, 4)
7	2	4	LH	(1, 1), (1, 2), (2, 3), (2, 4)
8	2	4	LH	(1, 1), (1, 2), (2, None), (1, 3), (1, 4)
9	3	3	HLH	(1, 1), (2, 2), (2, 3), (3, None)
10	3	3	HLH	(1, 1), (1, 2), (2, None), (1, 3), (3, None)
11	3	3	LHL	(1, 1), (2, 2), (3, 2), (None, 3)
12	3	3	HLH	(1, 1), (2, 1), (3, 2), (3, 3)
13	3	4	HLH	(1, 1), (1, 2), (2, None), (1, 3), (3, None), (None, 4)
14	3	4	HLH	(1, 1), (2, 2), (3, 3), (3, 4)
15	3	4	HLH	(1, 1), (1, 2), (2, 3), (3, None), (2, 4)
16	3	4	HLH	(1, 1), (1, 2), (2, None), (1, 3), (3, None), (1, 4)
17	3	4	LHL	(1, 1), (2, 2), (2, 3), (3, None), (2, 4)
18	3	4	LHL	(1, 1), (1, 2), (2, 3), (3, None), (2, 4)
19	3	4	LHL	(1, 1), (1, 2), (2, None), (1, 3), (3, None), (1, 4)
20	4	2	HLHL	(1, 1), (2, 2), (3, None), (4, None)
21	4	2	HLHL	(1, 1), (1, 2), (2, None), (3, None), (4, None)
22	4	2	LHLH	(1, 1), (1, 2), (2, None), (3, None), (4, None)
23	4	2	LHLH	(1, 1), (2, 2), (3, 2), (4, None)
24	4	2	HLHL	(1, 1), (2, 1), (3, 2), (4, 2)
25	4	3	LHLH	(1, 1), (2, 2), (2, 3), (3, None), (4, None)
26	4	4	HLHL	(1, None), (None, 1), (2, None), (None, 2), (3, None), (None, 3), (4, None), (None, 4)

Table 11: 26 tonotactic constraints in autosegmental representations in Hausa

## C LLH and HLL patterns in Hausa

Form	Meaning
tàuràaróo	“the star”
tùnkùyáu	“the flea”
<b>shànshàaníi</b>	“the centipede”
màkèesúu	“the firefly”
<b>zàzzàbíi</b>	“the fever”
<b>kàikàýíi</b>	“the itch”
màlàfáa	“the hat or cap”
gìzàagóo	“the adze”
màràicée	“the evening”
jùuyàayíi	“the anxiety”
<b>gùngùníi</b>	“to mumble”
shùugàbáa	“the president”
tàbàráu	“the spectacles/glasses”
tùrùríi	“the steam”
sùrùkái	“the parents-in-law”
kwàràngwál	“the carcass”
sábòodà	“because”

Table 12: Hausa Words with either LLH or HLL pattern

## References

- Awagana, A., Wolff, H. E., and Löhr, D. (2009). Hausa. In Haspelmath, M. and Tadmor, U., editors, World Loanword Database. Max Planck Institute for Evolutionary Anthropology, Leipzig.
- Cahill, M. (2007). More universals of tone. Ms. SIL, ms.
- Chandlee, J., Eyraud, R., Heinz, J., Jardine, A., and Rawski, J. (2019). Learning with Partially Ordered Representations. arXiv:1906.07886 [cs].
- Chandlee, J. and Jardine, A. (2019). Autosegmental input strictly local functions. Transactions of the Association for Computational Linguistics, 7:157–168.
- Coleman, J. and Local, J. (1991). The “no crossing constraint” in autosegmental phonology. Linguistics and Philosophy, pages 295–338.
- Goldsmith, J. A. (1976). Autosegmental phonology. PhD thesis, Massachusetts Institute of Technology.
- Hammond, M. (1988). On deriving the well-formedness condition. Linguistic Inquiry, 19(2):319–325.

- Haspelmath, M. and Tadmor, U., editors (2009). WOLD. Max Planck Institute for Evolutionary Anthropology, Leipzig.
- Heinz, J. (2021). Doing Phonology Computationally.
- Hyman, L. M. (2011). Tone: Is it different? The handbook of phonological theory, pages 197–239.
- Hyman, L. M. and Schuh, R. G. (1974). Universals of tone rules: Evidence from west africa. Linguistic inquiry, 5(1):81–115.
- Jardine, A. (2016). Computationally, tone is different. Phonology, 33(2):247–283.
- Jardine, A. (2017). The local nature of tone-association patterns. Phonology, 34(2):363–384.
- Jardine, A. and Heinz, J. (2015). A concatenation operation to derive autosegmental graphs. In Kuhlmann, M., Kanazawa, M., and Kobele, G. M., editors, Proceedings of the 14th Meeting on the Mathematics of Language (MoL 2015), pages 139–151, Chicago, USA. Association for Computational Linguistics.
- Leben, W. R. (1971). The morphophonemics of tone in Hausa. In Papers in African Linguistics, pages 201–218.
- Leben, W. R. (2017). Tone and length in mende. UCLA Working Papers in Phonetics (Memorial volume for Russ Schuh).
- Newman, P. (2002). The Hausa Language: An Encyclopedic Reference Grammar, volume 122. Yale University Press, New Haven, US.
- Odden, D. (1994). Adjacency parameters in phonology. Language, pages 289–330.
- Odden, D. (2013). Introducing phonology. Cambridge University Press.
- Rogers, J. and Lambert, D. (2019). Extracting subregular constraints from regular stringsets. Journal of Language Modelling, 7(2):143–176.
- Strother-Garcia, K., Heinz, J., and Hwangbo, H. J. (2017). Using model theory for grammatical inference: a case study from phonology. In International Conference on Grammatical Inference, pages 66–78. PMLR.
- Vu, M. H., Zehfroosh, A., Strother-Garcia, K., Sebok, M., Heinz, J., and Tanner, H. G. (2018). Statistical relational learning with unconventional string models. Frontiers in Robotics and AI, 5:76.
- Wang, W. S.-Y. (1967). Phonological features of tone. International journal of American linguistics, 33(2):93–105.
- Yip, M. (2002). Tone. Cambridge University Press.
- Zhang, J. (2009). Contour tone distribution is not an artifact of tonal melody mapping. Studies in the Linguistic Sciences, 33(1/2):73–132.

- Zhu, Y. (2020). Extending the autosegmental input strictly local framework: Metrical dominance and floating tones. In Proceedings of the Society for Computation in Linguistics 2020, pages 127–135.
- Zoll, C. (2003). Optimal tone mapping. Linguistic inquiry, 34(2):225–268.