Contents

1 Introduction **3**

1.1 Thesis Structure 4

1.2 Motivations 5

1.2.1 Representation 5

1.2.2 Grammar 6

1.3 Contributions 7

2 Tones: Theory and Computation **9**

2.1 Representations of Tone 9

2.1.1 Letter System 10

2.1.2 Phonetic-Based Notation 11

2.1.3 Tonal Feature 13

2.1.4 Q-Theory 14

2.2 Autosegmental Theory 15

2.2.1 Tone Autonomy 16

2.2.2 Tone spreading and shift 18

2.2.3 Floating tones 19

2.3 Computational Approaches to Tone 19

2.3.1 Tonotactics 19

2.3.2 Tonal Processes 19

2.3.3 Computational Models of Autosegmental Structure 19

3 Formal Background **21**

3.1 Formal Language Theory 21

3.2 Tonal Representations 21

3.3 Tonal Transduction 21

3.4 Learning Algorithm 21

4 Tonotactics **23**

4.1 Introduction 23

4.2 Learning Tonotactics 24

4.2.1 Tonal Mapping Problem 24

4.2.2 Learning Models 27

4.3 Preliminaries . . . 28
4.3.1 String Representations . . . 29
4.3.2 Autosegmental Representation as Graphs . . . 29
4.3.3 Structure Containment . . . 31
4.3.4 BUFIA . . . 33
4.4 Learning over Autosegmental Representations . . . 34
4.4.1 BUFIA-AR . . . 34
4.4.2 How to Expand Autosegmental Representations . . . 36
4.4.3 Containment Relation . . . 41
4.4.4 Summary . . . 42
4.5 Case Study: Hausa . . . 43
4.5.1 Experiment Setup . . . 44
4.5.2 Evaluation . . . 45
4.6 Discussion . . . 51
4.6.1 Optimisation versus Satisfaction . . . 51
4.6.2 Gradience versus Categoricity in Grammars . . . 52
4.7 Conclusion . . . 53

1 Introduction

Tonal languages are estimated to constitute 60 to 70 percent of the world's languages (Yip, 2002), making tonal phonology a critical and unique subfield within phonology. One key characteristic distinguishing tonal phonology from segmental phonology is its autonomy: tones can operate independently of the segments with which they are associated. This autonomy has been captured by Autosegmental Phonology which uses a multi-tiered structure for tonal representation (Goldsmith, 1976) and gives more accurate predictions and explanations than linear phonology (Clements and Goldsmith, 1984; Odden, 2013). Computationally, processes in segmental phonology are characterized as sub-sequential and can be handled by a deterministic transducer (Heinz and Lai, 2013) while some processes in tonal phonology such as unbounded tone spreading are found to be more complex and non-deterministic (Jardine, 2016a, 2017). However, beyond the formal properties of tones (Bird and Klein, 1990; Coleman and Local, 1991; Kornai, 2018; Jardine and Heinz, 2015), limited work has been done on tonal pattern learning, and more importantly, little effort has been made to capture the multi-linear properties of tones in the computational linguistics field (Albro, 1994; Kornai, 2018), despite their prevalence in the theoretical world.

In the proposed dissertation, I aim to provide a computational expression using formal logic to make tonal patterns and processes applicable for computational modeling and learning. While the previous attempts in tonal learning use a string representation (Chandlee, 2019; Oakden and Chandlee, 2020), this work plans to develop a multi-linear structure faithful to linguistic representation that captures the autonomy of tones. Secondly, autosegmental phonology has achieved a great number of success in African tonal languages (Clements and Goldsmith, 1984; Hyman, 2009), but far less research has explored Asian tonal languages which are known for having intricate tone-segment interactions. Incorporating these interacting features into the modeling is another goal of this project. Finally, it is noteworthy that one possible reason that tonal phonology has not obtained the same level of discussion in computational linguistics is due to limited tools and resources. No algorithm to date has been able to effectively learn the input-output mapping of autosegmental representations, which are intrinsically graphs. Besides, tonal languages are often transcribed without tonal specifications (as in African language dictionaries) or are transcribed in their underlying representations (as in Mandarin Chinese). For those languages that do have sufficient data, there are usually varying representations or models (Bao, 1999; Yip, 1989; Wang, 1967). The proposed work aims to clarify these issues and hopes to develop more tools for tonal linguists to

use in learning tonal patterns and processes. These goals will help to gain more insights into tonal phonology and its learnability, provide implications for theoretical generalizations, and get a better understanding of underrepresented languages with limited data.

1.1 Thesis Structure

Chapter 3, "Computation of Tonotactic Patterns," is ongoing and is planned to be completed by the end of this year. Building on my second qualifying paper, this work involves expanding the existing algorithm to encode syllable weight and word boundaries in Hausa, and then apply it to Japanese Nouns. Corpora for both languages have already been gathered and are ready to be implemented into the modified BUFIA pipeline. My goal is to complete both experiments by the end of this semester and write up the chapter.

Chapter 4, "Interaction between Tone and Other Features" will be carried out roughly from the winter spanning over the spring semester of 2025. The winter time will be used for research, reading, and writing for the backgrounds. From February to April, I will start experimenting with perhaps two case studies from Shanghai Chinese, Mandarin, or Fuzhou. The plan for Chapter 3 depends on the progress of the two experiments.

Chapter 5, "Tonal Process Learning," will take the longest to complete, as there is limited research on the computation and modeling of this topic. I will need additional time to learn new material and conduct further reading. In the timeline, I have allocated May to September for this chapter, but I have also reserved October and November for both writing and addressing any unresolved issues that may arise during the completion of this chapter.

The introduction chapter provides an overview, including three main points. First, tonotactic well-formedness and tonal processes differ from those at the segmental level, making it reasonable to adopt multi-tiered structures rather than string representations. Second, from a learning perspective, computational tonal phonology is still underexplored. Most existing studies use string representations; many focus on African languages, likely due to the convenience of obtaining tonal data; and very few learning models have been developed for either tonal processes or tonotactic patterns. These research gaps raise important questions regarding representation, patterns, and learning. These questions will be the main focus of this dissertation and will be addressed in the introduction chapter. Chapter One will also include logistical details, such as an outline of the dissertation.

Chapter 2 will serve as a Literature Review chapter and it will mainly cover two broader aspects around tonal phonology. First, as in the overview in this proposal, I will review the issues about how representation, grammar, and typology of tonal languages. Second, I will discuss how previous work regulates the formal properties of tonal representations using model theory or other mathematical methods.

As the first major part of the dissertation, this chapter will investigate the tonotactic grammar of tonal languages by examining forbidden structures in all surface-true forms. In this

chapter, I choose Hausa and Japanese as case studies to represent two typologically different tonal languages. Hausa uses a level tone system where H and L contrast and Japanese employs a pitch accent system. These patterns are argued to be local and their grammar can be learned by searching for the negative literal conjunctions. My goal is to represent the tonal patterns in model theory, apply a bottom-up learning algorithm to learn the grammar, and lastly compare the constraints found by the algorithm with previously reported generalizations, and see to what degree they match with each other.

The frequently discussed tonal patterns in Hausa are generalized as No Rising Contour, Avoid LL sequence, No initial Contours, and No three-tone contours, but it is unclear to what extent these rules can explain the data, and there are many counterexamples against these generalizations. In my QP2, I used a small Hausa corpus, from which I translated the orthographic representations into ARs and implemented BUFIA on it. The experiment revealed 26 distinct ARs among 664 surface-true Hausa words and identified 7 AR structures (syllable number ≤ 3, tone number ≤ 3) missing from the data, which constitute the constraints in the tonal grammar. The computationally discovered constraints either match the ones previously proposed or provide more specific generalizations that better account for the data.

1.2 Motivations

Effective tone modeling requires choosing appropriate representations and grammar. The motivation behind this work stems from the most common questions in tonal phonology, which address tonal representation, grammar, and learning.

1.2.1 Representation

The first question is what phonological representation should be used for tones. Under the influence of the classic framework of *Sound Patterns of English* (Chomsky and Halle, 1968a), where phonological strings were divided into discrete feature matrices, tones were likewise treated as intrinsic properties of segments and represented with features indicating pitch ($\pm$ H) and contour shape ($\pm$ contour). Tonal changes are the result of changing feature values and insertion/deletion of segments. Contour comes into form due to two opposite pitch features are adjacent to each other. This approach, however, was soon argued to be inadequate for predicting correct attested patterns, and it failed to capture the commonly observed patterns among tonal languages (Hyman, 2008; Odden, 2013). Therefore, "remarkably little insight was produced by the SPE framework into the behavior of tones" (Clements and Goldsmith, 1984, p.8).

Abundant evidence from African tonal languages shows that when considering tonal forms, tonal units and their associations with timing units also need to be taken into account. For example, languages with "restricted melodies," such as Mende (Leben, 2017), have a fixed set

of five melodies to any form regardless of the number of syllables of the form. These melodies "must be abstracted away from the TBUs on which they are realized" (Voorhoeve, 1973; Hyman and Schuh, 1974). To capture this, Autosegmental Theory (Goldsmith, 1976; Clements, 1980) enriches the geometry of phonological representation by proposing a multilinear structure: tones are represented on a separate autonomous tier and are related to segments on the adjacent tier through association lines governed by specific well-formedness conditions (WFC) (see also Clements and Goldsmith (1984)). Common tonal patterns such as contours or tone plateaus can be explained as multiple tones connecting to the same segment or one tone spanning across multiple segments. The insertion or deletion of one tier does not necessarily trigger the same operation on the other tier. The temporal rearrangements between tones and TBUs are handled by changing the association lines between two tiers. Autosegmental Theory has proven especially useful in explaining tonotactic patterns and tonal processes where linear phonological models often struggle (Hyman, 2011; Odden, 2013).

Feature matrices and Autosegmental Representations (ARs) are not the only approaches to tonal analysis. Inkelas and Shih (2013) introduced *Q-theory*, initially designed to account for complex diphthongs, which later expanded to contour tones. Under this framework, each tone (or segment) can be divided into three temporally ordered, quantized subsegments. This is motivated by the observation that most complex tones do not exceed three tonal levels and very rarely have four-tone contours (Shih and Inkelas, 2013). However, this approach has been argued to be logically equivalent to ARs in model theory and logical interpretations (Jardine et al., 2021). Kornai (2018) also compared different ways of encoding string-based representations of ARs. Similarly, the two tonal models proposed by Yip (1989) and Bao (1999) are also shown to be notationally equivalent using formal logic (Oakden, 2020).

Given that varying models and representations are argued to be logically equivalent, or at least translatable, formal logic is possible to represent their internal structure and facilitate learning explicitly. This is the motivation for this study.

1.2.2 Grammar

Besides representations, the second question in tonal phonology concerns the type of grammar that should be used. The first distinction is whether the grammar is rule-based or constraint-based. Take the classic tonal mapping problem for instance. In a rule-based approach, tonal mapping requires a globally evaluated direction (leftward vs. rightward), along with other parameters such as morphological category and tone quality (Goldsmith, 1976), but this cannot account for those two-edge effects, and sometimes leads to wrong predictions. Zoll (2003) argues that the directionality needs to be reconsidered, and proposes the optimal tonal mapping where directionality can be handled by the interaction between universal violable constraints: a morphological direction constraint Align and two quality-sensitive markedness constraints Clash and Lapse.

Jardine (2017) argues that the grammar of tonotactic well-formedness consists of language-specific, inviolable local constraints. These constraints can be expressed locally over ARs, which are intrinsically graphs. A well-formed AR structure can be viewed as a graph that does not contain any forbidden subgraphs. Using Model Theory, each AR graph can be abstracted into a mathematical model with a unique signature to indicate its distinct internal structure. This method captures the locality and nonlinearity makes it possible to learn tonotactic patterns over ARs, and also explains the North Karanga Shona "edge-in" pattern that Optimal Tonal Theory fails to generalize. More importantly, this approach of searching subfactors to get grammar can be learned through The Bottom Up Factor Inference Algorithm (BUFIA, Chandlee et al., 2019).

1.3 Contributions

fukc you fuck you

2 Tones: Theory and Computation

Languages with tonal distinctions have been studied as a more or less separate subfield known as tonal phonology. This division reflects a natural difference between the nature of tone and that of segments, as evidenced by a large body of empirical evidence from Asian and African languages. Rather than being inherent properties of vowels, tones may change independently of the segments that bear them. This property challenges traditional linear rule-writing approaches, which provide a cumbersome means of capturing tonal generalizations. Consequently, any discussion of tone necessarily involves a discussion of its representations, ranging from accent marks and feature-based representations to phonetically based notational systems (Chao, 1930). These approaches can be found throughout both the African and Asian tonal phonology literature. Among them, Goldsmith's (1976) autosegmental theory has become the most influential and widely adopted framework. The first half of this chapter provides an overview of the major approaches to tonal representation and typology.

The issue of representation further poses important requirements for computational modeling. Since the introduction of autosegmental theory by Goldsmith (1976), there has been a clear shift in theoretical phonology toward structurally enriched representations. In contrast, computational approaches have been slower to adopt such representations. Works by Jardine (2016a); Chandlee and Jardine (2019); Oakden and Chandlee (2020) argue that linguistically meaningful representations can reduce computational complexity. These representations include autosegmental graphs (Jardine, 2016a), metrical structures **[lamount]**, and feature-based representations for segmental learning **[magda]**. These proposals, together with their computational implementations, highlight an important direction for computational modeling: the use of linguistically informed representations to improve interpretability, data efficiency, and linguistic adequacy through the development of small, specialized language models. The second half of this chapter reviews computational approaches to tonal phonology, including finite-state transducers and other computational models. It also discusses the challenges and opportunities involved in developing computational models that effectively capture the complexities of tonal phonology.

2.1 Representations of Tone

Estimates of the proportion of the world's tonal languages vary from 42.9% **[thet]**

Yip (2002), for example, suggests that as many as 60–70% of the world's languages may be tonal, whereas Maddieson (2013) estimates the proportion to be closer to 40%. This discrepancy reflects differences in language sampling as well as in the criteria used to distinguish tone languages from languages with pitch accent or marginal tonal contrasts. Nevertheless, tone languages have a non-even geographical distribution: they are especially widespread in Africa, Central America, and East and Southeast Asia, while occurring more sparsely in Eurasia, North America, and Australia (Welmers, 1962; Maddieson, 2013; Kao, 2017; Gordon, 2016).

Since tonal languages are widespread across different geographic regions, different conventions for representing tones have been developed to accommodate the languages in questions. Yip (2002) classified three main notational traditions for tones: Africanist, Asianist, and Central Americanist. I group the latter two together for the present discussion since both use numerical notation, although they differ in how the numbers correspond to pitch levels and contours.

It is worth mentioning that, in this section, I use the terms "notation" and "representation" interchangeably. Although the two terms differ, all the methods discussed below have been used in theoretical or computational analyses of tonal phonology. I review five such systems: letter notation, number notation, feature-based representation, Q-Theory, and autosegmental representation. The first two function primarily as notational systems in previous work, whereas the remaining three are more formal representational frameworks that have also been applied beyond tonal phonology.

2.1.1 Letter System

An Africanist convention is to represent tone using accent marks. Tones are commonly marked with diacritics, even when they are not represented in a language's everyday orthography. Table 2.1 shows a basic system.

Table 2.1: Common Africanist tone notation

Tone	Symbol	Letter
High	*á*	H
Low	*à*	L
Mid	*ā*	M
Falling	*â*	F
Rising	*ǎ*	R

It should be noted that Africanist notation varies across linguists and languages. First, the unmarked tone in a two-tone system may be left without a diacritic; similarly, the mid tone is often unmarked in a three-tone system. Conversely, the macron (ā) is also used to indicate vowel length in some literature. **[Provide examples of unmarked-tone conventions in several African languages]**

Second, the notation of contour tones may be ambiguous, particularly in languages in which contour tones have greater distributional flexibility. As shown in Table 2.1, a falling tone may be written with a circumflex on the nucleus vowel (*â*), on a long vowel (*â:*), or on one element of a complex nucleus (*âi*, *ân*). Additionally, contour tones can emerge in the surface form as a morphological or phonological processes, even though the language does not have underlying contour form. One practice is to distinguish underlying contour from deritational contour with F vs HL and R vs LH. but this praxrixe is not consistnetly used. therefore, contour tones may alternatively be represented as (*áà* = *â:*). In languages that permit both concave and convex contour tones, circumflex and caron diacritics alone may be insufficient to represent more complex contours unambiguously.**[Mende tonal notation system]**

Table 2.2: Common Africanist tone notation

Tone	**Symbol**	**Letter**
High	*á*	H
Low	*à*	L
Mid	*ā*	M
Falling	*â*,*â:*,*áà*	F/HL
Rising	*ǎ*	R/LH
Fall-rise	˽	FR/HLH
Rise-fall	*a*	RF/LHL

Although these rules describe the observed mappings, they obscure the generalization that the tonal components are preserved while the number of tone-bearing units is reduced. A representation that decomposes contour tones into sequences of level tones captures this generalization more directly.

2.1.2 Phonetic-Based Notation

In the Asianist tradition, tone is commonly represented using Chao's five-level system (Chao, 1930). An example of this system is shown in Table 2.3, which lists the nine lexical tones of Cantonese.

Table 2.3: Lexical tones in Cantonese

Non-entering				Entering			
Tone	Value	Description	Example	Tone	Value	Description	Example
T1	55	High level	*si1* 'poem'	T7	5	High entering	*sik7* 'know'
T2	25	High rising	*si2* 'history'	T8	3	Mid entering	*sek8* 'tin'
T3	33	Mid level	*si3* 'to try'	T9	2	Low entering	*sihk9* 'eat'
T4	21	Low falling	*si4* 'time'				
T5	23	Low rising	*si5* 'market'				
T6	22	Low level	*si6* 'matter'				

Chao's system has been widely adopted in the study of Chinese languages and dialects. The specific rules are as follows:

1. The natural pitch range of a speaker's normal voice is divided into five levels, with 1 representing the lowest pitch and 5 the highest.
2. A sequence of digits indicates the starting point, turning point(s), and the endpoint of a pitch contour.
3. A toneless or neutral-tone syllable may be represented by 0.
4. An entering-tone syllable—that is, a syllable ending in a stop coda—is conventionally represented by a single digit.

In Central Americanist notation, tones are represented in a similar way, but in a reversed order, with 1 representing the highest pitch and 5 the lowest. This system is used in the study of Mayan languages, such as K'iche' and Q'eqchi'.

This way of representing tones is effective in describing their productive and perceptual details. However, this can also leads to its limitation and explains why such system is more treated as a notation rather than a formal mental representation of speakers. Transcriptions of the same languages can vary greatly across linguists due to allphonic (or allotonic) and interspeakers variability. Take Shanghai Chinese (also known as Shanghainese, a variaty in Wu Chinese) as an example. Shanghainese has five tones, and their tone values have had different numerical descriptions from different sources, as shown in Table 2.4.

Table 2.4: Tone values reported across different sources

	Tones				
Sources	T1	T2	T3	T4	T5
JSS 1960	53	34	13 / 14	5	2
Walton 1983	53	34	14	5	2
Xu et al. 1981–83	53	34	13	55	12
Xu & Tang 1988	53 / 52	334 / 434	113 / 223	55 / 54	12 / 23
T. Shen 1981 (1985)	52	334 / 34	113 / 13	4 / 5 / 7 (45)	23
Yuan 1983	42	34 / 434	24	5	23
Norman 1988	42	35	24	55	23

Obviously, tonal processes can be generalized using this numerical system, but the generalization appears to involve arbitrary changes in pitch values (which is, as the reader progresses in Section **[later]**, same issues for feature system). For example, the fall-rise tone (T3) in Mandarin is consistently transcribed as 214. When two fall-rise tones are adjacent, the former one will undergo a sandhi process and become a high rise. This process is formalized as 214 + 214 = 35 + 214. Although, this gives a clear depiction of the phonetic details, but the relationship between 214 and 35 appears arbitrary and does not reveal the phonological structure underlying the change.

2.1.3 Tonal Feature

From this section onward, we shift from tone "notations" to attempts of their mental "representations". In segmental phonology, consonants and vowels are represetned as a matrices of feature values, where every segment has its distinctive features to seperate itself from others.

[SPE, Pike, Woo etc]

Wang (1967) proposes one of the earliest feature theories of lexical tones to compensate for the numerical system does not directly reveal the phonological structure of tones, and enable tones to be studied in natural classes and typology. Wang uses the feature [pmCONTOUR] to distinguish level tones from contour tones; the three features [pmHIGH], [pm CENTRAL], and [pmMID] to represent pitch height; [RISING] and [FALLING] to describe the direction of pitch movement; and [CONVEX] to distinguish rise–fall contours from fall–rise contours. These features are shown in Table 2.5.

Table 2.5: Tones and their features

Tone	1	2	3	4	5	6	7	8	9	10	11	12	13
Feature	˥	˩	˧	˨	˦	˥˧	˧˥	˥˩	˩˧	˧˥˧	˩˧˩	˥˩˥	˩˥˩
Contour	−	−	−	−	−	+	+	+	+	+	+	+	+
High	+	−	+	−	−	+	−	+	−	+	−	+	−
Central	−	−	+	+	+	−	−	−	−	−	−	−	−
Mid	−	−	−	−	+	−	−	−	−	−	−	−	−
Rising	−	−	−	−	−	+	+	−	−	+	+	+	+
Falling	−	−	−	−	−	−	−	+	+	+	+	+	+
Convex	−	−	−	−	−	−	−	−	−	−	−	+	+

Under this theory, H, L, R, F, the common tones in african languages, can be presented as [+H,-contour], [-H,-contour], [-H,+contour], [+H,+contour]. In other words, R and F will be a natural class of [+contour] and L and R will be [-H]. For Mandarin Chinese, only two features are able to distinguish the froru tones, see table 2.6

Table 2.6: Pekingese tones represented by features

	Tone 3 [214]	Tone 4 [51]	Tone 2 [35]	Tone 1 [55]
FALLING	+	+	−	−
RISING	+	−	+	−

Mandarin third-tone sandhi can now be represented as follows. Compared with the numerical notation, this generalization reveals more clearly that a fall–rise tone loses its falling component when it precedes another fall–rise tone.

(1) [+falling] → [-falling] / [__, +rising] [+falling, +rising]

One property of this feature system is that it treats contour tones as single units. However, **[woo]** argues that contour tones should be decomposed into sequences of level tones. Thus, whether contour tones should be represented as single units is an important question for feature theories of tone.

Example 2 presents two common processes: H becomes R after L, F, and L becomes F after H, R. These processes cannot be described using the feature [CONTOUR] because neither L, F nor H, R forms a natural class in this system. Moreover, the theory predicts unattested patterns, as shown in Example 3.

(2) H → R / {L,F} __
L → F / {H,R} __

(3) *[+H] → [+contour] / [-H]__ equivalent to H → F / {L,R} __ (unattested)
*[-H] → [+contour] / [+H]__ equivalent to L → R / {H,F} __ (unattested)

There are also other attempts to adopt features to tone fields. Zhu (1999) merged Chao's phonetic notation and feature system, with one added feature - [Register] to captures the register issues in many chinese dialects. Schachter and Fromkin's (1968) landmark generative phonology of Akan (Kwa, Ghana) describes a high tone unassociated with a tone bearing unit (TBU) as [+seg, +tone, ØF].

2.1.4 Q-Theory

Inkelas and Shih (2013) introduced *Q-theory*, initially designed to account for complex diphthongs and later expanded to contour tones. Under this framework, each tone (or segment) can be divided into three temporally ordered, quantized subsegments. A phonological representation follows the format Q(q_1, q_2, q_3), where a traditional representation, or a segment, can be decomposed into three subsegments q. Here, "q" is loosely based on the concept of quantized temporal subphases of the unit phonologists call a "segment." A vowel with a rising-falling tone can be represented as V(à^{1}á^{2}à^3). Inkelas and Shih (2013) argue that Q-theory can compensate for the difficulty of Aperture Theory, which limits the number of aperture positions to a maximum of two.

Q-theory is motivated by the typological observation that most complex tones do not exceed three tonal levels and that four-tone contours are very rare (Shih and Inkelas, 2013). It provides a useful harmony–disharmony typology based on whether the target of change is the whole segment or a subsegment.

The first limitation of Q-theory is that, even though the typological observation suggests a lack of four-tone contours, there is in fact a typological category found in some Chinese dialects known as the "double circumflex" (Zhu et al., 2012). Double-circumflex tones have

two perceptually salient turning points. Three variants of the double-circumflex tone can be distinguished by the relative heights of these turning points, as shown in Table 2.7.

Table 2.7: Three variants of double-circumflex tones

Type	Contour	Variety	Example
Back-low	4232	Yuejiashan Mandarin (Zhongyang)	*xiang* 'make a sound'
Equal-height	3232	Deqing Wu (Zhejiang)	*ma* 'hemp'
Front-low	3242	Qiyang Xiang (Hunan)	*dong* 'freeze'

One possible solution for accommodating these tones within Q-theory is to consider a single contour (a fall or rise) as the subsegmental unit. However, if a language with double-circumflex tones also has level tones (such as 55) or short tones (also known as "entered tone", typically noted as 5), Q-theory is again insufficient to account for all these tone types within a unified representation.

Secondly, Q-theory could also be seen as a flat-string version of a nonlinear representation. Q could be seen as a tonal node, while its subsegments could be seen as tone units. Whether a subsegment or the whole segment undergoes a change also corresponds to the two tonal models proposed by Yip (1980) and Bao (1999). Jardine et al. (2021) argues that Q-theory is logically equivalent to ARs under model theory and logical interpretations. This means that, using a set of logical formulas, there is a bijective transduction from one representation to the other, and vice versa. The differences are presentational, but the representations are notationally equivalent. Oakden (2020) provides a more detailed discussion of notational equivalence.

2.2 Autosegmental Theory

Autosegmental theory marks a shift away from linear representations of generative phonology developed in *The Sound Pattern of English* (Chomsky and Halle, 1968b). Goldsmith (1976) proposes a non-linear theory in which tones and segments are represented on separate autosegmental tiers.

> In autosegmental representation, however, we posit two or more parallel tiers of phonological segments. Each tier itself consists of a string of segments, but the segments on each tier differ with regard to what features are specified in them. (Goldsmith, 1976, p.8)

Elements on different tiers are linked by association lines, which are subject to a variety of well-formedness conditions and conventions. This framework provides the important insight that speech is organized into parallel tiers and allows phonological representations to be visualized as graph-like, multitiered structures.

A core mechanism and major advantage of autosegmental theory is that associations can be manipulated independently of the elements they connect. A phonological process may

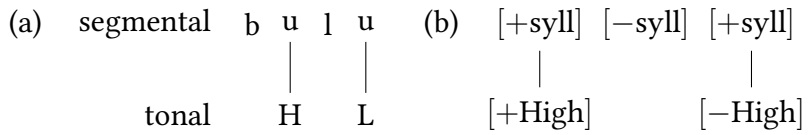


Figure 2.1: Two representations of the same tonal structure.

add or delete an association line without deleting or changing the associated tone or segment. This separation makes it possible to represent phenomena such as tone spreading, delinking, floating tones, and contour tones in a unified way.

Goldsmith (1976) formalizes a set of well-formedness conditions (WFCs) governing the relationship between tones and the units that bear them and require association changes to be minimal while maximising satisfaction of basic representational requirements. In their original formulation, these conditions require every vowel to be associated with at least one tone, every tone to be associated with at least one vowel, and association lines not to cross.

1. All vowels are associated with at least one tone.
2. All tones are associated with at least one vowel.
3. Association lines do not cross (the No-Crossing Constraint, NCC).

The significance of autosegmental theory extends beyond its ability to provide simple formalizations of tonal association. As Leben (2018) comments, "these simple rules interacted in different ways in different languages, expressing patterns that might be language-particular but using mechanisms that were universal"

The rest of this section demonstrate how autosegmental representations capture tonal processes in both African and Asian languages more effectively than string-based representations. These processes also serve as case studies in later chapters.

2.2.1 Tone Autonomy

A central claim of autosegmental theory is that tones can behave autonomously from the segments that bear them. This autonomy is not directly captured by linear representations, in which tones and segments are encoded as one unit. One evidence that challenge this practice comes from vowel deletion: deleting a vowel does not necessarily delete its associated tone. Instead, the tone may survive and reassociate with a neighboring tone-bearing unit. Such patterns provide evidence that tones and segments must be represented as distinct phonological objects.

This subsection examines vowel-deletion processes in three languages: Engenni, Etsako, and Ogoja Yala (Kao, 2017). In all three languages, the first vowel in a hiatus sequence is deleted. However, the tone originally associated with V1 undergoes a different process in each language. I classify these patterns as (1) host-dependent deletion, (2) position-based preservation, and (3) quality-based preservation. From an autosegmental perspective, the

three patterns show that how a change on the segmental tier may trigger different adjustments on the tonal tier.

Let us first examine an example from Engenni compounds - this is a pattern where I refer as host-dominant. When vowel hiatus arises at a morpheme boundary, the first vowel in the sequence V1 is deleted, while the second vowel V2 survives. What, happens to the tone borne by the deleted vowel? The answer is simple: it is deleted along with its host.

(4) **Engenni (tone deletion):** Deleting the vowel also removes its associated tone.

L H H	H L L
/ɔ̀kò # édèì/ → [ɔ̀kédèì]	/fùnú # èdhí/ → [fùnèdhí]
man canoe → "a man's canoe"	*climb palm tree → "to climb a palm tree"*

The second pattern is found in Etsako, a tonal Niger-Congo language (Elimelech, 1976). Unlike in Engenni, the tone T1 originally associated with V1 is not deleted with its segmental host. Instead, T1 survives and reassociates with its cloest available host V2. Consequently, V_2 bears both T_1 and its original tone, T_2. The two tones preserve their original order and combine to form a contour tone. I refer to this pattern as *position-based preservation*, because the outcome preserves both tones and their order rather than selecting a tone on the basis of its tonal value.

(5) **Estako (contour formation):**

L H LH	H L HL
/ówà # ówà/ → [ówǒwà]	/ɔ̀ # dé # àkpà/ → [ɔ̀dâkpà]
house house → "every house"	*he buy cup → "he buys a cup"*

The third pattern is found in Ogoja Yala. Here, the outcome is determined by competition between pitch values: the tone with the higher pitch always survives, regardless of whether the winning tone was originally associated with the deleted vowel or the surviving vowel. A high tone is preserved whenever one is present, whereas a low tone is deleted whenever it competes with a higher tone. If the two tones have the same value, it does not matter which one survives, since there is no difference in the surface form. What could change is M tone, which could be deleted if it's next to a H or presrve if it is L. This pattern can therefore be described by the tonal hierarchy H > M > L. I refer to it as *quality-based preservation* because the surviving tone is selected according to its tonal value rather than its original position.

(6) **Ogoja Yala (higher-pitch preservation)**

L H H	H L H	M L M
/dè # áchí/ → [dáchí]	/má # ɔ̀chí/ → [mɔ́chí]	/bī # ɔ̀chí/ → [bɔ̄chí]
give grass → "to give grass"	*see tree → "to see tree"*	*carry stick → "to carry a stick"*

[add another example where vowel is quality senstive, but tone is not]

2.2.2 Tone spreading and shift

Hyman (2011) points a few aspects where tonal phonology distingusih itself from segmental phonology. One striking property of tones is its long-distance effects, which sometimes can go greatly beyond the current domain. This property does not find analogous counterparts in segmental or mertrical phonology. In Yoruba **[(Laniran & Clements 2003:207)]**, when the input forms have an alternating sequence of Hs and Ls, a series of phrase-internal cntour tones are formed, resutlsing a an alternating sequences of Falls and Rs.

(7) H L H L H
má yò mì rà wé → [máyộ mǐ râ wě] 'Mayomi bought books'

Hyman referes to this as bounded tone spreading. Another type of tone spreading can go across many positions, even into different words. One example comes from Ndebele [Zimbabwe] **[Sibanda 2004]**, which hyman calss unbouned tone spreading. The H prefix /ú/ will spread all the way up to the antepenult. In Zule, it si diffenret, the high tone will delink from the original TBU, and then asscoiate to the antepenult.

(8)

Ndebele	**Zulu**	
a. *ú-kú-hlek-a*	*u-kú-hlek-a*	'to laugh'
ú-kú-hlék-is-a	*u-ku-hlék-is-a*	'to amuse (make laugh)'
ú-kú-hlék-ís-an-a	*u-ku-hlek-ís-an-a*	'to amuse each other'

The advnage of having two tiers from Autosegmental theory not only benefits african languages but also analyzing asian toneal languges (?). New shanghainese could serve as one of these examples. In New shangianese, all tones are undelrying have two level tonesfrom H, M and L. When words are compounding together, only the initial syllable's tone will be retained, all other will be delinked and 'deleted'. The first tone from the initial sylalbel will stay at the first syllable, well the seoncd tone will shift to the second syllable. SO that all syllables have a level tone which is not undelrying. If there are more words in the phrases, they automatically delinks and associates with a L tone.

	ço	*çi*	*wã*	*du*	*ŋ*
(9)	MH	HL	LH	LH	LH
	'small'	'fresh'	'yellow'	'big'	'fish'

The unbounded tone spreading so far is triggered by one trigger on one side. One special case discussed in the preivous work, refered to as "unbounded circumambient process" (Jardine, 2016b; Odden, 2013; Hyman, 2011). Jardine (2016b) defines an unbounded circumbabient process for which application is dpendnt on both sides of the target, and on both sides, there is no bound on how dar this inforamtion may be from the target **[Zigula, kenstowicz]**

An example can be drawn from Luganda compounds.

(10) **Luganda compounds** (Hyman and Katamba 2010: (26), (52))

a. /mu-tund-a/ + /bi-kópo/ → [mutunda-bikópo] 'cup seller'

b. /mu-tém-a/ + /bi-sikí/ → [mutémá-bísíkî] 'log chopper'

2.2.3 Floating tones

2.3 Computational Approaches to Tone

In previous subsection, we have discussed the pros and cons of each theory. In this section, we will go over how previous work learn tones: Usually two areas: learning tonaltactis, and the second tonal processes.

Previous work has typically reduced these structures to string representations, either through synchronized tapes or by hard-coding association types (?????). In contrast, we propose learning the association transformations directly over graphs. As a first step, we focus on tonal processes involving a single deassociation and/or reassociation operation.

2.3.1 Tonotactics

Jardine (2017, 2016b) first adopted a formal theory to analyze tonotactics. He proposed that by using model theory to point out that phonotactic grammar of tones can be generalized as a set of forbidden subgraphs. Such way of using conjunctive negative literals, or learning from the psotiive limit is effeicient. **?** implemented a bottom-up learning algorithm over autosegmental representation that learned hausa grammar over a corpus of orthpgrpahyic.

2.3.2 Tonal Processes

(?) argues that tonal phonology is different from segmental phonology also in the computational sense for many common processes seen in tonal phonology, such as bounded or unbounded tonal spreading or shift, cannot be learned using deterministic finite state machines.

They are not weakly determinisitc, which constrasts with previous typological work on segmental phonology.

Chandlee and Jardine (2019)**[chandlee glossa paper]** lists that many tonal processes that are not local over a string representation can acutally modeled in a local way using autosemgnetal representations.

2.3.3 Computational Models of Autosegmental Structure

3 Formal Background

3.1 Formal Language Theory

3.2 Tonal Representations

3.3 Tonal Transduction

3.4 Learning Algorithm

4 Tonotactics

4.1 Introduction

This paper provides one method for learning tonotactic patterns over autosegmental representations (ARs). A key insight of phonological theory is to view forms in tonal languages as *autosegmental graphs*, rather than one-dimensional linear strings (Goldsmith, 1976; Coleman and Local, 1991; Jardine, 2017; Oakden, 2020). These representations have been argued to offer advantages for analysing tonal patterns (Hyman, 2009; Clements and Goldsmith, 1984; Odden, 2013). Computationally, grammars over ARs have been shown to express both local and long-distance constraints, while maintaining a relatively low level of computational complexity (Jardine, 2017; **?**). However, despite the importance of ARs in tonal phonology and previous research on their formal properties (**?**), there has been limited work on learning tonal grammars, and, to our knowledge, no phonotactic learning models have adopted ARs.

This paper focuses on learning tonotactic grammars. Following Jardine (2017), the tonotactic grammar consists of a set of language-specific inviolable constraints over ARs. The constraints themselves only specify marked AR structures. Logically, such grammars are simply the conjunction of negative literals (**?**). In this setting, the tonotactic learning problem boils down to identifying those constraints (the marked AR structures) which are consistent with the entire dataset.

The Bottom-Up Factor Inference Algorithm (BUFIA) is a learning algorithm that deterministically searches a space of possible constraints and is provably guaranteed to find the most general constraints consistent with its input data (Chandlee et al., 2019; **?**). Crucially, it is couched in model theory (**??**) so that it can operate with any particular representational scheme.

This work presents BUFIA-AR, an instantiation of BUFIA that learns tonotactic constraints expressed with ARs. As will be explained, this implementation utilises a matrix-based representation for ARs, which is an adaptation of matrix-based representations of graphs (**??**).

BUFIA-AR was tested on a curated corpus of Hausa (**??**), a Chadic language in West Africa. The corpus consists of 621 mono-morphemic words, which altogether provide a collection of 64 distinct ARs. Using this corpus as input, we assess the performance of BUFIA-AR through a combination of qualitative and quantitative evaluations. Qualitatively, BUFIA-AR returns a grammar consisting of 11 AR constraints when syllable, tone, and mora counts are limited to three. These BUFIA-AR-learned constraints either align with, or are more specific than, those

previously proposed in the literature. The algorithm additionally identifies novel patterns that have not been previously reported. Quantitatively, a cross-validation experiment shows that BUFIA-AR achieves a mean accuracy over 90% in both token- and type-based settings, which outperforms the linguistic baseline grammar and a BUFIA learner using string representations (BUFIA-Str). Altogether, these results demonstrate the feasibility of using BUFIA-AR for tonotactic grammar inference. All data and programs used in this paper are accessible in the Dryad repository.

In this way, this paper takes a first step towards learning tonal grammars expressed in terms of ARs. We acknowledge that tonal grammars are much more than just the tonotactic component. The grammars that BUFIA-AR returns have nothing to say about how underlying representations of tonal languages are learned, nor how these underlying representations are transformed into surface representations. We also acknowledge that some surface constraints have been argued to follow from the tonal processes themselves in some languages (see Section 4.2), which is again something the grammars returned by BUFIA-AR cannot express. Nonetheless, despite these limitations, BUFIA-AR still makes a contribution to our understanding of how tonal grammars expressed in terms of ARs can be learned. We expect that some of the insights made here will persist in future research which addresses these broader problems regarding the learning of tonal grammars.

This paper is structured as follows. Section 4.2 provides an overview of the tonotactic learning problem and current models. Section 4.3 introduces formal language theory, model theory, and graph theory relevant to ARs and BUFIA. Section 4.4 provides a detailed description of the BUFIA-AR implementation. Section 4.5 presents the Hausa case study and model evaluations. Section 4.6 discusses issues relating to constraint optimisation, constraint satisfaction, and constraint weighting. Section 4.7 concludes the paper and points to some future directions.

4.2 Learning Tonotactics

4.2.1 Tonal Mapping Problem

A central question in tonal phonology concerns how languages organise tones, syllables, and/or moras into well-formed structures. A closely related modelling question asks what kinds of well-formed tonal patterns are attested in a language, and how such patterns can be systematically characterised. To illustrate these issues, we begin with a frequently discussed example: the tonal mapping problem in two West African tonal languages, Hausa and Mende. Hausa is a Chadic language, while Mende is a Southwestern Mande language spoken in Sierra Leone (?). Both languages use High (H) and Low (L) tones as basic tonemes with the mora as the tone-bearing unit (TBU). However, they show distinct patterns in how their tonal melodies are mapped to TBUs (??Zoll, 2003; Jardine, 2017).

As shown in Table 4.1, the two languages first differ in their monosyllabic contour inven-

	Mende	Hausa
contour inventory	H L mbu 'owl' L H L ngewɔɔ 'God' L H mbaa 'rice'	H L mai 'oil'
tone mapping	H L f e l a l a 'junction' L H l e l e m a 'mantis'	H L j ee m aa g ee 'bat' L H j i m i n a 'ostrich'

Table 4.1: Contour inventories and tone mapping in Mende and Hausa.

tory. Mende permits a wider range of monosyllabic contours, including light HL, heavy HL, and heavy LH, whereas Hausa allows only the heavy HL contour. Secondly, when a two-tone melody maps onto a trisyllabic word, Mende and Hausa have distinct surface forms: HL surfaces as H.L.L in Mende (*fé.là.là* 'junction') but as H.H.L in Hausa (*jée.máa.gèe* 'bat'); similarly, LH surfaces as L.L.H in Mende (*lè.lè.má* 'mantis') but as L.H.H in Hausa (*jì.mí.ná* 'ostrich')[1,2].

There are mainly three approaches to the tone mapping problem. The first identifies three key factors that determine the surface forms: *directionality* (whether tone spreads rightwards or leftwards), *morphological categories* (prefix or suffix), and *tone quality* (whether spreading is sensitive to H or L) (**?**Leben, 2017; Hyman, 2009).

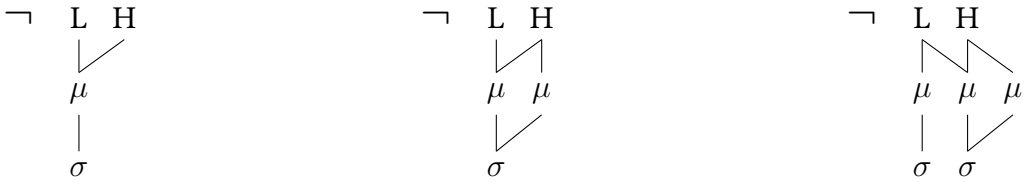
The second account is Optimal Tone Mapping (OTM; Zoll, 2003). Zoll argues that directionality-based accounts of phonological mapping can in some cases yield incorrect surface predictions. Under OTM, surface tonal patterns emerge from language-specific rankings between morphological faithfulness constraints and markedness constraints, Clash and Lapse, which are sensitive to tone quality.

The third approach, proposed by Jardine (2017), is to use language-specific, inviolable constraints, where the constraints are marked, forbidden ARs; that is, ARs to be avoided. For example, the No Short Rise constraint in Mende (Leben, 2017) that forbids the monomoraic syllable with a LH melody can be represented as in Figure 4.1a.

Following Jardine (2017), if Figure 4.1a is forbidden in the language, any AR *containing* it,

[1]This paper follows the tonal transcription method commonly used by Africanists. For instance, "H.H.L" represents two H-toned syllables followed by a L-toned syllable. In addition, tones are always marked on the nuclear vowels. Long vowels are transcribed as double vowels.

[2]**?** offers a different analysis of Mende tones than **?**'s, and provides supplemental data in which L.H.H forms also occur (e.g. *ndàvúlá* 'sling'). **?** account for the different patterns by proposing two distinct mappings.



(a) $^*\check{\mu}$ (b) $^*\check{\mu}\acute{\mu}$ (c) $^*\grave{\mu}\check{\mu}\acute{\mu}$

Figure 4.1: The forbidden structure in (a), $^*\check{\mu}$ (corresponding to No Short Rise), and two ARs in (b–c) containing it.

such as Figure 4.1b-4.1c, will also be forbidden. Formally, if Grammar G is the set of constraints in the language, such that $G = \{AR_1, AR_2, \ldots, AR_n\}$, a well-formed AR satisfies the *conjunctive negative literals*: $\neg AR_1 \wedge \neg AR_2 \wedge \ldots \wedge \neg AR_n$ (Jardine, 2017; **?**). It follows that the tonotactic learning problem is to identify a set G of banned ARs given a set of example surface forms. This paper follows the third approach of Jardine (2017).

One potential criticism of Jardine (2017), anticipated by the author and also applicable to the present work, is that it explains tonal patterns exclusively in terms of surface-true constraints. One consequence is that well-formed patterns that are not directly observable in surface forms, or at the lexical level, are not realised by the grammar. The grammar inferred at this level therefore reflects only the final outcomes of tonal mapping, rather than intermediate representations produced by tonal rules, which potentially misses important phonological generalisations.

Hausa provides a clear illustration of this point. The absence of rising tones on a single syllable can be explained as follows. An underlying LH sequence may arise as an intermediate representation, either through synchronic processes or as the result of historical change, but it is subsequently simplified to either H in non-final position or L elsewhere (**??**). Likewise, the absence of final L.L sequences on long vowels is attributed to Low Tone Raising, which changes underlying L.L to L.H when the second L occurs on a long vowel (**??**).

This example illustrates that unattested surface patterns do not necessarily reflect illicit structures in the grammar itself, but rather intermediate representations that are subsequently subject to some repair processes. While acknowledging the limitations of these grammars, we argue that learning such grammars is nonetheless valuable. They still provide meaningful insights into tonotactic patterns and serve as a foundation for further work for three reasons.

First, a grammar containing a set of forbidden ARs that is consistent with observable surface forms is not in conflict with a grammar for the tonal processes; rather, it captures their pronounceable outcomes. Given available access to a lexicon at a different level of abstractness, another grammar could likewise be learned, and these two grammars can be systematically compared. This comparison between different levels of abstractness itself yields non-trivial insights into tonal processes.

Second, considering that ARs have been long used in tonal phonology, yet are absent from

computational learning, we believe a more pressing priority is establishing representational adequacy for learning tones, which is the goal of this work. Without an explicit framework that encodes the geometry of tonal representations, modelling more complex tonal alternations, such as those discussed above, is more difficult. Establishing this representational foundation is therefore a useful precondition for future work on deeper tonal processes.

Lastly, we are not aware of existing learning models that operate with ARs directly for either tonal processes or tonotactics. In the next section, we review previous learning approaches related to tier-based representations and learning, and then introduce BUFIA, the algorithm that our model is based on.

4.2.2 Learning Models

This section reviews systems for phonotactic and tonotactic learning. One line of research employs statistical methods from machine learning (**???**). These methods have been deployed on string and syllabic representations for learning segmental phonotactic patterns as well as stress patterns, but have not been used for learning tonotactics with ARs.

? synthesises the Maximum Entropy (MaxEnt) strategy of **?** with a procedure that induces phonological tiers in order to learn non-local phonotactics. Related work by **?** (**?**), **?** (**?**), **?** (**?**), and **?** (**?**), and **?** (**?**) provide theoretical results and algorithms regarding the induction of phonological tiers for non-local phonotactics. Other work has proposed procedures for learning tiers for grammars modelling non-local phonological processes (**???**). While the successful induction of tiers from data is an important achievement of the 21st century, the constraints induced over the projected tiers are still strings, and not ARs. For example, none of the aforementioned methods can learn a constraint that describes a many-to-one association from the tonal tier to the TBU tier or vice versa.

BUFIA represents another approach (Chandlee et al., 2019; **?**). BUFIA differs from the aforementioned approaches in a couple of ways. First, it is not specific to any particular representation or data structure, such as strings or trees. For this reason, Chandlee et al.'s (2019) presentation of BUFIA abstracts away from a specific representation, and presents BUFIA in terms of arbitrary model-theoretic relational structures. As long as the representations of interest can be expressed model-theoretically, BUFIA can be implemented. Since strings, trees, and graphs are all examples of structures that have model-theoretic representations, versions of BUFIA can in principle be designed for finding constraints over any of these representations. This generality not only allows it to be instantiated and ultimately deployed for the learning of a variety of constraint types, but it also makes clear how the structure of the space of possible constraints facilitates learning.

Second, BUFIA is a deterministic learning algorithm and in its current form uses no statistical methods or randomness in its decision-making. Basically, BUFIA checks possible constraints against the data one at a time, adding those that are consistent with the data. It

returns a grammar consisting of inviolable constraints. As such, this grammar is a conjunction of negative literals: $\neg C_1 \wedge \neg C_2 \wedge \ldots \wedge \neg C_n$, where each C_i is a constraint, that is, a forbidden relational structure.

As will be explained in Section 4.3.4, BUFIA works because any representational scheme orders the expressible constraints in a natural way. By beginning with the smallest constraint, BUFIA is guaranteed to search as much of the space as possible and return the most general constraints (that is, the smallest forbidden structures) consistent with the data, before a stopping condition is met. Furthermore, BUFIA has been shown to be at least as successful in learning feature-based, segmental phonotactics of Quechua (**?**) as phonotactic learning models based on the principle of Maximum Entropy (**??**). However, BUFIA has not previously been used in tonotactic learning over non-string representations.

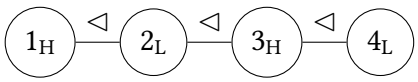
To sum up, while ARs have long been conceptualised as graph structures, no existing work learns grammars with ARs, nor has BUFIA been instantiated to operate with ARs. To address this gap, we present BUFIA-AR, a BUFIA-based learner capable of learning tonotactic constraints directly over ARs. To our knowledge, BUFIA-AR is the first learning algorithm that directly infers such constraints from ARs. As an instantiation of the original BUFIA framework, BUFIA-AR inherits its strengths and limitations.

A more detailed presentation of BUFIA-AR is given in Section 4.4. Before turning to that discussion, we first introduce in Section 4.3 several elements from formal language theory, model theory, and graph theory that are essential for understanding the algorithm and its internal structure. Some of this background knowledge is well established in computer science (**??**); our presentation is adapted to suit those only familiar with phonological theory.

Section 4.3 and 4.4 are somewhat technical, and researchers only interested in BUFIA-AR's performance may skip them. However, we believe these sections are important to include because they provide the necessary details for someone (1) to understand how BUFIA-AR is an instantiation of the more general BUFIA algorithm, (2) to implement the algorithm themselves (in order to replicate the study we conducted), and (3) to understand how the algorithm's behaviour as well as how it reflects the geometry of tonal phonology.

4.3 Preliminaries

In this section, we begin with structures for representing strings (Section 4.3.1), and then extend them to ARs using graphs and matrices (Section 4.3.2). Central to BUFIA is Model Theory (**?**), which views structures as mathematical objects, and uses formal logic to represent structural units and the relationships between them. Model-theoretic analysis has been used to compare differences between different representational theories of phonological structure (**??**). The matrix-based representation used here is a common method in mathematics and a particularly convenient way to represent AR graphs. These formal representations are essential for subsequent algorithm development and for understanding the properties and learnability



$$\mathcal{M} = \{D \mid H, L, \lhd\}$$
$$D = \{1, 2, 3, 4\}$$
$$H = \{1, 3\}$$
$$L = \{2, 4\}$$
$$\lhd = \{\langle 1, 2\rangle, \langle 2, 3\rangle, \langle 3, 4\rangle\}$$

Figure 4.2: String model for *HLHL* over $\Sigma = \{H, L\}$.

of ARs.

4.3.1 String Representations

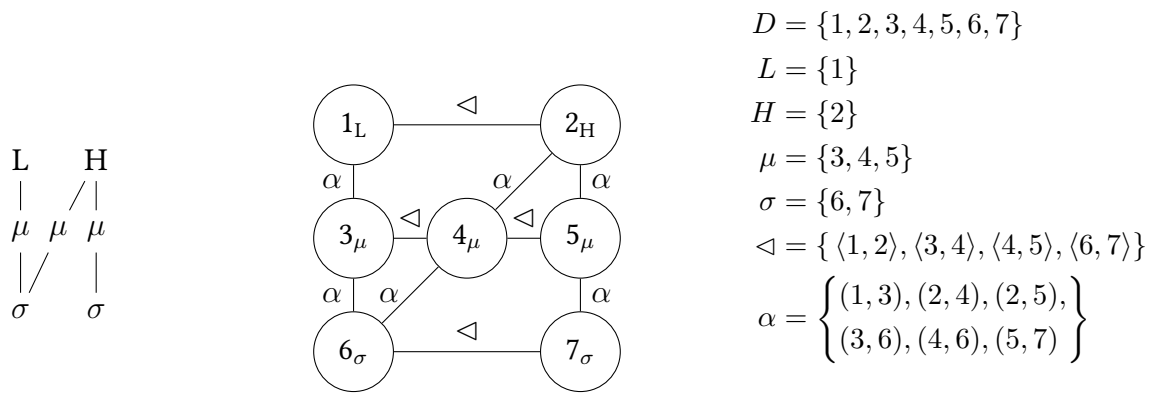
Tonal melodies can be represented as strings, that is, sequences of symbols. Let Σ be a finite alphabet of symbols and Σ^* the set of all strings over Σ. For example, in a language with a tonal inventory $\{H, L\}$, we have $\Sigma = \{H, L\}$, and Σ^* contains all possible strings over Σ, such as *HL*, *HHL*, *HLHL*, and so on. Let $|w|$ denote the length of a string w. For $0 \leq n < m \leq |w|$, let $w[n : m]$ denote a non-empty substring of w starting at position n and ending just before position m. The empty string ϵ is trivially a substring of every string.

A string can also be represented *model-theoretically* with two components: the domain D and a finite set of n-ary relations $\mathcal{R} = \{R_1, R_2, \ldots, R_i\}$ (?Chandlee et al., 2019; ?). A string model of *HLHL* is shown in Figure 4.2. In this model, $D = \{1, 2, 3, 4\}$ indicates that there are four elements in the string. Positions $\{1, 3\}$ are labelled with the unary relation H, while $\{2, 4\}$ are the positions labelled with L. Elements are connected via a binary *successor* relation, denoted by $\lhd$. For example, $\lhd = \{\langle 1, 2\rangle, \langle 2, 3\rangle, \langle 3, 4\rangle\}$ specifies that position 1 (labelled H) immediately precedes position 2 (labelled L), and position 2 immediately precedes position 3, and so on.

4.3.2 Autosegmental Representation as Graphs

Unlike the string model, where elements are ordered linearly along a single axis, an AR distributes linguistic information across multiple tiers. In this work, in addition to the basic tone and syllable tiers, we incorporate a mora tier, and treat the mora as a secondary unit subordinated to syllables to indicate the syllable weight. Therefore, we assume each syllable must be associated with at least one mora. Figure 4.3a illustrates a disyllabic form with an LH.H melody, which is visualised as a graph in Figure 4.3b, which corresponds to the model-theoretic relational structure in Figure 4.3c.

A graph consists of a set of vertices V and edges E. As shown in Figure 4.3c, the graph consists of seven *labelled nodes* which are circles numbered with $\{1, 2, \ldots, 7\}$, representing linguistic units, each labelled with its corresponding properties, such as L, H, σ, or μ. The lines connecting elements are referred to as *edges*, which may be either *directed* or *undirected*. For clarity, we distinguish two types of connections: *association relations*, denoted by α, refer



(a) AR (b) Graph (c) Model

Figure 4.3: Three representations of a disyllabic LH.H

to the undirected edges across tiers; and successor relations, denoted by $\lhd$, represent ordered relationships within a single tier. All the successor connections in the current example are given by $\lhd = \{\langle 1,2\rangle,\ \langle 3,4\rangle,\ \langle 4,5\rangle,\ \langle 6,7\rangle\}$. The association relations, on the other hand, are expressed as a set of unordered pairs: $\alpha = \{(1,3),\ (2,4),\ (2,5),\ (3,6),\ (4,6),\ (5,7)\}$. Together, the tuple $\langle H, L, \mu, \sigma, \lhd, \alpha\rangle$ constitutes the *signature* of this model-theoretic representation of an AR.

In addition to representing model-theoretic relational structures with graphs, one can use *adjacency matrices* (**??**), which use binary values to denote whether there is an edge between two nodes. Two nodes, v and w, are considered *adjacent* if there is an edge connecting them. If G is a graph with n nodes then its adjacency matrix A is an $n \times n$ matrix. If i and j are connected, the entry (i, j) has the value of 1; otherwise it is 0. Using this method, an AR with t tones, m moras, and s syllables can be encoded as a $t \times m$ tone-mora matrix and a $m \times s$ mora-syllable matrix, with successor relations implicit in the indexing. For instance, Table 4.2 provides an alternative representation of the same AR in Figure 4.3. In the tone-mora matrix, the header row reflects the tonal sequence LH, and the index column reflects the mora sequence. Since the L tone is associated with the first mora μ_1, and the H tone is associated with both μ_2 and μ_3, the corresponding cells $(L, \mu_1), (H, \mu_2), (H, \mu_3)$ are marked with 1, and the others with 0 which indicates no connection between them.[3] If we examine the matrix by columns, one can infer that the L tone occupies the first mora, while the H tone occupies the remaining two. Likewise, in the mora syllable matrix, since σ_1 is associated with μ_1 and μ_2, and σ_2 with μ_3, the corresponding cells are marked with 1. If we examine this matrix by rows, it can be seen that σ_1 is heavy while σ_2 is light. Hereafter, we refer to the rows and columns in the tone-mora matrix as the mora rows and tone columns respectively; and to the rows and columns in the mora-syllable matrix as the syllable rows and mora columns respectively.

[3]For convenience, we refer to each cell by the labels of the nodes. Node indices will be omitted from here on. For example, we may refer to $(1, 3)$ and $(2, 4)$ with (L, μ_1) and (H, μ_2), respectively. However, these are the labels of the nodes rather than the nodes themselves.

	L	H
μ_1	1	0
μ_2	0	1
μ_3	0	1

	μ_1	μ_2	μ_3
σ_1	1	1	0
σ_2	0	0	1

Table 4.2: Adjacency matrices: tone–mora matrix (left) and mora–syllable matrix (right)

These AR adjacency matrices capture precisely the same information as the graphs and model-theoretic structures. If we replace the header row and index column with the node numbers used in the graph model signature, and replace the 1s with the indices of row and column, we obtain the table filled as in Table 4.3. As can be seen, these pairs are identical to the information encoded in Figure 4.2.

	1	$\lhd$	2
3	$(1,3)$		0
$\lhd$			
4	0		$(2,4)$
$\lhd$			
5	0		$(2,5)$

	3	$\lhd$	4	$\lhd$	5
6	$(3,6)$		$(4,6)$		
$\lhd$					
7					$(5,7)$

Table 4.3: Equivalent node-indexed encodings of the adjacency matrices in Table 4.2, where $\lhd$ and α are the successor and association predicates of the model signature.

Furthermore, certain patterns can be easily identified by inspecting the matrices. For example, a series of contiguous 1s may indicate tone spreading over multiple moras if observed in the tone column, or a mora shared by multiple tones if seen in the mora row. Two 1s positioned diagonally suggest a one-to-one mapping. The associations (L, μ_1) and (H, μ_2) in the tone-mora matrix in Table 4.2 present an example of this. Some particular or ill-formed patterns can also be readily detected, which will be discussed in Section 4.4.

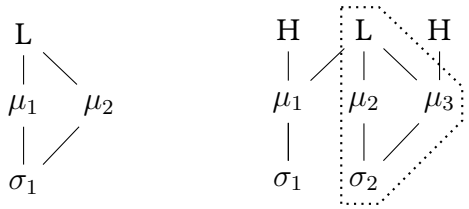
4.3.3 Structure Containment

An important concept when considering model-theoretic relational structures, either strings, sequences of feature bundles, or ARs, is *containment*. Intuitively, containment refers to the situation when one structure includes another structure inside of it. We formalise this notion using two related definitions, following Chandlee et al. (2019): *restriction* (Definition 4.1) and *containment* (Definition 4.2). Readers are referred to **?** for an alternative formulation. In the definitions below, it is assumed the structures A and B have the same signature $\langle R_1, \ldots R_n \rangle$.

Definition 4.1. *$A = \langle D^A; R_1^A, \ldots, R_n^A \rangle$ is a* restriction *of $B = \langle D^B; R_1^B, \ldots, R_n^B \rangle$ iff $D^A \subseteq D^B$ and for each m-ary relation R_i^A, we have $R_i^A = \{(x_1, \ldots, x_m) \in R_i^B \mid x_1, \ldots, x_m \in D^A\}$.*

Definition 4.2. *Structure A is* contained *in structure B ($A \sqsubseteq B$) if there exists a restriction of B, denoted B', and there exists a function $h : A \to B'$ such that for each m-ary relation R_i in the model signature: $R_i(a_1, \ldots, a_m) \in A \Longrightarrow R(h(a_1), \ldots, h(a_m)) \in B'$.*

These definitions mean that A is *contained* in B whenever there exists a subpart of B onto which A projects via a function h, and furthermore, that the relations that hold among elements of A hold among their corresponding elements in B. For instance, the AR in Figure 4.4a is contained in the AR in Figure 4.4b because the dotted region in 4.4b preserves all internal relations of 4.4a, despite the differences in the indexing. In particular, μ_1, μ_2, σ_1 in 4.4a project to μ_2, μ_3, σ_2 in 4.4b, respectively.



(a) Heavy L syllable (b) AR containing a heavy L

Figure 4.4: A heavy L-toned syllable and a structure in which it is contained

The containment relation is particularly important because it provides a natural *ordering* over the space of possible structures. The ordering is not total, like a linear order; it is a *partial* ordering. In such a partially-ordered space, some structures are comparable in terms of containment, but not necessarily all of them are. Chandlee et al. (2019) provides some additional terminology: if one structure A is contained in another structure B, then structure A is a *subfactor* of structure B, whereas B is a *superfactor* of A. They prefer these terms to substructures and superstructures, which have different technical meanings in model theory.

To illustrate, Figure 4.5 displays a non-exhaustive portion of the partial ordering of possible strings over the alphabet $\Sigma = \{a, b\}$. At the bottom of the ordering is the empty string ϵ, with more complex strings appearing higher in the ordering. Whenever two elements are connected by a line, it means that the higher element contains the lower element. For example, no line relates the a to b, because neither contains the other. However, a line is drawn from a to aa, ab and ba, because a is contained in each of these structures. As defined above, aa, ab, ba and aaa, and any other strings that contain a are superfactors of a; correspondingly, a is a subfactor of each of these strings.

There can be many layers between a superfactor and one of its subfactors. When there are no structures ordered between a superfactor and one of its subfactors, the superfactor is called a *next superfactor*. In this sense, among all superfactors of a such as $\{aa, ab, ba, aaa, aab, \cdots\}$, only those superfactors that are exactly one level above are next superfactors of a, namely $\{aa, ab, ba\}$.

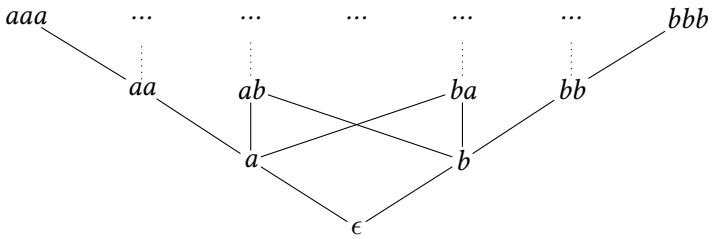


Figure 4.5: Strings over $\Sigma = \{a, b\}$ ordered by the containment relation (non-exhaustive)

4.3.4 BUFIA

Recall that given a set of positive examples, BUFIA returns a grammar just like the ones considered by Jardine (2017): a conjunction of negative literals: $\neg C_1 \wedge \neg C_2 \wedge \ldots \wedge \neg C_n$. BUFIA begins its search of the possible structure space with the most general structure (the empty structure) and progresses to more specific ones. In other words, it proceeds from the bottom of the inverted pyramid upwards. In this process, two functions are central to BUFIA: Contain and NextSupFact. For each candidate structure S, BUFIA applies the Contain function to check whether it is attested in the positive data. If it is, BUFIA concludes it is not a bona-fide constraint, discards it, and uses NextSupFact to generate a set of next superfactors to continue the search. If, on the other hand, S is absent from the positive data, it is added to the grammar as a constraint. For structures that are labelled as constraints, BUFIA does not apply NextSupFact. Consequently, all structures/possible constraints that contain it are pruned from the remaining search. The learning process continues until a stopping condition is met, which in this work is a predetermined upper bound on the size of the constraints.

Using the same example from Figure 4.5, suppose the language forbids adjacent bs, formalised as a constraint $^*bb$, which excludes strings $\{bb, bba, bbb, abb, \cdots\}$. For concreteness, assume a data set $D = \{aba, baabaa\}$. BUFIA begins from the most general hypothesis: applies Contain to check whether the empty string ϵ is attested in the positive data. Since ϵ is contained in every string trivially, Contain returns true, and BUFIA does not add ϵ as a constraint. Then, BUFIA applies NextSupFact to generate the next superfactors of ϵ, which are the two singletons a and b. In a breadth-first order, BUFIA checks whether a and b are contained in the positive data. In the same fashion, the Contain function confirms both strings are attested as substrings of D, discards them, and then generates their superfactors for later checking. When BUFIA reaches bb, Contain returns false because bb is not contained in any datum in the data set (neither aba nor $baabaa$ contains bb). BUFIA therefore adds $^*bb$ to the grammar and skips generating any superfactors of bb. The search then continues with the remaining candidates until the stopping condition is met (e.g., the size of structures exceeds a predefined size such as two).[4]

[4] An alternative stopping condition is to set a number of constraints. Both kinds of stopping conditions control the amount of generalisation. The longer the search is permitted, more constraints may be found, and amount of generalisation is reduced. Without any limits, it will overfit the data. However, when the search terminates sooner, generally fewer constraints will be found, and the grammar makes broader generalisations.

Chandlee et al. (2019) prove that BUFIA returns a set of constraints that (1) are consistent with the data (no constraint returned by BUFIA is contained in any datum in the positive data); (2) contain the most general constraints (obtains the simplest possible constraints according to the order imposed by the containment relation, e.g. only $^*bb$ is returned but not $^*bba$, $^*bbb$ or others); and (3) generate the smallest set of structures consistent with the data, among the other possible grammars that are consistent with the data subject to the same conditions (generalises conservatively).

One concern for BUFIA is its time complexity: BUFIA runs in exponential time in the worst case (Chandlee et al., 2019). However, as **?** point out, data sparsity is highly advantageous for learning since the hypothesis space is thus pruned more efficiently.

Another concern raised by the reviewers is that, in its present form, BUFIA is fragile with respect to noise. For example, if there is a single data point violating a constraint C in the data (a "noisy intrusion"), then C will not be included in the output grammar, even if C is an otherwise valid generalisation. It is important to put this current limitation in context. First, even in the absence of noise, it is currently unknown how to learn constraints expressed with ARs. From the viewpoint of scientific practice, it makes sense to address an easier problem before addressing a harder problem. Second, BUFIA can be adapted to handle noise in different kinds of ways. Prior literature suggests multiple strategies for handling noisy intrusions, exceptions, and sparse data (**?????**). Considering versions of BUFIA that are robust to all kinds of noise is a valuable direction for future research. Third, it is worth pointing out that the mere use of statistical methods for learning does not automatically make learners robust to noise (**?**). In other words, noisy data is an issue, and one that can be studied separately from the fundamental issue of successful generalisation. See **?** for additional discussion on factoring learning problems.

4.4 Learning over Autosegmental Representations

This section introduces the BUFIA-AR in Section 4.4.1. Sections 4.4.2 and 4.4.3, address how its two core functions, NextSupFact and Contain, are realised specifically for ARs. Those sections will also demonstrate that the use of adjacency matrices simplifies the computational implementation, making it more straightforward and interpretable. Our goal is to walk through each piece step by step, as they are essential for (i) understanding the logical reasoning behind tonotactic learning over ARs, (ii) properly interpreting the results that follow, and (iii) replicating our study or adapting BUFIA-AR for other AR-like structures in future work.

4.4.1 BUFIA-AR

BUFIA-AR instantiates BUFIA with ARs, and the Contain and NextSupFact functions operate in the same manner as in Section 4.3.4 but only differ in representations. In the case of

BUFIA-AR, the stopping condition is defined as predetermined upper bounds on the numbers of tones, moras, and syllables. These steps are formalised in Algorithm 1, and are discussed in more detail below.

Algorithm 1 BOTTOM-UP FACTORIAL INFERENCE ALGORITHM OVER AUTOSEGMENTAL REPRESENTATIONS

Require: Positive data D, initial empty AR ar_0, maximum tier limits t, m, s
Ensure: Set of learned generalisations G

1: $Q \leftarrow ar_0$
2: $G \leftarrow \emptyset$
3: $V \leftarrow \emptyset$
4: **while** $Q \neq \emptyset$ **do**
5: $ar \leftarrow Q.\textsc{dequeue}()$
6: **if** $ar \notin V$ **then**
7: $V \leftarrow V \cup \{ar\}$
8: **if** $\exists x \in D$ s.t. $ar \sqsubseteq x$ **then**
9: $\mathcal{A} \leftarrow \textsc{NextSupFact}(ar)$
10: $\mathcal{A}' \leftarrow \{ar' \in \mathcal{A} \mid \text{size}(ar') \leq (t, m, s) \wedge \nexists g \in G \text{ s.t. } g \sqsubseteq ar'$
11: $Q.\textsc{enqueue}(\mathcal{A}')$
12: **else**
13: $G \leftarrow G \cup \{ar\}$
14: **end if**
15: **end if**
16: **end while**
17: **return** G

BUFIA-AR initially requires three inputs: a finite set of positive data over ARs D, an initial empty AR ar_0, and a set of maximum constraint limits $t, m, s \in \mathbb{N}$. The positive data D contains a finite set of well-formed ARs. The positive integers t, m, s limit the numbers of tones, moras, and syllables of an AR considered by BUFIA-AR. Additionally, BUFIA-AR initialises 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 ARs to avoid revisiting superfactors that have already been examined.

A queue Q is used to store structures to be checked by BUFIA-AR, following a standard breadth-first search method. As shown in Line 5 and 11, there are two operations related to Q: DEQUEUE and ENQUEUE: the former takes an AR structure ar out of the queue for checking against the data, and the latter adds the next structure into Q, making it available for the learner to evaluate later in a breadth-first manner.

BUFIA-AR is called with the initial structure ar_0 set to the most generalised and least complex structure, the empty structure. The CONTAIN function takes the current structure as an argument and checks whether it is a subfactor of the data. If so, the NEXTSUPFACT function generates its next superfactors which must also satisfy the following conditions: the tone, mora, and syllable of an AR do not exceed t, m, s, have not yet been visited, and do not contain any structures already forbidden by the grammar. The eligible superfactors will be

"enqueued" for later checking. If the current structure is absent from the dataset—that is, it is not a subfactor of any data point—then it is added to G. This process repeats until the size of the structure exceeds the predetermined threshold t, m, s.

As mentioned, function Contain and NextSupFact are crucial in the implementation of BUFIA-AR. In the following sections, we will first introduce how to expand a given AR to get its next superfactors (Section 4.4.2). Then we explain how to check whether one AR is contained within another (Section 4.4.3).

4.4.2 How to Expand Autosegmental Representations

Here we explain how we compute the next superfactors of an AR. This can be achieved by adding new units to tiers or by adding associations between tiers. In a general graph-theoretic setting, any addition at any position would in principle be allowed when constructing a supergraph. However, for ARs, two principles must be taken into account: linguistic meaningfulness and computational efficiency.

By linguistic meaningfulness, we require that the resulting graphs be linguistically interpretable and obey basic universal constraints on phonological representations. For example, expansions that result in inconsistent tier projections, such as adding a mora to the tonal melody tier, or expansions resulting line crossing, will be excluded. Examples of linguistically invalid expansions are shown in Figure 4.6 with dotted lines indicating impossible associations. By computational efficiency, we aim to obtain distinct graphs while minimising

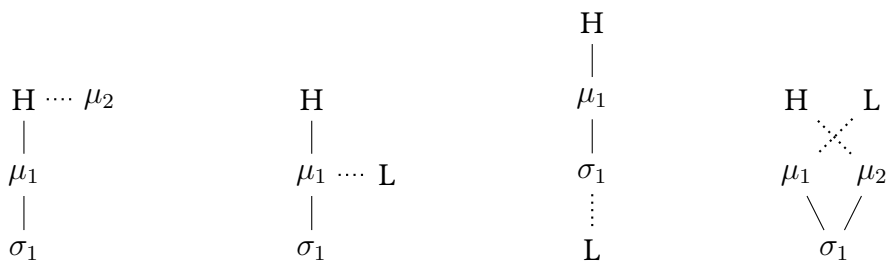


Figure 4.6: Invalid AR Expansions

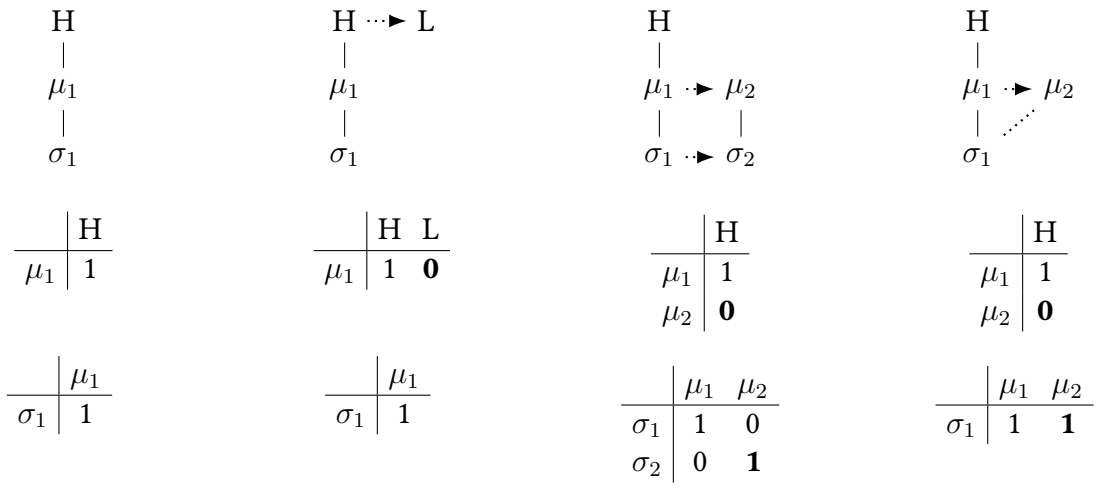
redundant construction of identical structures. In many cases, the same resulting AR can be obtained via multiple expansion paths. For example, inserting a mora between two existing moras may yield the same structure as appending a mora to the right edge of the tier or prefixing one to the left edge. Such redundant derivations incur unnecessary computational cost by repeating equivalent work, and we therefore seek to avoid them.

Adding Units

Based on these considerations, we require that new elements be added only to their projected tiers and only append them to the right edge, consistent with the temporal relation. This yields three specific Expansion Rules (ERs):

- **ER1** Append a tone to the tone tier.
- **ER2** Append a mora to the mora tier.
- **ER3** Append a monomoraic syllable to the syllable tier.

ERs (1–3) specify how a new tone, mora, or syllable can be appended to the end of each tier, analogously to extending a string. Figure 4.7 illustrates how the base AR in Figure 4.7a can be expanded in three different ways: by adding a tone L (4.7b); by adding a monomoraic syllable σ_2, which by default associates to a single mora μ_2 (4.7c); or by adding a mora to the rightmost syllable (4.7d).



(a) Base AR (b) Add a tone (c) Add a syllable (d) Add a mora

Figure 4.7: Base AR and its superfactors by adding a new tone, syllable, and mora

One might wonder how ER1 relates to the Obligatory Contour Principle (OCP). There is cross-linguistic evidence for pressures to avoid sequences of identical tones, although whether the OCP should be a universal rule has been debated (Goldsmith, 1976; **?**).

Our Hausa case study enforces the OCP by appending a tone that differs from the final tone in the current AR. There are two reasons for this practice. First, there is no clear evidence that argues against the OCP in Hausa; second, enforcing the OCP also helps reducing the search space by restricting tonal sequences. Nonetheless, relaxing this restriction is possible and is left for future research.

Adding Association Lines

Besides adding units, an AR can also be changed by modifying an association line and leaving the number of the segments or features unchanged. This ability to manipulate associations independently of segments forms the core mechanism of autosegmental theory. Concerning the association lines, Goldsmith (1976) proposes a set of well-formedness conditions (WFCs) as a guidance of how association lines may be added or deleted. These conditions require

association changes to be minimal while maximising satisfaction of basic representational requirements.

1. All vowels are associated with at least one tone.
2. All tones are associated with at least one vowel.
3. Association lines do not cross (No-Crossing Constraint, NCC)

From a computational perspective, unrestricted growth of association lines is costly, as it allows arbitrary connections between any two nodes. The NCC plays a crucial role in reducing this complexity by limiting the set of admissible association lines. There have been several works providing formal definitions of the WFCs (Goldsmith, 1976; Coleman and Local, 1991; Jardine and Heinz, 2015; Jardine, 2017). Based on these, the NCC rule can be formalised as following. Letting α be the set of existing associations, a new association (i, j) is valid if and only if there does not exist an association $(i', j') \in \alpha$ such that $(i' > i \wedge j' < j) \vee (i' < i \wedge j' > j)$.

At the same time, to further avoid adding association lines which give rise to identical graphs, we impose an additional constraint for computational efficiency: an association line may be added only to the the first unassociated unit. In other words, a fully-specified AR (all tones, moras, syllables are already associated with at least one element), will not add new association lines. If an AR contains more than one unassociated unit, the new association must be formed to the first available unassociated unit. This is to satisfy WFC 1 and 2. Finally, from a linguistic perspective, we assume that the association lines between moras and syllables are pre-specified. Consequently, new association lines are formed only between moras and tones. This yields the ER4:

ER4 Add an association line between tone and mora, subject to the following conditions:

a. do not violate the No-Crossing Constraint;
b. involve at least one unassociated unit; and
c. be formed adjacent to the final mora–tone association.

The final tone-mora association in an AR can be expressed mathematically as in Definition 4.3.

Definition 4.3. *An edge (t, m) in the tone–mora matrix is defined as the final tone–mora association if $t \in H \cup L$, $m \in \mu$, and $(t, m) \in \alpha$, and there exists no edge $(t', m') \in \alpha$ such that $t' > t$ or $m' > m$.*

Once the final tone-mora association is identified at (t, m), there are three options, as listed in Definition 4.4, to form an association line (t', m') while adhering to ER4:

Definition 4.4. *Given the final tone–mora association (t, m), a new edge (t', m') adjacent to (t, m) can be formed in three ways:*

a. $(t', m') = (t{+}1,\ m)$ *if* $t{+}1 \in D$
b. $(t', m') = (t,\ m{+}1)$ *if* $m{+}1 \in D$
c. $(t', m') = (t{+}1,\ m{+}1)$ *if* $t{+}1,\ m{+}1 \in D$

For example, the AR in Figure 4.8a contains an unassociated H tone and two unassociated moras, μ_3 and μ_4, as reflected by zeros in the tone–mora matrix. Taking the final association (L, μ_2) as the reference point, three new ARs can be formed by adding one adjacent association: (i) linking the unassociated tone to the nearest mora (Figure 4.8b); (ii) linking an unassociated mora to the nearest tone (Figure 4.8c); or (iii) directly linking an unassociated tone and an unassociated mora (Figure 4.8d).

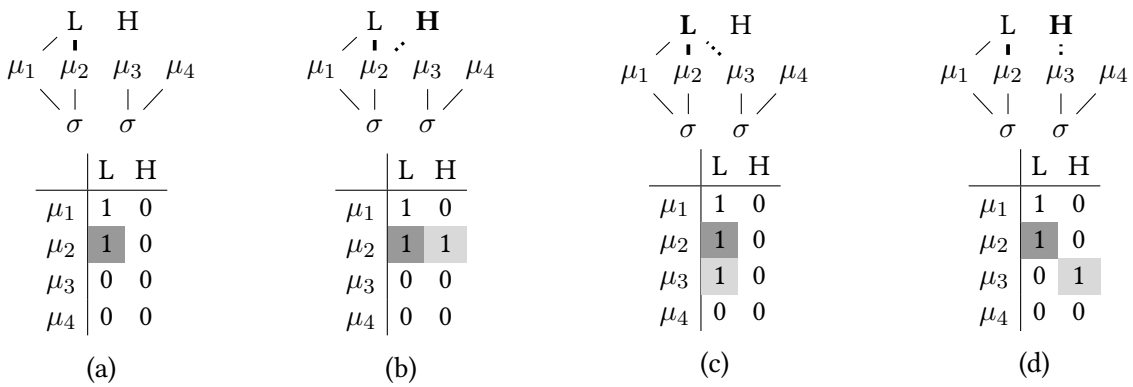


Figure 4.8: Three options to form new associations

These operations offer several advantages both computationally and linguistically. Computationally, we control the growth of the next superfactors (by reducing the number of times a unique structure is produced) without sacrificing expressiveness. We do not claim the expansion rules are optimal, only that they outperform naive approaches to superfactor generation. Linguistically, these operations generate a full tonal inventory, including light, heavy, superheavy syllables, as well as simple and contour tones and tone-spreading patterns.

Recall from Section 4.3 that factors stand in a containment relation, whereby superfactors contain subfactors. This relation yields a hypothesis space, illustrated in Figure 4.9, in which lines (some shown as dotted for readability) connect a subfactor (lower-ranked) to its next superfactors (higher-ranked). If two factors are not connected by any path in this graph, they are incomparable, such as the unassociated H and the unassociated L.

Interim Summary

To summarise, this subsection has presented a concrete method for computing the next superfactors of a given AR. Four Expansion Rules were defined: they expand the structure either by adding a unit on a single tier or by introducing a new association.

We used adjacency matrices to represent the internal associations, and stipulated some rules to reduce redundancy. Specific matrix patterns were shown to diagnose ill-formed structures.

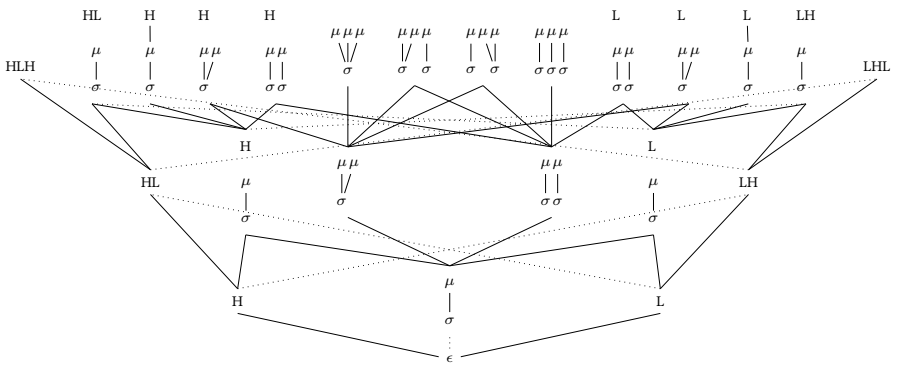
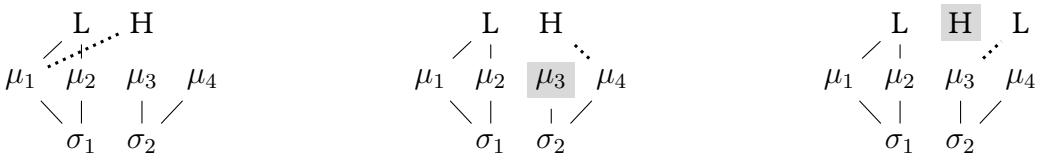


Figure 4.9: A partially-ordered hypothesis space over ARs



	L	H
μ_1	1	1
μ_2	1	0
μ_3	0	0
μ_4	0	0

(a) lines crossing

	L	H
μ_1	1	0
μ_2	1	0
μ_3	0	0
μ_4	0	1

(b) mora skipping

	L	H	L
μ_1	1	0	0
μ_2	1	0	0
μ_3	0	0	1
μ_4	0	0	0

(c) tone skipping

Figure 4.10: Invalid association lines

For example, Figure 4.10 illustrates three invalid ARs. An anti-diagonal (from upper right to lower left) of 1s marks a line-crossing violation, as in Figure 4.10a. A column or row that is filled exclusively with zero indicates an unassociated tone or mora. Such zero-domination is problematic when it occurs between two columns or rows that contain associations, as this pattern indicates that an intervening unit has been skipped, resulting in an improper association formed across it (as in 4.10b and 4.10c) as they violate the WFC 1 and 2.

An alternative interpretation of skipped units is that they are *floating*. We distinguish between *floating* and *unassociated* units. Floating tones, which are common in African languages, are not associated to any TBU and manifest themselves under specific phonological conditions. Unassociated units, by contrast, are available to be linked during the expansion process.

BUFIA-AR does not express floating tones *per se*, only unassociated units. In principle, floating tones could be incorporated by either introducing them into structures explicitly, or by interpreting unassociated units as floating. We leave this matter for future research.

It should be emphasised that explicitly defining how ARs are expanded is crucial for two reasons. First, the expansion procedure forms the basis of BUFIA-AR: without it, the space of possible constraints cannot be traversed, or even represented, and thus the learnability of

AR constraints cannot be properly understood. Second, this procedure highlights the multilinear structure and associations between autosegmental tiers, which are hallmarks of tonal phonology.

4.4.3 Containment Relation

Section 4.4.2 introduces the four ways to obtain next superfactors of an AR. This section addresses a different question: given two ARs, how can we determine if one AR *contains* the other?

This question is important for any implementation of BUFIA because it provides an explicit, formal criterion for checking containment rather than an impressionistic intuition. For example, one might assume that a light HL contour is contained in a heavy HL syllable, and similarly that a light LH is contained in a heavy LH. Under this assumption, any language that allows a heavy contour would necessarily also allow the corresponding light contour. This is, however, not accurate. As shown in Table 4.4, Hausa accepts heavy HL but not light HL, whereas Mende accepts heavy LH but not light LH.

In other words, the algorithm must be able to determine that none of the structures in Table 4.4 contains one another. Containment between ARs is not determined solely by tonal string matching or by the number of TBUs, which are straightforward to verify (e.g., *L* is contained in *HLH*, and a bimoraic sequence is contained within a trimoraic sequence). A more challenging task is to identify if the associations across tiers (i.e., the temporal arrangement of autosegments) in one AR are fully preserved in another.

	light LH	**light HL**	**heavy LH**	**heavy HL**
AR	L H \ / μ | σ	H L \ / μ | σ	L H | | μ μ \ / σ	H L | | μ μ \ / σ
Hausa	N	N	N	Y
Mende	N	Y	Y	Y

Table 4.4: Contour Inventory in Hausa and Mende

There is more than one method to achieve this algorithmically. We use a matrix-based containment checker to determine whether two ARs, even with the same tone sequence and the same number of syllable and mora, will have the same associations.

The matrix containment follows exactly parallel to the general notion of any structural containment: matrix A is *contained* in matrix B if there exists a *submatrix* of B, denoted B', that functions as a restriction discussed in Definition 4.1, such that all relations within A are relatively preserved in correspondence with those in B', as in Definition 4.2. The following

two definitions (Definition 4.5 and 4.6) specialize these concepts to the matrix representation of ARs.

Definition 4.5. *Let matrix A be an $m \times n$ matrix. Matrix B is a* submatrix *of A by taking k $(1 \leq k \leq m)$ contiguous rows and v $(1 \leq v \leq n)$ contiguous columns from A.*[5]

Definition 4.6. *An $m \times n$ matrix $A = [a_{ij}]$ is* contained *in a matrix $B (A \sqsubseteq B)$, if there exists an $m \times n$ submatrix in B, denoted by $B' = [b'_{ij}]$ such that for all $i \in [0, m], j \in [0, n],\ b'_{ij} \geq a_{ij}$.*

Definition 4.5 defines what a submatrix is. Definition 4.6 states that a smaller matrix $A = [a_{ij}]$ is contained in a larger matrix $B = [b_{ij}]$ if we can find a submatrix $B' \subseteq B$, with the same dimensions as A, such that the value of each entry in B' is at least as large as the corresponding entry in A; that is, $b'_{ij} \geq a_{ij}$ for all i, j. Since the entries of these matrices are binary, this condition implies two cases:

(a) If $a_{ij} = 1$, meaning there is an association between node i and node j in the smaller matrix A, then b'_{ij} must also be 1. This ensures an association found in A is preserved in B' at the same position;

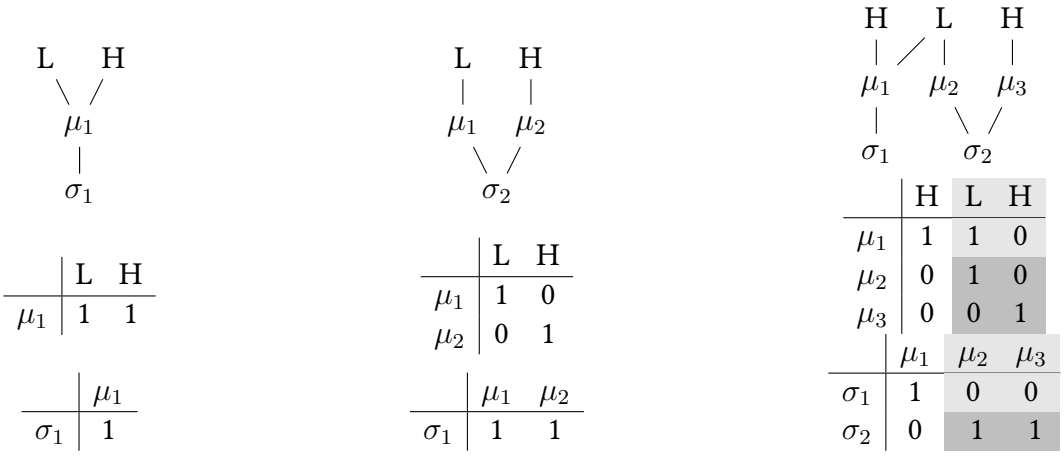
(b) If $a_{ij} = 0$, meaning no association exists between node i and node j in the smaller matrix A, then b'_{ij} may be either 0 or 1. In other words, B' may or may not have an association at the corresponding position. It is permitted to contain additional associations beyond those in A, but it must not omit any that are present in A.

The comparison proceeds by first identifying whether the tonal string of one structure is contained within another, and then verifying whether the corresponding association matrices align. Even when a tonal substring match is found, containment is established only if the associated tone–mora and mora–syllable submatrices match within the aligned region. The full algorithm is provided in the Supplementary Material, along with a detailed walk-through. Figure 4.11 illustrates this procedure: (a) is not contained in (b) or (c), whereas (b) is contained in (c), since the submatrices in (c) (the shaded cells) match the matrices of (b) in terms of associations as well as tonal strings, but do not match those of (a).

4.4.4 Summary

This section established BUFIA-AR, which and implementation of two functions that are core to the BUFIA-AR framework: NextSupFact and Contain, which are used in the case study described in the next section.

[5] The term "submatrix" is generally defined by deleting arbitrary rows and/or columns from the original matrix. However, in our case, we require a more restricted definition: the remaining rows and columns must be *contiguous* in the original matrix, which reflects tonal locality.



	L	H
μ_1	1	1

	μ_1
σ_1	1

(a)

	L	H
μ_1	1	0
μ_2	0	1

	μ_1	μ_2
σ_1	1	1

(b)

	H	L	H
μ_1	1	1	0
μ_2	0	1	0
μ_3	0	0	1

	μ_1	μ_2	μ_3
σ_1	1	0	0
σ_2	0	1	1

(c)

Figure 4.11: Light LH (11a), heavy LH (11b), and a superfactor of heavy LH (11c)

4.5 Case Study: Hausa

is grounded in autosegmental theory. In particular, we explained the rationale

This section presents a case study applying BUFIA-AR to identify tonotactic constraints over ARs in Hausa. We examine Hausa because suitable data was available; we are unaware of a similar dataset for Mende. The analysis focuses on non-derived native Hausa words, examples of which are shown in Table 4.5. As mentioned earlier, in Hausa monomoraic syllables can bear either H or L, but cannot bear contours. A bimoraic syllable can take an HL contour (each tone associates to one mora) but LH contours are not allowed. Lastly, Hausa does not have superheavy syllables, therefore monosyllabic three-tone contours are also not allowed.

Monosyllabic	ɗáa 'son'	mèe 'what'	mâi 'oil'	
Disyllabic	ƙá.sáa 'land'	mù.tûm 'person'	tú.dùu 'hill'	màa.gée 'cat'
Trisyllabic	í.nú.wàa 'shadow'	ƙàn.ƙá.ráa 'ice/snow'	múu.jì.yáa 'owl'	fà.dá.màa 'swamp'

Table 4.5: Nonderived native Hausa words (**?**)

?, p.606 notes non-derived disyllabic words exhibit canonical normal tone patterns: H.L, L.H, and H.H. For trisyllabic nouns, H.L.H, H.H.L, L.H.L, and L.H.H are very common, but L.L.H is less common with basic lexical items. Words ending in L.L, either disyllabic or polysyllabic words containing L.L, are relatively uncommon, especially when the second syllable bears a long vowel. When L.L forms do occur, they are typically restricted to ideophones or loanwords (**?**, p.606). Zoll (2003) argues that the absence of H.L.L and L.L.H is a result of the constraint *Lapse (which forbids L from spreading across adjacent syllables) outranking *Clash (which forbids H from spreading across two syllables). In addition, for canonical,

Generalisations	Interpretation
*Rise$_\sigma$	Syllables may not be associated with rising tones (LH)
*Contour$_\mu$	Moras may be associated with at most one tone.
*3t-Contour	No trimoraic contour
*NonFinal-Fall	No nonfinal HL contour for polysyllabic words
*Lapse	No L spreading across syllables

Table 4.6: Phonological Generalisations in Hausa

*Rise$_\sigma$	*Contour$_\mu$	*3t-Contour	*NonFinal-Fall	*Lapse
L H | | μ μ \ / σ	X X \ / μ | σ	μ μ μ |/ σ	H L | | μ μ μ \ / | σ σ	L / \ μ μ | | σ σ

Table 4.7: Phonological Generalisations Translated into forbidden ARs

non-derived forms falling (F) tones typically fall on the last syllable (?, p.606).

Based on the above, the linguistically attested constraints in Hausa are summarised in Table 4.6, which are translated into AR constraints in Table 4.7. In this study, we treat these constraints as the baseline grammar against which performance of BUFIA models is evaluated. Specifically, we investigate whether BUFIA-AR can accurately learn AR-based constraints, and if so, how the computationally attested constraints (henceforth, BUFIA constraints) correspond to previously identified linguistic ones (baseline constraints). We also include a BUFIA-string model as a comparison, which learns constraints over the same data using string representations.

4.5.1 Experiment Setup

The data used for the current experiment comes from the Hausa dataset (?) in The World Loanword Database (WOLD) (?). It consists of around 1460 core words whose meanings are present in the Loanword Typology meaning list. This list includes forms and concepts that all languages have terms for. For the Hausa dataset, the vocabulary contains 1668 core meaning-word pairs, along with other information in fields titled Form, Meaning, Loanword Status, and Analysability. To ensure consistency with previous generalisations, which are based on canonical, non-derived tonal patterns, we restrict our dataset to monomorphemic words that are clearly non-borrowed, and all others were filtered out. In addition, forms containing glides (ƙw, ƙy, kw, ky, gw, gy) were excluded to avoid ambiguous syllabification and uncertain syllable weight. As a result, a total of 621 words transcribed in orthography remained and were stored in a .csv file. All orthographic forms were first syllabified (?) then converted into its corresponding AR representation using the method described in Jardine and Heinz (2015).

Table 4.8 shows how an orthographic form with tonal markers is transformed into its corresponding matrix representation. After conversion and removal of duplicate ARs, 64 distinct AR types were identified among the 621 words, which serve as the positive dataset on which BUFIA-AR operates.

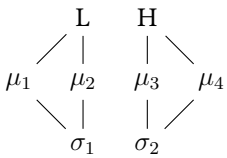
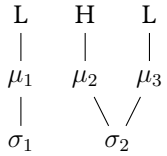
Form	Tones	AR Diagram	Matrix-based Encodings
kòo.gíi ‘river’	LH		$\begin{array}{c\|cc} & \text{L} & \text{H} \\ \hline \mu_1 & 1 & 0 \\ \mu_2 & 1 & 0 \\ \mu_3 & 0 & 1 \\ \mu_4 & 0 & 1 \end{array}$ $\begin{array}{c\|cccc} & \mu_1 & \mu_2 & \mu_3 & \mu_4 \\ \hline \sigma_1 & 1 & 1 & 0 & 0 \\ \sigma_2 & 0 & 0 & 1 & 1 \end{array}$
mù.tûm ‘person’	LF		$\begin{array}{c\|ccc} & \text{L} & \text{H} & \text{L} \\ \hline \mu_1 & 1 & 0 & 0 \\ \mu_2 & 0 & 1 & 0 \\ \mu_3 & 0 & 0 & 1 \end{array}$ $\begin{array}{c\|ccc} & \mu_1 & \mu_2 & \mu_3 \\ \hline \sigma_1 & 1 & 0 & 0 \\ \sigma_2 & 0 & 1 & 1 \end{array}$

Table 4.8: Corresponding lexical forms, ARs, and their matrix-based encodings.

We ran BUFIA-AR with the number of syllables, moras, and tones limited to a maximum of three. The number three was chosen for two reasons. First, two of the baseline constraints in Table 4.7 have three moras. If BUFIA-AR stopped searching for constraints before structures with three moras, it would not find those constraints. Second, we wanted to show that BUFIA-AR could return constraints with unassociated units, which is a unique aspect of ARs as compared to linear representations.

4.5.2 Evaluation

In this section, we provide two types of evaluation: qualitative and quantitative. The qualitative evaluation examines how the grammar learned by BUFIA-AR compares with the baseline grammar. The quantitative evaluation uses k-fold cross-validation, a common method in machine learning for assessing how well a predictive model generalises to unseen data.

Qualitative Evaluation: Constraints Comparison

BUFIA-AR found 11 constraints when the numbers of syllable, mora, and tone are limited to three. More generally, the output of BUFIA-AR for smaller values s, μ and the number of tones are subsets of the 11 constraints in Table 4.9. Cells with a dash indicate that no constraints were found with those s and μ sizes.

	$\mu = 1$	$\mu = 2$	$\mu = 3$
$s = 1$	L H \ / μ | σ (a) H L \ / μ | σ (b)	L H | | μ μ \ / σ (c)	μ μ μ |/ σ (d)
$s = 2$	-	-	H L | /| μ μ μ |/ | σ σ (e) H L |\ | μ μ μ | | σ σ (f) H L H /|\ μ μ μ |/ | σ σ (g)
$s = 3$	-	-	H L | /| μ μ μ | | | σ σ σ (h) L H L |\\ μ μ μ | | | σ σ σ (i) H L H |\ μ μ μ | | | σ σ σ (j) L H L | |\ μ μ μ | | | σ σ σ (k)

Table 4.9: Hausa constraints returned by BUFIA-AR when $s \leq 3,\ t \leq 3,\ \mu \leq 3$

As shown in Table 4.10, all of these constraints together constitute three categories: one where the baseline constraints have matching BUFIA constraints, one where the baseline constraints do not have corresponding BUFIA constraints, and one where BUFIA constraints do not have corresponding baseline constraints. In Table 4.10, this third category of constraints also provides names for the BUFIA constraints (4.9e-4.9h) in terms of the melody and syllabification implicated by the structures in Figure 4.9. Boldface indicates the tone on heavy syllables. These labels serve only as shorthand for the ARs shown in Table 4.10 and may be ambiguous, as they do not fully capture the structures.

Present in both BUFIA and Baseline	**Baseline Constraints Absent from BUFIA**	**BUFIA Constraints Absent from Baseline**
*Rise$_\sigma$ (4.9c) *Contour$_\mu$ (4.9a–4.9b) *3T-Contour (4.9d)	*NonFinal-Fall *Lapse	*F.L (4.9e) *H.F (4.9f) ***H**.H-LH (4.9g) *H.L.L (4.9h) *L.L.L-HL (4.9i) *H.H-LH (4.9j) *L.H.H-L (4.9k)

Table 4.10: Comparing baseline constraints with BUFIA constraints

The first category is when the BUFIA-AR attested constraints align with the linguistic generalisations previously argued for in the literature. For monosyllabic forms (when $s = 1$),

BUFIA-AR returned four constraints: a monomoraic LH (4.9a), a monomoraic HL (4.9b), a bimoraic LH (4.9c), and a trimoraic structure (4.9d). Each of these matches a baseline constraint. In particular, (4.9a) and (4.9b) together account for the absence of monomoraic contour tones, or the constraint *Contour$_\mu$. The two constraints further rule out any structures in which a single mora associates with more than two tones. Secondly, the BUFIA-constraint (4.9c) precisely forbids a heavy LH contour, which is identical to *Rise constraint in Table 4.7. Its counterpart, a heavy HL syllable, is not returned, which appropriately addresses the contour asymmetry problem discussed earlier. Finally, (4.9d) correctly disallows trimoraic syllables as expressed by the constraint *3T-Contour. BUFIA-AR does not return any specific tonal forms for this structure, since any tonal sequence it carries will ultimately violate the forbidden structure. One thing to note here is that both Contour$_\mu$ and 3T-Contour describe structural information: Contour$_\mu$ prohibits binary branching on a mora node, while 3T-Contour prohibits ternary branching on a syllable node. The reason that 3T-Contour can be ruled out by a single constraint 4.9d, whereas two constraints, 4.9a and 4.9b, are needed to account for Contour$_\mu$ is that in the current implementation, tonal specifications must be fully specified, and we did not include an umbrella feature, or permit under-specified tonal nodes, that would match both H and L. Overall, BUFIA constraints (4.9a-4.9d) correctly reflect frequently reported patterns regarding the tonal inventory, TBU, and syllable weight, and arguably cover some of the most fundamental aspects of Hausa tonology. This demonstrates that BUFIA-AR is not only capable of learning these tonotactic patterns but also expressing them in terms of ARs. The second category regards baseline constraints without matching BUFIA constraints. There are two of these constraints, *NonFinal-Fall and *Lapse. Each of these appear to be partially, not totally, accounted for by some of the BUFIA constraints. For example, constraint (4.9e) partially captures the generalisation *NonFinal-Fall, which requires that HL contours do not occur in non-final position, and in particular, when the following syllable bears a L tone. This is a clue that the BUFIA constraints appear to allow for the possibility of an HL contour followed by a H syllable because no constraint banning such a configuration was returned. Indeed, a reverse search revealed six words that exhibit these structures, as shown in Example (11).[6] These data points indicate that the constraint *NonFinal-Fall is too strong and it must be further refined if it is to account for the full range of observed data.

(11) Hausa words exhibiting HL.H pattern (*NonFinal-Fall)

a.	jîn.ƙái	"the pity"
b.	kûn.née	"the ear"
c.	tsûm.máa	"the handkerchief or rag"
d.	sâi.wáa	"the root"
e.	bâu.táa	"worship"
f.	yân.yáa.wàa	"the fox (fennec of Sahara)"

[6]?, p. 607 lists eight non-derived words with an initial HL tone, possibly reflecting the historical loss of a low-tone vowel. The word *kûn.née* 'the ear' is one of these; the others did not appear in Newman's records.

Consider the next baseline constraint *Lapse, which disfavours the L-tone spreading, therefore predicts the absence of H.L.L and L.L.H (Zoll, 2003). The BUFIA constraint (4.9h) *H.L.L bans one form of H.L.L when all three syllables are monomoraic, but does not entail banning other H.L.L structures, such as ones that contain a heavy syllable. An example in the data is the word *sá.bòo.dà* 'because.' Even though this word expresses the H.L.L pattern, it does not contain the structure shown in (4.9h) and therefore it satisfies BUFIA constraint (4.9h) *H.L.L.[7]

Furthermore, BUFIA-AR did not return a constraint prohibiting the factor L.L.H. In fact, there are also instances of this pattern as shown in Example (12).

(12) Hausa words exhibiting the L.L.H pattern

a.	mà.là.fáa	"the hat or cap''
b.	tùn.kù.yáu	"the flea''
c.	gì.zàa.góo	"the adze''
d.	tàu.ràa.róo	"the star''
e.	gàa.jì.màa.rée	"the cloud''

To summarise, the reason BUFIA did not find the baseline constraints *NonFinal-Fall and *Lapse is simply because they are not true of the data, which has multiple forms violating those constraints.

This brings us to the third category, which includes BUFIA constraints without matching baseline constraints. Generally, these constraints partially match *NonFinal-Fall and *Lapse. However, since both baseline constraints are not generally true of the data, the BUFIA constraints are typically more specific refinements of them, forbidding narrower classes of forms. We can classify these constraints to two types: constraints concerning contour (4.9e, 4.9f) and constraints concerning spreading (4.9g, 4.9i, 4.9j, 4.9k).

Constraint (4.9e) and (4.9f) specify two environments that forbid a HL contour: when a HL syllable is followed by a light L-toned syllable (4.9e, *F.L), or when it is preceded by a light H-toned syllable (4.9f, *H.F). The forms of F.L and H.F, termed as "non-contrasting" tone by Zoll (2003), have been reported in other languages. For instance, in Mende, disyllabic R.H undergoes tone absorption and simplifies to L.H, and F.L simplifies into H.L (Leben, 2017). In Hausa, since the light syllables with rising tones are banned, BUFIA-AR only returns *F.L and *H.F constraints. **?**, p.598 points out that F.L optionally changes to H.L in complex relative pronouns and adverbs; however, such simplification does not apply to our dataset, with which BUFIA-AR remains consistent.

In terms of the constraints concerning spreading, traditional OT includes *Clash and *Lapse,

[7]Visually, $\begin{array}{ccc} \text{H} & & \text{L} \\ | & & /\,| \\ \mu_1 & \mu_2 & \mu_3 \\ | & | & | \\ \sigma_1 & \sigma_2 & \sigma_3 \end{array}$ might appear to be contained in $\begin{array}{ccccc} \text{H} & & \text{L} & & \\ | & & /\,| & & \backslash \\ \mu_1 & \mu_2 & \mu_3 & & \mu_4 \\ | & | & / & & | \\ \sigma_1 & \sigma_2 & & & \sigma_3 \end{array}$ but in fact the two structures are incomparable in the partially-ordered space because μ_3 is associated to σ_3 in the former but not in the latter.

which ban adjacent syllables linked to either H or L. Among BUFIA constraints, *Clash and *Lapse are reflected in more specific forms as in constraint (4.9g-4.9k).

Constraints (4.9e, *F.L), (4.9h, *H.L.L), and (4.9i, *L.L.L-HL) address the cases which forbid L spreading in particular circumstances. Constraint (4.9e) and (4.9h) prevent it after H tones, and (4.9i) prevents a trisyllabic L span before a HL melody.

Similarly, the constraint (4.9f, *H.F), (4.9g, ***H**.H-LH), (4.9j, *H.H-LH), and (4.9k, *L.H.H-L) restrict H-spreading in different environments. For example, (4.9g) and (4.9j) prohibit an H tone (preceding an LH) from spreading across two syllables, while (4.9k) forbids an H tone from spreading to two syllables when it is wedged between two L-toned syllables.

Finally, four BUFIA constraints (4.9g,***H**.H-LH), (4.9i,*L.L.L-HL), (4.9j,*H.H-LH), and (4.9k,*L.H.H-L) potentially demonstrate the expressive advantage of ARs over string representations. These constraints involve unassociated units, either unassociated tones or syllables. For example, Constraint (4.9g) describes the following condition: when a H spreads over a heavy syllable to another syllable, the tonal tier can only be extended by adding one more tone. The association of the unassociated tones does not have to be specified. A similar pattern is seen in (4.9i) where if L tone spreads over three light syllables, the tonal tier will only accommodate one more tone. These patterns reflect a central fact regarding autosegmental phonology: units from different tiers do not need to be synchronised. This idea is more difficult to express with purely linear string representations.

In summary, this subsection reports the qualitative evaluation of BUFIA-AR by comparing BUFIA constraints with the constraints reported by linguists. It was found that BUFIA-AR is able to learn tonotactic patterns over AR, and that there are interesting differences between the constraints produced by BUFIA-AR and those posited by earlier researchers.

Quantitative Evaluation: Cross-validation

The k-fold cross-validation procedure partitions the dataset into k folds. In each iteration, BUFIA-AR is trained on $k-1$ folds and evaluated on the remaining held-out fold. This process is repeated k times so that each fold serves as the evaluation set exactly once.

Since BUFIA is a deterministic categorical learner, the distinction between token- and type-based datasets does not ultimately affect the learned grammar. However, we report both settings for completeness. In the type-based setting, the dataset consists of 64 distinct Hausa AR types, randomly divided into four folds (16 types per fold), with 48 used for training and 16 for testing. In the token-based setting, the full dataset of 621 tokens is also randomly divided into four folds (155/156 per fold). Within the train set (465/466 in total), there were 55–62 unique types.

In each round, BUFIA-AR is trained on three folds and evaluated on the remaining fold. The baseline grammar is evaluated on the same held-out fold. Accuracy is defined as the proportion of correctly accepted forms in the held-out set (also known as *recall*).

	Type-Based				Token-Based			
Fold	**Train/Test**	**AR**	**Str**	**Baseline**	**Train/Test**	**AR**	**Str**	**Baseline**
1	48/16	0.938	0.500	0.812	465/156	0.987	0.949	0.936
2	48/16	0.875	0.500	0.812	466/155	0.994	0.974	0.942
3	48/16	0.875	0.812	0.938	466/155	1.000	1.000	0.955
4	48/16	0.938	0.812	0.688	466/155	0.994	0.942	0.974
Avg		**0.907**	**0.656**	**0.812**		**0.994**	**0.966**	**0.952**

Table 4.11: Four-fold cross-validation in both type- and token-based settings. In the token setting, the number of distinct types in Train are indicated in parentheses.

In addition, we apply the same procedure using a string-based version of BUFIA (BUFIA-Str). Both BUFIA-AR and BUFIA-Str require a user-defined complexity bound, which determines the maximum size of constraints considered by the algorithm. However, this notion of complexity is not directly comparable across representations. For BUFIA-AR, it is controlled by multiple parameters (t, m, s) that limit different tiers; we set these to $t = m = s = 3$ to match the constraints found in 4.9. BUFIA-Str uses a single complexity parameter k. Since the maximum constraint length in Table 4.10 is 7 (i.e. 4.9j), we set $k = 7$ to match this bound. This setting allows us to assess whether constraints identified over AR can also be recovered using a string-based representation under a comparable complexity limit.

As shown in Table 4.11, BUFIA-AR outperforms BUFIA-Str and the baseline grammar, and reaches an average accuracy of 0.907 in the type-based setting and 0.994 in the token-based setting. The overall performance increase in the token-based setting is expected because of the uneven distribution of tokens over types: most types are rare (**?**). Consequently, frequently attested structures are likely to be attested in training and form a larger proportion of the testing data.

BUFIA-Str performs below the baseline in the type-based setting (0.656 < 0.812), but outperforms in the token-based setting (0.966 > 0.952). The lower performance in the type-based setting is likely due to the relatively high complexity threshold ($k = 7$), which leads to overfitting in the type-based setting. In the token-based setting, this effect is compensated. These results suggest that the BUFIA-AR supports stronger generalisation over distinct types, including rare ones, even without frequency information.

In addition, across all four folds, BUFIA-AR consistently returns the constraints (4.9a–4.9f) and (4.9h), while the remaining constraints vary slightly across folds. Altogether, this indicates that BUFIA-AR not only captures the core generalisations of Hausa tonotactics but also extends to rarer and more exceptional patterns that the baseline fails to account for. Overall, the quantitative evaluation demonstrates that BUFIA-AR generalises robustly to unseen data and empirically outperforms a similar algorithm using string representations and a theoretically motivated baseline grammar.

4.6 Discussion

4.6.1 Optimisation versus Satisfaction

The grammars output by BUFIA-AR compute the well-formedness of a word by checking whether its structure *contains* any forbidden structures (the negative literals), or equivalently, whether it *satisfies* all of the constraints. This is in contrast to globally *optimising* over *violable* constraints. The pattern obtained by satisfying the 11 BUFIA constraints captures the asymmetric distribution of spreading and contour formation while accommodating the rarer data points. On the other hand, In Zoll (2003)'s Optimal Tone Mapping program, tonal mapping arises from the optimisation over morphological directionality constraints (Align-R and Align-L) and quality-sensitive markedness constraints (*Clash and *Lapse).

What are we to make of these differences? Consider how each approach explains the common absence of non-contrasting forms such as *F.L and *H.F. Optimal Tone Mapping utilises *Clash and *Lapse to help explain the absence of these unattested forms. Zoll observes this is an improvement over the traditional mapping approach, which explained their absence by the lack of a motivation for spreading. Zoll (2003) writes, "with one-to-one linking built into the Association Convention, nothing automatically forces one tone to spread onto a syllable that already bears tone." On the other hand, BUFIA-AR returns no *Lapse or *Clash constraints, but instead returns more specific versions of them. There appears to be a trade-off between optimisation-based grammars using simpler constraints and satisfaction-based grammars using more complex constraints. One may then prefer optimisation-based grammars over satisfaction-based ones on the grounds that its constraints are simpler.

However, we argue that such a conclusion is premature. First, from a computational perspective, this trade-off may not be as significant as it appears because the phonological patterns being described are ultimately finite-state (**?**) or subregular (**??**). There are many ways to decompose a phonological pattern into parts, and it has been argued that understanding each of these ways is valuable (**?**).

Second, it is also the case that the dataset obtained from **?** shows that Zoll's (2003) proposal, as it is presented, cannot adequately capture the exceptional data points in the dataset because *Lapse is undominated. Zoll acknowledges in footnote 18 that exceptions to the general pattern in Hausa could be accommodated by a prespecificational account (**?**). However, another alternative that Zoll does not consider would be to add to the grammar refined versions of *Lapse like the ones that BUFIA-AR found. In other words, even optimisation-based grammars would benefit from more complex constraints of the variety found by BUFIA-AR.

One objection to this latter approach is that it stands in opposition to a core idea behind Optimality Theory, which is that CON is a universal set of simple constraints and all language-specific variation is a product of the interaction of these constraints via ranking. Under this perspective the only learning that needs to take place regards the ranking of the constraints

(??). This argument is fine as far as it goes, but we are aware of no theory of CON which provides a simplicity criterion its constraints must satisfy.

It is also the case that more recent research has shown how constraints themselves can be learned from examples (????). This line of work coheres with the idea that not all constraints in all languages are universal, and that some are in fact learned. Provided a theory of generative grammar admits the learning of some language-specific constraints, it is not clear to us why it would be a problem for language-specific constraints that are a little more complex than the ones attributed to CON to be included in a grammar.

In this regard, it is important to point out that BUFIA-AR does not return arbitrarily complex constraints. The collection of constraints output by BUFIA-AR forms a grammar which corresponds to a particularly weak form of logic (the conjunction of negative literals), and the structures themselves (ARs) are representations well-motivated by linguistic theory.

In sum, the debate between optimisation-based and satisfaction-based approaches is one with a long history (see for example ?), and will not likely be settled soon. Nonetheless, the fact remains that BUFIA-AR returns a set of inviolable constraints consistent with the data comparable to the ones utilised in earlier analyses.

4.6.2 Gradience versus Categoricity in Grammars

Another ongoing debate in phonotactic learning concerns whether the grammar should be categorical or gradient. Learning models like BUFIA(-AR) make clear distinctions between what is well-formed and what is not since constraints are inviolable. By contrast, gradient learning models, such as MaxEnt (??) discussed in Section 4.2.2, treat constraints as *weighted*. In these models, Constraints extracted from a lexicon are assigned with probabilistic weights that correspond to well-formedness judgements from native speakers, with lower probabilities representing lower degrees of acceptability. In this line of research, gradience is inherent to the grammar, and probabilistic induction is seen as inevitable for phonotactic learning (??).

We argue that the absence of probabilistic induction in BUFIA-AR should not be regarded as a deficiency for two main reasons. First, whether weights are *necessary* in learning or in the grammar remains a matter of debate. Second, although the presented implementation of BUFIA-AR is categorical, it can be augmented with probabilistic values if a gradient model is desired in certain learning scenarios.

Regarding the first point, several lines of argument have challenged the view that phonotactic grammars are gradient. Directly equating probability with well-formedness, as ? do, has been argued to be mistaken because unlikely outcomes can still be well-formed and likelihood ultimately decreases as size increases, but well-formedness does not (?). Others have argued that gradience in acceptability judgements may stem from performance-related factors, such as misperception (?) or psycholinguistic processing (?). Additionally, there is some evidence that categorical models can perform just as well as, and sometimes even better than, gradient

models (??????). For example, ? conducted a comparative study between BUFIA and MaxEnt on Quechua (?) and obtained comparable results. ? demonstrates that when MaxEnt weights are either equalised or entirely removed, the model can actually yield stronger predictions. Other recent studies further highlight the versatility of categorical learners: ? shows how BUFIA can be modified to learn positive grammars, while ? proposes a categorical learning model capable of filtering out exceptions.

Another point of view that has been expressed is that the gradient/categorical distinction is not intrinsic to phonotactic learning or the phonotactic grammar itself, but only differs in how grammars assign values to constraints. In other words, the categorical and gradient approaches can be analysed the same way; only the values that a learner or grammar associates with constraint evaluation changes (???). We are sympathetic to this view because this reconciliation foregrounds grammatical structure and backgrounds the particularities of the values. The values (whether binary, scalar, or real) are details, but the grammatical structure itself is paramount. This is precisely what theoretical computational studies strive to elucidate (?).

This is why it is worth noting that the current categorical learning model of BUFIA-AR can be flexibly extended into a gradient learning model. Doing so only requires that a different value-assigning function be applied to the constraints. Accordingly, the current BUFIA-AR model can be adapted into a gradient version in multiple ways. Weights could be assigned to constraints obtained by BUFIA-AR on the basis of any number of calculations including frequency of the factors, their expected frequency, or based on a regression fit to these or related numbers. Another approach would be to make the ARs themselves gradient. In the current implementation, the binary values (0/1) in the matrices simply represent the presence or absence of association lines, which treats all types of associations in one of these two ways. These binary values can be replaced with probabilistic values of specific associations, so as to get a gradient grammar (??).

Finally, we emphasise that, despite the long-standing shift in tonal phonology from segment/feature-based representations to non-linear representations, existing gradient learning models do not, as far as we are aware, operate directly over non-linear representations such as ARs. We believe that it will be easier to obtain a gradient tonotactic learning algorithm over ARs by modifying BUFIA-AR as opposed to adapting existing string-based gradient learning models to ARs.

In sum, the categorical nature of BUFIA-AR does not prohibit it from being an important contribution to our understanding of how tonotactic patterns can be learned.

4.7 Conclusion

This paper presents the first tonotactic learning algorithm over autosegmental representations to our knowledge. Through a case study of Hausa, we constructed a database containing 64 distinct ARs from 621 surface-true Hausa words and discovered, using BUFIA-AR, a tonotactic

constraint grammar consisting of 11 constraints (limited to three syllables, three tones, and three moras). Furthermore, by comparing BUFIA constraints with the baseline constraints, we found partial overlaps: in some cases, the two matched, but more often the BUFIA constraints were more specific. We also found data points in conflict with previous analyses, and identified novel patterns not previously reported in the linguistic literature, such as certain constraints prohibiting tone spreading and contour formation in particular environments. These findings demonstrate both the feasibility of tonotactic grammar inference over ARs and the advantages of using algorithmic approaches to refine and extend earlier generalisations.

Another contribution of this paper is the modelling of ARs, including their expansion and containment checking, in a manner that is both computationally efficient and linguistically faithful. The pipeline employed in this study, from data preparation and orthography-to-AR conversion to the implementation of BUFIA-AR, can serve as a general tool for exploring tonotactic constraints in other languages, and as a foundation for future work on more complex tonal process learning.

There are several directions for future research. It would be valuable to investigate the psychological reality of the constraints inferred by BUFIA-AR, seeking external evidence from cognition and perception studies. Such evidence could bear on the stopping criteria for BUFIA-AR.

Another future direction is to learn tonal processes. **?** presents a method within the framework of Boolean Monadic Recursive Schemes (**?**) for learning phonological transformations over segments represented with features. Since her approach makes use of BUFIA-style generalisation and model-theoretic relational structures, it becomes an interesting question how her approach can be extended to input-output mappings over ARs.

Finally, although this paper primarily targets tonotactic patterns in African languages, the proposed autosegmental learning framework naturally extends to Asian tonal languages such as Shanghai and Fuzhou Chinese, where tonal distributions are tightly conditioned by syllable structure and segmental features (Yip, 2002). It would be interesting to explore how to model such an enriched tonal geometry, and how tonal grammars could be learned over ARs in these languages.

Bibliography

Albro, D. M. (1994). *AMAR: a computational model of autosegmental phonology.* PhD thesis, Massachusetts Institute of Technology.

Bao, Z. (1999). *The structure of tone.* Oxford University Press.

Bird, S. and Klein, E. (1990). Phonological events. *Journal of Linguistics*, 26(1):33–56.

Chandlee, J. (2019). A Computational Account of Tone Sandhi Interaction. *Proceedings of the Annual Meetings on Phonology*, 7.

Chandlee, J., Eyraud, R., Heinz, J., Jardine, A., and Rawski, J. (2019). Learning with partially ordered representations. In *Proceedings of the 16th Meeting on the Mathematics of Language*, pages 91–101.

Chandlee, J. and Jardine, A. (2019). Autosegmental input strictly local functions. *Transactions of the Association for Computational Linguistics*, 7:157–168.

Chao, Y.-R. (1930). A system of tone letters. *Le maître phonétique.*

Chomsky, N. and Halle, M. (1968a). The sound pattern of english. *Haper and Row.*

Chomsky, N. and Halle, M. (1968b). *The Sound Pattern of English.* Harper & Row, New York.

Clements, G. N. (1980). *Vowel harmony in nonlinear generative phonology.* Indiana University Linguistics Club Bloomington.

Clements, G. N. and Goldsmith, J. (1984). Autosegmental studies in bantu tone: introduction. *Autosegmental studies in Bantu tone*, 3:1–17.

Coleman, J. and Local, J. (1991). The "No Crossing Constraint" in Autosegmental phonology. *Linguistics and Philosophy*, pages 295–338.

Elimelech, B. (1976). *A Tonal Grammar of Etsako.* PhD thesis, University of California at Los Angeles.

Goldsmith, J. A. (1976). *Autosegmental Phonology.* PhD thesis, MIT.

Gordon, M. K. (2016). *Phonological typology.* Oxford University Press.

Heinz, J. and Lai, R. (2013). Vowel harmony and subsequentiality. In *Proceedings of the 13th Meeting on the Mathematics of Language (MoL 13)*, pages 52–63.

Hyman, L. M. (2008). Issues in african language phonology. *UC Berkeley PhonLab Annual Report*, 4(4).

Hyman, L. M. (2009). The representation of tone. *UC Berkeley PhonLab Annual Report*, 5(5).

Hyman, L. M. (2011). *Tone: Is it Different?*, chapter 7, pages 197–239. John Wiley & Sons, Ltd.

Hyman, L. M. and Schuh, R. G. (1974). Universals of Tone Rules: Evidence from West Africa. *Linguistic inquiry*, 5(1):81–115.

Inkelas, S. and Shih, S. S. (2013). Abc+ q: Contour segments and tones in (sub) segmental agreement by correspondence. In *21st Manchester Phonology Meeting*.

Jardine, A. (2016a). *Locality and non-linear representations in tonal phonology*. PhD thesis, University of Delaware.

Jardine, A. (2016b). *Locality and non-linear representations in tonal phonology*. PhD thesis.

Jardine, A. (2017). The local nature of tone-association patterns. *Phonology*, 34(2):363–384.

Jardine, A., Danis, N., and Iacoponi, L. (2021). A formal investigation of q-theory in comparison to autosegmental representations. *Linguistic Inquiry*, 52(2):333–358.

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

Kao, S. (2017). *Phonological Learning Bias in Tone Patterns*. PhD thesis, State University of New York at Stony Brook.

Kornai, A. (2018). *Formal phonology*. Routledge.

Leben, W. R. (2017). Tone and length in mende. *UCLA Working Papers in Phonetics (Memorial volume for Russ Schuh)*.

Leben, W. R. (2018). A brief history of autosegmental phonology with special reference to its founder. In Brentari, D. and Lee, J. L., editors, *Shaping Phonology*, pages 26–45. University of Chicago Press, Chicago.

Maddieson, I. (2013). Tone. *The world atlas of language structures online*.

Oakden, C. (2020). Notational equivalence in tonal geometry. *Phonology*, 37(2):257–296.

Oakden, C. and Chandlee, J. (2020). A computational analysis of tone sandhi ordering paradoxes. In *Proceedings of the 50th Annual Meeting of the North East Linguistic Society (NELS 50)*, volume 1.

Odden, D. (2013). *Introducing Phonology*. Cambridge University Press.

Shih, S. and Inkelas, S. (2013). A subsegmental correspondence approach to contour tone (dis) harmony patterns. In *Proceedings of the annual meetings on phonology*.

Voorhoeve, J. (1973). Safwa as a restricted tone system. *Studies in African Linguistics*, 4(1):1.

Wang, W. S.-Y. (1967). Phonological features of tone. *International journal of American linguistics*, 33(2):93–105.

Welmers, W. E. (1962). The phonology of kpelle. *Journal of African Languages*, 1:69–93.

Yip, M. (1980). *The tonal phonology of Chinese*. PhD thesis, Massachusetts Institute of Technology.

Yip, M. (1989). Contour Tones. *Phonology*, 6(1):149–174.

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

Zhu, X. (1999). *Shanghai Tonetics*. LINCOM studies in Asian linguistics. LINCOM EUROPA.

Zhu, X., Zhang, T., and Yi, L. (2012). Types of concave tones: With a discussion on the syllabic nature of the rising tone in beijing mandarin. *[Zhongguo Yuwen]*, (5):420–436.

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