Learning Tonotactic Patterns over Autosegmental Representations[*]

Han Li
Stony Brook University

1 Introduction

Autosegmental representations (ARs; Goldsmith 1976) have been widely used in tonal phonology to capture the autonomy of tones from segments, and have demonstrated advantages over linear theories especially among African tonal languages (Hyman, 2011; Odden, 2013). Despite their significance in tonal phonology, limited research on leveraging ARs for learning (Coleman and Local, 1991; Jardine, 2013; Jardine and Heinz, 2015; Kornai, 2018).

This paper presents the first attempt to learn tonotactic patterns using ARs instead of string-based representations. Following Jardine (2017), this paper considers the grammar of tonotactic well-formedness as language-specific, inviolable local constraints over ARs. A well-formed AR structure can be modeled as a graph that does not contain any forbidden subgraphs which can be learned by identifying the missing structures in the language given a set of positive data. Chandlee et al. (2019) propose the Bottom-Up Factor Inference Algorithm (BUFIA), which is capable of discovering the most general forbidden subfactors represented in model-theoretic forms and guarantees full coverage of the data space.

This paper shows how an AR graph is abstracted into a mathematical representation using Model Theory, and implements a specific version of BUFIA for ARs (BUFIA-AR). BUFIA-AR is evaluated on a case study of Hausa, a tonal language primarily spoken in West Africa. In our case study, orthographic forms in a Hausa dictionary (Awagana et al., 2009) were converted into ARs using Model Theory, and BUFIA-AR was applied to this dataset. The experiment discovered 26 distinct ARs among 664 surface-true Hausa monomorphemic forms and identified seven AR structures (syllable number ≤ 3, tone number ≤ 3) absent from the data, which constitute constraints in the tonal grammar. These computationally-found constraints either align with previously proposed analyses or provide more specific generalizations that better account for the entire dataset. Although the current implementation does not encode the difference between light and heavy syllables or mark word boundaries, both of which have been reported to influence tonal mapping in Hausa (Newman, 2002), the results show that BUFIA-AR is capable of identifying tonotactic patterns over ARs. Future research will further develop BUFIA-AR to incorporate additional representational details. It is hoped that this will not only provide deeper insights into tonotactic patterns in computational contexts but also demonstrate the efficacy of ARs for tonal language data.[1]

2 Preliminaries

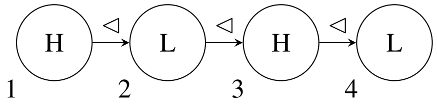
Let Σ be a finite alphabet of symbols and Σ^* the set of all strings over Σ. Let $|w|$ indicates the length of $w \in \Sigma^*$. For two strings w and v, wv refers to their concatenation, and for a set $L \subseteq \Sigma^*$ of strings and a string w, by wL we denote $\{wv \mid v \in L\}$. A string u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

[*] The author thanks Jeff Heinz, Ellen Broselow, and Thomas Graf for their valuable feedback and discussions. The author also acknowledges support from the Institute for Advanced Computational Science at Stony Brook University.

[1] All code from this study is accessible at https://github.com/lihan829/AR_learning.

Proceedings of *AMP 2023/2024*

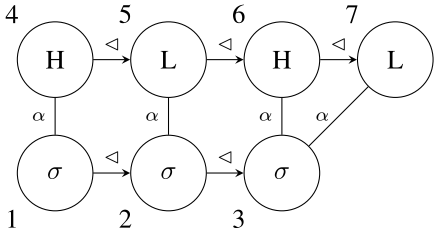
A string can be represented model-theoretically, which contains two parts: the Domain D and a finite set of n-ary relations $\mathcal{R} = \{R_1, R_2, \ldots, R_i\}$, also known as the *signature* of the model. The signature $\{R_1, R_2, \ldots, R_i\}$ includes different types of relations: unary relations representing symbols in the string, and binary relations, such as a successor relation (denoted by $\lhd$), which indicate the linear order of elements. For example, consider a string *HLHL* formed from the alphabet $\{H, L\}$ as shown in Figure 1. The domain D contains a total of four elements $\{1, 2, 3, 4\}$. The relations R_H and R_L are unary relations indicating the positions of H and L tones, respectively. The binary successor relation $\lhd = \{\langle 1,2\rangle, \langle 2,3\rangle, \langle 3,4\rangle\}$ indicates the order of elements.



$$M_t = \{D \mid R_H, R_L, \lhd\}$$
$$D_t = \{1, 2, 3, 4\}$$
$$R_H = \{1, 3\}$$
$$R_L = \{2, 4\}$$
$$\lhd = \{\langle 1,2\rangle, \langle 2,3\rangle, \langle 3,4\rangle\}$$

Figure 1: Tonal string model *HLHL* and its model signature

Such string models as shown in Figure 1 are referred to as *one-dimensional* models, or *conventional* models since the elements are ordered linearly along a single axis (Chandlee et al., 2019; Strother-Garcia et al., 2017; Vu et al., 2018). ARs, on the other hand, are *unconventional* since they consist of at least two strings: a tonal string (representing the tone sequence) and a timing string (representing the tone-bearing unit (TBU) sequence). Both strings are one-dimensional models preserving successor relations. What distinguishes the AR model from a string-based model is the addition of a binary relation (denoted by α), which connects elements from one tier to another, mirroring the association lines of an AR. Therefore, an AR model is the union of the tonal string and the timing string along with the binary association relation that goes between tiers. Let *k* denotes the size of an AR structure such that *k* is the summation of the number of syllable, tones, and association lines $k = |\sigma| + |H| + |L| + |\alpha|$. Figure 2 shows an AR model of a trisyllabic four-tone structure with the first syllable being node 1 and the first tone node 4.



$$M = \{D \mid R_H, R_L, R_\sigma, \lhd, \alpha\}$$
$$D = \{1, 2, 3, 4, 5, 6, 7\}$$
$$R_\sigma = \{1, 2, 3\}$$
$$R_H = \{4, 6\}$$
$$R_L = \{5, 7\}$$
$$\lhd = \{\langle 1,2\rangle, \langle 2,3\rangle, \langle 4,5\rangle, \langle 5,6\rangle, \langle 6,7\rangle\}$$
$$\alpha = \{\langle 1,4\rangle, \langle 4,1\rangle, \langle 2,5\rangle, \langle 5,2\rangle, \langle 3,6\rangle, \langle 6,3\rangle, \langle 3,7\rangle, \langle 7,3\rangle\}$$

Figure 2: AR model for a word $\acute{\sigma}\grave{\sigma}\hat{\sigma}$

An AR can be expanded in the same way a string can be extended by adding more characters to the end. Given an AR of size k, it can be expanded through three possible modifications: the addition of a tone to the tonal tier (following the Obligatory Contour Principle, OCP); the addition of a TBU to the timing tier; or the addition of an association between the two tiers (following the No Crossing Constraint, NCC). We define Structure *A* as an *immediate superfactor* of Structure *B* if *A* is expanded from *B* by a single addition of a tone, TBU, or association. Consequently, the size of *A* is *k+1* if the size of *B* is *k*. In Section 3, we will discuss the implementation details of these modifications in depth.

3 Bottom-Up Factor Inference Algorithm over AR

Chandlee et al. (2019) discuss two directions of grammatical inference, a top-down and a bottom-up direction, and argue the latter one is more efficient and guaranteed to find the constraints in a language. In the bottom-up learning fashion, the learner traverses a space of logically-possible structures from the most general (usually the empty structure) to the more specific, and identify its presence in the positive data. If a structure is missing, BUFIA will identify the current structure as a constraint in the language, and will no

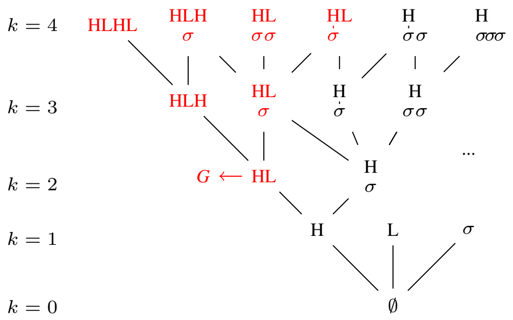


Figure 3: Bottom-Up Learning over ARs

longer consider any of its superfactors, i.e., structures that *contains* the constraint. This algorithm guarantees to find the most general grammar which covers the given data.

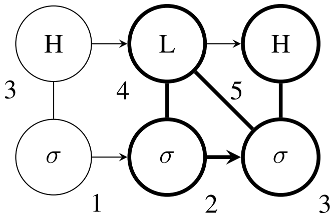
It should be mentioned that the implementation of BUFIA depends on the precise representation. Here we will use tonal pattern over ARs as examples. As mentioned in Section 2, a superfactor of an AR can be generated from three potential changes: addition of a tone, or a TBU, or an association between the two tiers. As can be seen from Figure 3, when $k = 1$, the structure can be a floating H, a floating L, or a floating syllable. Take the floating H for instance, its superfactors ($k = 2$) include a floating HL sequence, and a floating H with a floating syllable. If the HL sequence is found missing from the data D, the algorithm will cease generating any superfactors and add HL into the grammar as a constraint forbidden by the language (labeled in red).

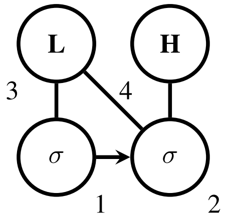
Two functions `NextAR` and `Contain` are essential parts for BUFIA-AR to infer grammars. In the rest of the section, we will define these functions in terms of ARs.

3.1 *Contain* The function `Contain` is based on the definition of *restriction* and *subfactor* given by Chandlee et al. (2019:p.94).

Definition 1. *$A = \langle D^A; \lhd, R^1_A, \dots, R^n_A\rangle$ is a restriction of $B = \langle D^B; \lhd, R^1_B, \dots, R^n_B\rangle$ iff $D^A \subseteq D^B$ and for each m-ary relation R_i, we have $R^i_A = \{(x_1, \dots, x_m) \in R^i_B \mid x_1, \dots, x_m \in D^A\}$.*

In other words, a restriction can be understood as a substructure in which no relations have been altered. Consider the model in (1) as an example. The bold portion constitutes a restriction of (1) if it is extracted without modifying any of its relations.

(1)


(2)


Using the concept of restriction, we can determine whether an AR A is contained within another AR B by checking if there exists a restriction of B, denoted as B', such that there is a relation-preserving bijection between A and B'. If such a bijection exists, A is identified as a subfactor of B, and B as a superfactor of A.

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

[2] Chandlee et al. (2019) requires a structure needs to be "connected". For ARs, connectedness is required on both tiers

In other words, if all relations that hold in A also hold in a corresponding way in B, then A is considered a subfactor of B, and B is considered a superfactor of A. For example, (1) can be identified as a superfactor of (2) since there exists a restriction, the bold portion, matches (2) in the sense that all relations in that restriction are consistently preserved in (2), with a function relabeling the indices. Trivially, an empty structure is contained by all ARs, but only contains itself.

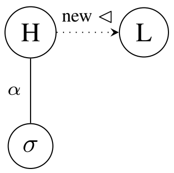
3.2 *Next AR* When the algorithm identifies the presence of a given structure, the `NextAR` function will generate its immediate superfactors. As mentioned in Section 4.2, this process can be achieved by an AR can be expanded using three operations: `addTone`, `addTBU`, or `addAssociation` only when there are floating units in the AR, that is, there are elements in the successor relation but not in association relation.

Definition 3. For AR = $\{D \mid R_H, R_L, R_\sigma, \lhd, \alpha\}$, a unary relation $R_i \in \{R_H, R_L, R_\sigma\}$ is considered *floating* if $R_i \in \lhd \wedge R_i \notin \alpha$.

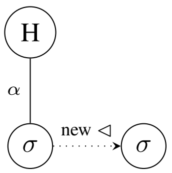
$$\texttt{NextAR(k)} = \begin{cases} \texttt{addTone} \cup \texttt{addTBU} & \text{if not } \texttt{containFloat(k)} \\ \texttt{addTone} \cup \texttt{addTBU} \cup \texttt{addAssociation} & \text{if } \texttt{containFloat(k)} \end{cases}$$

First, the function `containFloat` checks for the presence of a floating unit in the AR. The `addAssociation` function applies only to ARs that contain floating units. If an AR does not contain any floating units, the `NextAR` function will generate its superfactors by either adding a new tone (obeying the OCP, as in (3)) or a new TBU (as in (4)). If there are any floating units in the structure, in addition to new tone and new TBU, new associations will be generated and it will only be considered valid if and only if it does not violate No-Crossing Constraint (NCC) (Goldsmith, 1976; Coleman and Local, 1991; Hammond, 1988). Formally, an association $(i', j') \in R_\alpha$ is only valid if $(i' > i \wedge j' < j) \vee (i' < i \wedge j' > j)$ (see also in Bird and Klein (1990); Jardine (2013); Jardine and Heinz (2015)). Following these principles, there are three ways to form possible new associations: the floating syllable connects to the associated syllable as in (5); the floating tone connects to the associated syllable as in (6); the floating tone connects to a floating syllable as in (7).

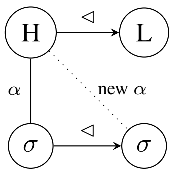
(3)



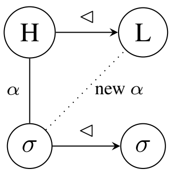
(4)



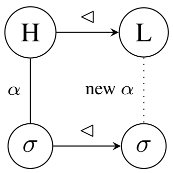
(5)



(6)



(7)



In summary, by using these two function `Contain` and `NextAR`, we systematically generate all logically possible AR structures in a bottom-up fashion while checking their presence in the positive data. The next section provides a case study to test the implementation of BUFIA-AR.

4 Case Study: Hausa

4.1 *Data Preparation* In this case study, we applied BUFIA-AR to a small corpus of Hausa (Awagana et al., 2009), a tonal language spoken in West Africa, and evaluated the model by comparing the constraints it identified with previously established linguistic generalizations. Hausa has two contrastive tones, H and L, which can combine to form a falling contour (HL) when they occur on a single heavy syllable. However, monosyllabic LH contours and three-tone contours are not permitted. Previous research on Hausa tones has established the generalizations and rules summarized in Table 1 (Newman, 2002; Leben, 1971; Zoll, 2003).

The corpus contains 1,668 core meaning-word pairs, from which we filtered out loanwords and polymorphemic forms, leaving 664 monomorphemic words. These orthographic forms were then converted into ARs, and BUFIA-AR was applied to the resulting data. A Python script was developed to process the 664 string representations, yielding 26 distinct ARs. Table 2 shows the data representation conversion. A

but not between tiers. This flexibility allows the structures without any association lines, such as a sequence of floating tones.

Generalizations/Constraints	**Explanation**
***Rise**	No monosyllabic rising
***3T-Contour**	No three-tone contour
Final-Contour	No initial HL contour for polysyllabic words
***LL, Lapse ≫ Clash**	Avoid LL spreading

Table 1: Phonological Generalizations in Hausa

pair (t, s) in the shorthand coding represents the association line between the tone indexed by t and the TBU indexed by s.

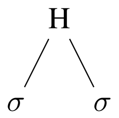
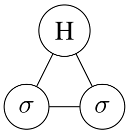
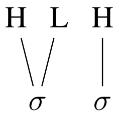
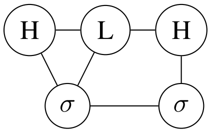
Orthography	**Tones**	**AR**	**AR Model**	**Shorthand Coding**
ƙásáa	HH			(H, $[(1,1),(1,2)]$)
bâutáa	FH			(HLH, $[(1,1),(2,1),(3,2)]$)

Table 2: Orthographic Forms and their corresponding representations

4.2 *Results* Table 3 lists the seven tonotactic constraints found by BUFIA-AR when the syllable size s is no greater than 3 and the tone size t is no greater than 3. No constraints were found when $t = 1$. When $t = 2$, three constraints were identified: one monosyllabic form (3a) and two disyllabic forms (3b-3c). When $t = 3$, four trisyllabic forms (3d-3g) were found. The following section will compare the previously established linguistic rules listed in Table 1 with the algorithm-detected constraints (hereafter referred to as "BUFIA constraints") and discuss the extent to which they match or differ from each other.

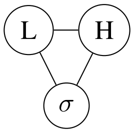
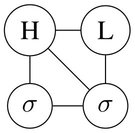
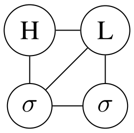
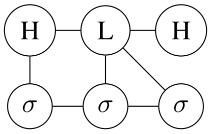
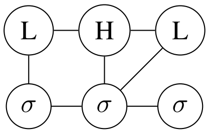
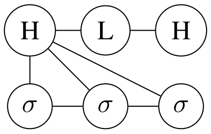
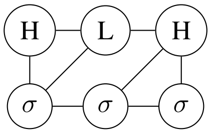
$t = 2$ $s < 3$	 **(a)**	 **(b)**	 **(c)**	
$t = 3$ $s = 3$	 **(d)**	 **(e)**	 **(f)**	 **(g)**

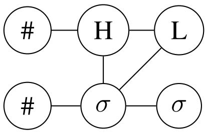
Table 3: BUFIA-AR Identified Constraints for Hausa Tones

The most fundamental rule in Hausa is that monosyllabic forms do not allow a rising tone (***Rise**) or a three-tone contour (***3T-Contour**). Correspondingly, BUFIA-AR identified a monosyllabic form connected to an LH sequence, as shown in (3a), which aligns with the prediction of *Rise. Furthermore, Constraint (3a) also corresponds to ***3T-Contour**, as any monosyllabic three-tone contour, whether LHL or HLH, necessarily contains (3a). Since the algorithm returns only the most generalized constraint, any superfactors containing this constraint are neither generated nor examined. As an interim summary, we can consider the algorithm to have accurately captured the monosyllabic tonal patterns.

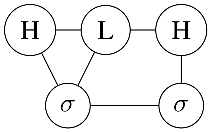
Another frequently discussed pattern in Hausa is **Final-Contour**, which forbids initial HL contours in polysyllabic words. One could expect BUFIA-AR to return a structure like (8). Although edge markers have not yet been incorporated into AR modeling in this version of BUFIA-AR, two constraints were reported that prohibit an HL contour before either an L-toned syllable (3c) or an H tone spreading over two syllables (3g). These two constraints, though not explicitly position-specified, still capture cases where the pattern occurs at the beginning of a word.

One important point to note is that the discrepancy between (3c) and (3g) suggests a permissible form: an HL contour followed by an H tone that spreads over only one syllable. In other words, structures such as (9) and (10), which BUFIA-AR does not label as constraints, could appear in the positive data despite violating **Final-Contour**. Indeed, a reverse search found seven words matching these two structures, as shown in Example (11). This result suggests that **Final-Contour** needs to be further refined to fully account for the dataset.

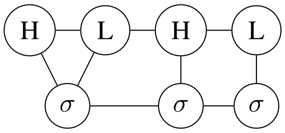
(8)



(9)



(10)



(11) Hausa words contain an initial HL

ƙyânwáa	"the cat"
kûnnée	"the ear"
tsûmmáa	"the handkerchief or rag"
sâiwáa	"the root"
jînƙái	"the pity"
bâutáa	"to worship, to obey"
yânyáawàa	"the fox (fennec of Sahara)"

Lastly, ***LL** indicates the avoidance of LL sequences. However, this is not a strict ban. Newman (2002:p. 606) noted that LL exceptions are "relatively uncommon and mostly represent identifiable or presumed loanwords." Zoll (2003) uses Lapse $\gg$ Clash to generalize the preference for H-tone spreading over L-tone spreading. Correspondingly, BUFIA-AR did not report any constraint when $t = 1$ and $s = 2$, confirming that there are cases (as in (12)) where disyllabic words contain LL sequences. In fact, four words in our dataset exhibit LL sequences. Notably, three out of these four words are question words, suggesting that morphological category may play a role in tone mapping.

(12)

màcè	"female"
yàayàa	"how"
yàushè	"when"
wànè	"which"

Additionally, while it has been commonly reported that in Hausa, when a two-tone HL sequence maps onto trisyllabic words, the preferred pattern is H.H.L ($\acute{\sigma}\acute{\sigma}\grave{\sigma}$) rather than *H.L.L ($*\acute{\sigma}\grave{\sigma}\grave{\sigma}$); and LH results in L.H.H ($\grave{\sigma}\acute{\sigma}\acute{\sigma}$) rather than *L.L.H ($*\grave{\sigma}\grave{\sigma}\acute{\sigma}$) (Zoll, 2003; Leben, 1971; Newman, 2002), the BUFIA constraint (3d) suggests that both disfavored patterns are actually present in the dataset. In fact, the L.L.H pattern is not uncommon: 16 words were found to follow this pattern. Based on (3d), we can revise ***LL** as follows: LL spreading is prohibited when two L-toned syllables occur between two H-toned syllables.[3]

In summary, this section reports the results of implementing BUFIA to find tonotactic patterns over ARs and compares computationally attested constraints with linguistically reported constraints. It is found that

[3] More precisely, (3d) states that no L tone can occupy two adjacent syllables while being wedged between one H-toned syllable and a second H tone, whose association does not have to be explicitly specified. Theoretically, the second H tone could either be linked to the rightmost syllable connected to the L tone or to a fourth syllable independently. However, the first option is unavailable since LH cannot dock onto the same syllable.

BUFIA can effectively learn tonotactic patterns over AR. Three outcomes emerge regarding the matching between BUFIA constraints and linguistic rules. First, the two coincide, indicating that BUFIA accurately captures exactly what previous linguistic rules formalize. Second, BUFIA constraints are more restrictive and specific than linguistic constraints without overgeneralizing the patterns. Lastly, the algorithm also uncovers some patterns, usually of a larger size, that have not been reported in any other literature.

5 Conclusion

This paper shows the first attempt to learn tonotactic patterns over autosegmental representations using a bottom-up learning algorithm. Through a case study of Hausa, 26 distinct ARs among 654 surface-true Hausa words were found, and tonotactic constraint grammar consist of seven constraints ($\leq$ three syllables and three tones). Furthermore, by comparing BUFIA constraints and linguistic rules, we found sometimes the two matches but more often BUFIA constraints were more specific in order to account for the entire dataset. These results show the feasibility of tonotactic grammar inference over ARs and extends the advantaged of using algorithm to enhance previous generalizations.

As mentioned before, edge markers and the syllable weight are not coded in the ARs so far, which will be the focus of the future work. Besides, the pipeline used in the current study, from data preparation, orthography-to-AR conversion, then to the implementation of BUFIA-AR, can serve as a useful tool to explore tonotactic constraints in other languages. It would be helpful to investigate the psychological reality of the constraints found by BUFIA and seek external evidence to contribute to the understanding of cognition and perception related to these languages.

References

Awagana, A., Wolff, H. E., and Löhr, D. (2009). Hausa. In Haspelmath, M. and Tadmor, U., editors, *World Loanword Database*. Max Planck Institute for Evolutionary Anthropology, Leipzig.

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

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, Toronto, Canada. Association for Computational Linguistics.

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

Goldsmith, J. A. (1976). *Autosegmental Phonology*. PhD thesis, Massachusetts Institute of Technology.

Hammond, M. (1988). On Deriving the Well-formedness Condition. *Linguistic Inquiry*, 19(2):319–325.

Hyman, L. M. (2011). Tone: Is it Different? *The Handbook of Phonological Theory*, pages 197–239.

Jardine, A. (2013). Logic and the Generative Power of Autosegmental Phonology. In *Proceedings of the Annual Meetings on Phonology*.

Jardine, A. (2017). The Local Nature of Tone-Association Patterns. *Phonology*, 34(2):363–384.

Jardine, A. and Heinz, J. (2015). A Concatenation Operation to Derive Autosegmental Graphs. In *Proceedings of the 14th Meeting on the Mathematics of Language (MoL 2015)*, pages 139–151.

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

Leben, W. R. (1971). The Morphophonemics of Tone in Hausa. In *Papers in African Linguistics*, pages 201–218.

Newman, P. (2002). *The Hausa Language: An Encyclopedic Reference Grammar*, volume 122. Yale University Press, New Haven, US.

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

Strother-Garcia, K., Heinz, J., and Hwangbo, H. J. (2017). Using Model Theory for Grammatical Inference: a Case Study from Phonology. In *International Conference on Grammatical Inference*, pages 66–78. PMLR.

Vu, M. H., Zehfroosh, A., Strother-Garcia, K., Sebok, M., Heinz, J., and Tanner, H. G. (2018). Statistical Relational Learning with Unconventional String Models. *Frontiers in Robotics and AI*, 5:76.

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