

# The expressivity of segmental phonology and the definition of weak determinism

Adam G. McCollum, Eric Baković, Anna Mai, Eric Meinhardt  
University of California, San Diego

## Abstract

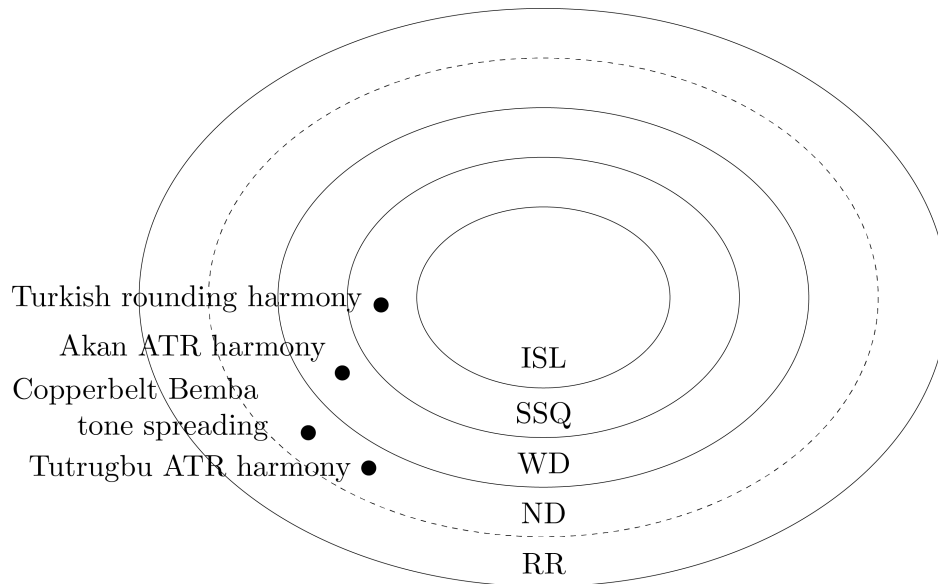
[Jardine \(2016\)](#) asserts a computational distinction between segmental and tonal phonology, arguing that certain tonal patterns require the strictly more expressive computational power of non-deterministic regular functions while segmental patterns require at most the power of weakly deterministic regular functions. We advance two claims bearing on this assertion. Empirically, we show that non-deterministic segmental patterns are in fact attested, focusing on the vowel harmony pattern found in Tutsi (McCollum and Essegbey 2018) and citing several others. We submit that these patterns are non-deterministic in exactly the same essential way as the tonal patterns discussed by [Jardine \(2016\)](#). Formally, we show that the definition of weakly deterministic regular functions offered by [Heinz and Lai \(2013\)](#) is incapable of distinguishing between non-deterministic patterns and the less complex weakly deterministic patterns it is intended to delimit. We offer a revised definition of weakly deterministic functions that makes the correct distinctions, subsuming the conditions imposed by [Heinz and Lai \(2013\)](#) under a proposed definition of ‘interaction’ between composed functions. We also conjecture that the inverse relationship between complexity and observed frequency is explainable by domain-general principles of learnability rather than a categorical cap on the complexity of a phonology-specific learning mechanism, contra [Heinz and Idsardi \(2011, 2013\)](#).

## 1 Introduction

Phonological theories make typological claims, predictions about what are and what are not possible phonological patterns. Much work has been devoted to questions about the expressivity of a given phonological theory. Is the theory powerful enough to correctly predict the existence of a given attested pattern? — and if not, how can the theory’s expressive power be appropriately extended? Is the theory so powerful that it incorrectly predicts the existence of an unattested, potentially ‘pathological’ pattern? — and if so, how can the theory’s expressive power be appropriately constrained?

Recent work has approached this question from the perspective of formal language theory, aiming to characterize particular attested or unattested phonological patterns in terms of well-defined, theory-independent computational classes of string sets (formal languages) or mappings between string sets (string functions). The larger goal of this work is to formally delimit the boundary between possible (*qua* attested) and impossible (*qua* unattested) patterns. [Heinz \(2011a, b, 2018\)](#), [Heinz and Lai \(2013\)](#), and [Jardine \(2016\)](#), among others, have advanced the broad hypothesis that phonological patterns are SUBREGULAR, meaning that they can be characterized by computational classes that are strictly less expressive than the class of regular string relations, which characterizes the expressivity of ordered rewrite rules ([Johnson 1972](#), [Kaplan and Kay 1994](#)). Figure 1 summarizes some subregular function classes and points to some relevant examples, including the example that is our empirical focus in this paper, Tutsi ATR harmony.

[Chandlee \(2014\)](#) and [Heinz \(2018\)](#) note that the vast majority of phonological patterns appear to be contained within the innermost, least expressive classes of the subregular hierarchy. However, there are some patterns that are relatively more complex, and [Heinz and Lai \(2013\)](#) and [Jardine \(2016\)](#) contend that the most complex segmental patterns are WEAKLY DETERMINISTIC. These types of patterns include prototypical bidirectional harmony processes. In contrast, [Jardine \(2016\)](#) demonstrates that a number of tonal patterns are more complex than this. These NON-DETERMINISTIC patterns all exhibit UNBOUNDED CIRCUMAMBIENCE, a dependence on information (e.g. triggers of spreading and blocking conditions) that are an unbounded



**Figure 1** Some regions of the subregular hierarchy of string-to-string mappings, with phonological examples. Examples of input strictly local functions (ISL) include nasal place assimilation and metathesis (Chandlee 2014); examples of subsequential functions (SSQ) include unidirectional harmony (Heinz and Lai 2013) and dissimilation (Payne 2017); examples of weakly deterministic functions (WD) include bidirectional harmony (Heinz and Lai 2013); examples of non-deterministic functions (ND) include unbounded circumambient processes (Jardine 2016); examples of regular relations (RR) include optional iterativity (Heinz 2018).

distance from the spreading target in both directions. In what are perhaps more familiar terms, these more complex patterns are both unbounded and non-myopic (Wilson 2003, Walker 2010).

We advance two claims in this paper, one empirical and the other formal. Empirically, we show that non-deterministic segmental processes are in fact attested, focusing in §3 on the vowel harmony pattern found in Tutrugbu (McCollum and Essegbey 2018) and citing several others in §5.1. The key property of the Tutrugbu pattern can be seen in the comparison between the pairs of forms in (1). ATR spreads leftward from roots to prefixes, and [−high] prefix vowels undergo harmony (1a,b) — unless the initial prefix vowel is [+high], in which case harmony is blocked by [−high] vowels (1c,d). We demonstrate that this pattern is non-deterministic in exactly the same way as the tonal patterns discussed by Jardine (2016) are.

- (1) Tutrugbu harmony
- |    |            |                   |    |         |               |
|----|------------|-------------------|----|---------|---------------|
| a. | a-tí-ba-bá | ‘3S-NEG-FUT-come’ | c. | ɪ-ba-bá | ‘1S-FUT-come’ |
| b. | e-tí-be-ċē | ‘3S-NEG-FUT-grow’ | d. | ɪ-ba-ċē | ‘1S-FUT-grow’ |

Formally, we show that the definition of weakly deterministic regular functions offered by Heinz and Lai (2013) is incapable of distinguishing between non-deterministic patterns like that in Tutrugbu and less complex weakly deterministic patterns. We offer a revised definition of weakly deterministic functions that makes the correct distinctions, subsuming the conditions imposed by Heinz and Lai (2013) under a definition of ‘interaction’ that we introduce and illustrate in §4.3.

The paper is structured as follows. In §2, we describe and give examples for three germane complexity classes within the subregular hierarchy. We examine ATR harmony in Tutrugbu in §3 and demonstrate that this pattern is non-deterministic for the same essential reason as the tonal patterns discussed by Jardine (2016): because it is unbounded circumambient. In §4 we demonstrate the insufficiency of the existing definition of weak determinism, propose a new definition constrained by a notion of ‘interaction’, and then discuss

how this notion properly separates weakly deterministic from non-deterministic patterns. In §5 we discuss other examples of unbounded circumambient segmental patterns, and the implications of these findings for explaining the typology of phonological complexity. We conclude the paper in §6.

## 2 Hypotheses concerning phonological complexity

The computational complexity of phonological grammars comprising ordered SPE-style rewrite rules (Chomsky and Halle 1968) corresponds to the class of regular string relations (Johnson 1972, Kaplan and Kay 1994). However, many phonological input-output maps do not require the full expressivity of regular relations. Synchronically attested metathesis, partial reduplication, and dissimilation patterns are all SUBSEQUENTIAL (Chandlee et al. 2012, Chandlee and Heinz 2012, Payne 2017). Moreover, Chandlee (2014)’s analysis of the PBase (Mielke 2007), a database of phonological processes, shows that most are describable using a highly restricted subclass of the subsequential functions.<sup>1</sup> Finally, the vowel harmony patterns analyzed in Nevins (2010) have also been argued to be subsequential (Gainor et al. 2012). For this reason, Gainor et al. (2012) argue that a more restrictive hypothesis better captures the minimum computational expressivity necessary to describe phonological patterns: the SUBSEQUENTIAL HYPOTHESIS (see also Heinz 2018).

Under the subsequential hypothesis, the most complex computational machinery necessary to model phonological patterns is that of subsequential functions (Mohri 1997). However, Heinz and Lai (2013) demonstrate that bidirectional stem-controlled and dominant-recessive vowel harmony patterns are not subsequential, and instead propose that phonological patterns are at most weakly deterministic, a class they define in terms of a restriction on compositions of two subsequential functions.

Significantly, Jardine (2016) discusses a number of tonal patterns that are neither subsequential nor weakly deterministic, arguing that subsequential and weakly deterministic functions properly include all attested *segmental* patterns, but that these tonal patterns require strictly more expressivity. Jardine shows that the class of unbounded circumambient tonal patterns require non-deterministic mappings. For this reason, Jardine (2016) argues that there is a computational difference between tonal and segmental phonology: while segmental phonology may be restricted to the class of subsequential functions and their weakly deterministic compositions, tonal phonology requires the expressive power of non-determinism.

While regular relations and their subclasses can be represented in many ways, including ordered rewrite rules, suitably restricted Optimality Theory grammars (Riggle 2004), and finite-state transducers (FSTs), we represent such mappings as FSTs in this paper for two related reasons: much work characterizing regular relations and their subclasses uses this formalism, and consequently, existing work on the complexity of phonology does as well. FSTs represent string-to-string functions in a way that emphasizes incremental calculation, proceeding symbol by symbol from one end of the input string to the other and restricting the amount of information about the observed prefix of the input that can be ‘remembered’ while deciding what to output at each step. Such differences in the amount and type of information a transducer is able to store can define more vs. less expressive classes of transducers, which can in turn be used to describe differences in the amount and type of information required to express different phonological patterns. For more detailed introductions to finite state transducers and their relation to phonology, we refer the reader to Jardine (2016), Kaplan and Kay (1994), and Mohri (1997). In the next three subsections we review definitions, key properties, and examples of subsequential, weakly deterministic, and non-deterministic patterns.

### 2.1 Subsequentiality

A subsequential finite state transducer (FST)  $\tau$  can be defined by seven parameters:

- $Q$ , a finite set of states,
- $\Sigma$ , a finite set of symbols representing the input alphabet,

<sup>1</sup>As Chandlee (2014) notes, the PBase is a set of analyses, not a typologically representative set of cross-linguistic data — but it is the closest thing to such a typologically representative set that is currently available.

- $\Delta$ , a finite set of symbols representing the output alphabet,
- $q_0 \in Q$ , a distinguished initial state,
- $F \subseteq Q$ , a set of final or accepting states,
- $\delta: Q \times \Sigma \rightarrow Q$ , a state transition function,
- $\omega: Q \times \Sigma \rightarrow \Delta^*$ , an output function,<sup>2</sup> and
- $\sigma: F \rightarrow \Delta^*$ , a partial function that assigns strings to final states.

Intuitively, a subsequential transducer is an FST whose incremental behavior is always DETERMINISTIC. That is, state transitions and output strings are deterministic functions of the current state and input symbol, and the string-to-string function  $\tau: \Sigma^* \rightarrow \Delta^*$  defined by a subsequential FST is also a deterministic mapping.

Subsequential FSTs can be divided into two partially overlapping classes based on the directionality of their computation (Chandlee 2014, Ch. 3). Left-subsequential FSTs read input strings from left to right, while right-subsequential FSTs read input strings from right to left. When using subsequential FSTs to model vowel harmony patterns, this distinction in directionality of computation maps intuitively onto the directionality of the harmony pattern. Canonical progressive harmony patterns are modeled with left-subsequential FSTs, and canonical regressive harmony patterns are modeled with right-subsequential FSTs.

In Figure 2 (see the Appendix for this and other figures), we provide an example of a left-subsequential FST which models the progressive rounding harmony pattern found in Turkish; some data exhibiting this pattern are shown in (2). Rounding spreads left to right from roots to suffixes; [+high] vowels undergo harmony (2f,g) while [−high] vowels block harmony (2h–j).

(2) Turkish progressive rounding harmony

	[−round] root	gloss		[+round] root	gloss
a.	ip-in-i	‘rope-2S.POSS-ACC’	f.	jyz-yn-y	‘face-2S.POSS-ACC’
b.	el-in-i	‘hand-2S.POSS-ACC’	g.	køj-yn-y	‘village-2S.POSS-ACC’
c.	ip-ler-i	‘rope-PL-ACC’	h.	jyz-ler-i	‘face-PL-ACC’
d.	el-ler-i	‘hand-PL-ACC’	i.	køj-ler-i	‘village-PL-ACC’
e.	ip-in-de	‘rope-2S.POSS-LOC’	j.	jyz-yn-de	‘face-2S.POSS-LOC’

In (3)–(5) we provide a ‘running tape’ representation of the mapping performed by the FST shown in Figure 2, using the word jyz-yn-de (2j) to illustrate both rounding harmony and its blocking. To begin: since harmony is stem-controlled in the language, the FST outputs all segments to the left of the root-suffix boundary (symbolized as  $\backslash$ ) without modification (3). To be clear, the FST does not as its first operation map the whole substring  $\times jyz \backslash$  to itself; it processes each input symbol incrementally. We refrain from showing each of the first five steps separately, and make similar abbreviatory choices in later examples.

(3) Turkish left-subsequential FST tape, stage 1: faithful root map

input	$\times$	j	y	z	$\backslash$	i	n	d	e	$\times$
output	$\times$	j	y	z	$\backslash$					

A suffix with a [+high] vowel is then introduced as input, and since the rounding value of the root vowel is known, the output function emits a vowel matching the rounding value of the root in (4). (Again, the two symbols of the suffix are technically processed incrementally. Since consonants do not participate in the harmony process, they are always mapped faithfully, and we thus refrain from showing these steps separately.)

<sup>2</sup>A transducer with an output function that can map each input symbol to a string of output symbols is a *word* (or *asynchronous*) transducer; if  $\omega$  can only map any given input symbol to a single output symbol, it is a *letter* (or *synchronous*) transducer. Almost any word transducer can be converted to a letter transducer (Roche and Schabes 1997, Ch. 1). We make use of letter transducers everywhere except when analyzing Copperbelt Bemba in §4.2.

- (4) Turkish left-subsequential FST tape, stage 2: [+high] suffix map

input	×	j	y	z	√	i	n	d	e	×
output	×	j	y	z	√	y	n			

A suffix with a [−high] vowel is then introduced as input. This vowel is mapped faithfully, since [−high] vowels block rounding harmony. The end of the word is then reached and the computation ends, as in (5).

- (5) Turkish left-subsequential FST tape, stage 3: [−high] suffix map; end of computation

input	×	j	y	z	√	i	n	d	e	×
output	×	j	y	z	√	y	n	d	e	×

In this way, progressive harmony patterns can be modeled with left-subsequential FSTs — and in similar fashion, regressive harmony patterns can be modeled with right-subsequential FSTs. In the next subsection, we define and discuss weakly deterministic patterns, which require an appropriately restricted composition of left- and right-subsequential FSTs.

## 2.2 Weak determinism

The ordering of rewrite rules  $\rho_1 < \rho_2 < \dots < \rho_n$  in an SPE-style analysis corresponds to the ordered composition of associated string-to-string functions  $\phi_{\rho_n} \circ \dots \circ \phi_{\rho_2} \circ \phi_{\rho_1}$ . While the composition of any two subsequential functions going in the same direction can only yield another subsequential function (Mohri 1997), a function defined by the composition of subsequential functions going in opposite directions can capture any regular relation, as detailed in the next subsection. Weakly deterministic regular functions (the class ‘WD’ in Figure 1) are described by Heinz and Lai (2013) as those FSTs that can be defined as the composition of two subsequential functions going in opposite directions such that the two functions do not use an intermediate alphabet containing symbols not present in the input alphabet.<sup>3</sup>

- (6) A regular string-to-string function  $\tau : X^* \rightarrow Y^*$  is weakly deterministic iff there exist two subsequential functions  $I : X^* \rightarrow X^*$  and  $O : X^* \rightarrow Y^*$  such that  $O$  is either left- or right-subsequential,  $I$  is the other, and  $\tau = O \circ I$ .

This definition of weakly deterministic regular functions is designed to pick out functions with two notable properties. First, this class of functions is able to capture bidirectional patterns that no right- or left-subsequential function can on its own. Second, the increased expressivity of this class is constrained by a restriction: no extra intermediate symbols, ensured by the condition that the output alphabet of the first (‘inner’) subsequential function,  $I$ , is the same  $X^*$  as the input alphabet of both functions. Without such a restriction, the composition of two subsequential functions is, in general, not subregular (Elgot and Mezei 1965): in fact, it is fully non-deterministic. In a weakly deterministic function, the behavior of the second (‘outer’) function does not depend on any form of ‘mark-up’ deposited by the first function into the intermediate representation, and the restrictiveness of this class of patterns is intended to follow from the definition’s prohibition of an intermediate alphabet with extra symbols. *Without* this restriction, the inner function could use additional symbols to effectively provide unbounded lookahead for the outer function, allowing the outer function to behave deterministically given this marked-up version of the input string.

Heinz and Lai (2013) show that weakly deterministic functions are capable of describing bidirectional harmony patterns. Figures 3 and 4 show two subsequential FSTs, the composition of which models bidirectional vowel harmony in Akan (Dolphyne 1988). The pattern in Akan differs from that of Turkish in two key respects: the harmonizing feature and directionality. In Akan, the feature which participates in harmony is advanced tongue root (ATR), and affixes on *both* sides of a root agree in ATR with the root (7).<sup>4</sup> Specifically, observe that the allomorphs of the 3S, FUT, and 3S.OBJ affixes in (7) all agree with the root for ATR.

<sup>3</sup>The full definition includes one additional condition we set aside for clarity of exposition here; our goal here is to communicate the intuition behind Heinz and Lai’s definition. We review their exact definition later in §4.

<sup>4</sup>But note that Turkish lacks prefixes, and is thus not able to exhibit bidirectional stem control. See Baković (2000) for discussion.

## (7) Akan bidirectional ATR harmony

	[-ATR] root	gloss		[+ATR] root	gloss
a.	ɔ-tsɪɾɛ-ɪ	‘3S-show-3S.OBJ’	c.	o-fiti-i	‘3S-pierce-3S.OBJ’
b.	ɔ-bɛ-tʊ-ɪ	‘3S-FUT-throw-3S.OBJ’	d.	o-be-tu-i	‘3S-FUT-dig-3S.OBJ’

In (8)–(13) we provide a running tape representation of the function performed by the FSTs in Figures 3 and 4, using the word [o-fiti-i] (7c). We begin with the left-subsequential function in Figure 3.<sup>5</sup> The first symbol after the left-word edge that the FST encounters is a prefix vowel. At this point, the ATR value of the root is not known, so the vowel is output faithfully, as shown in (8).

## (8) Akan inner, left-subsequential FST tape, stage 1: faithful prefix map

<i>input</i>	×	ɔ	√	f	i	t	i	√	ɪ	×
<i>output</i>	×	ɔ								

Following the prefix vowel, the FST encounters the prefix-root boundary, notated here with the symbol √. As shown in (9), the transducer progresses through the input string and outputs all symbols between the two root boundary symbols faithfully.

## (9) Akan inner, left-subsequential FST tape, stage 2: faithful root map

<i>input</i>	×	ɔ	√	f	i	t	i	√	ɪ	×
<i>output</i>	×	ɔ	√	f	i	t	i	√		

In (10), the FST reaches a suffix vowel. Because the ATR value of the root is known, the FST outputs a matching suffix vowel.

## (10) Akan inner, left-subsequential FST tape, stage 3: suffix vowel map; right edge reached

<i>input</i>	×	ɔ	√	f	i	t	i	√	ɪ	×
<i>output</i>	×	ɔ	√	f	i	t	i	√	i	×

Following the application of this inner function, we see that affixes to the right of the root have been appropriately harmonized, but affixes to the left of the root have not. To capture the bidirectionality of harmony in Akan, we compose the left-subsequential function in Figure 3 with the right-subsequential function in Figure 4, taking the output of (10) and applying the right-subsequential function to it. To keep track of the derivation, we add another tape to our running tape representation below. The top tape represents the *initial input*, prior to application of the left-subsequential function; the middle, *intermediate* tape represents the output of the left-subsequential function and the input of the right-subsequential function. The new bottom tape represents the *final output* of the computation.

The right-subsequential function in (11) mirrors the left-subsequential function that was previously applied. The computation begins from the right-hand end of the input string, and upon reaching the first vowel in the string, the vowel is output faithfully.

## (11) Akan outer, right-subsequential FST, stage 1: faithful suffix vowel map

<i>initial input</i>	×	ɔ	√	f	i	t	i	√	ɪ	×
<i>intermediate</i>	×	ɔ	√	f	i	t	i	√	i	×
<i>final output</i>									i	×

Following the suffix vowel, the FST encounters the root-suffix boundary symbol, and as shown in (12), the transducer outputs all characters between the root boundary characters faithfully.

<sup>5</sup>The decision to treat the left-subsequential function as the inner function is arbitrary, as we point out later in §4.4.



- (12) Akan outer, right-subsequential FST, stage 2: faithful root map

<i>initial input</i>	×	ɔ	√	f	i	t	i	√	ɪ	×
<i>intermediate</i>	×	ɔ	√	f	i	t	i	√	i	×
<i>final output</i>			√	f	i	t	i	√	i	×

Finally, in (13) the FST reaches the initial, presently disharmonic prefix vowel. The ATR value of the root is known at this point, and the FST outputs a vowel matching the root value for the harmonic feature.

- (13) Akan outer, right-subsequential FST, stage 3: prefix vowel map; end of computation

<i>initial input</i>	×	ɔ	√	f	i	t	i	√	ɪ	×
<i>intermediate</i>	×	ɔ	√	f	i	t	i	√	i	×
<i>final output</i>	×	o	√	f	i	t	i	√	i	×

The computation is now complete, and the initial input has been successfully harmonized to the final output form observed in Akan. The analysis of this harmony pattern requires the composition of two subsequential functions, but maintains the alphabet size of the initial input throughout the application of both functions, satisfying the definition of a weakly deterministic function given by [Heinz and Lai \(2013\)](#). Furthermore, the reader may note that the order in which the left- and right-subsequential functions are applied does not impact the final output, and that the two functions are not partial inverses of each other. This is sufficient to conclude that the two subsequential functions do not *interact*, a relation between two functions that we discuss in detail in §4.3. In the next subsection, we describe non-deterministic functions and a set of phonological maps which require non-determinism to be described — unbounded circumambient patterns.

## 2.3 Non-determinism

As noted in §2.1, a subsequential transducer  $\tau: X^* \rightarrow Y^*$  defines a string-to-string function that is deterministic in its input string and has state transition and output functions that are deterministic in the current state and input symbol: any input string  $w \in X^*$  is mapped by  $\tau$  to at most one string in  $Y^*$  and any given (state, input symbol) pair is mapped by  $\delta$  and  $\omega$  to at most one state and at most one output string, respectively. If  $\tau$  can map any input string to more than one output string, then  $\tau$  is a non-deterministic function on strings — a proper string *relation*; a transducer that maps any input string to at most one output string is said to be *functional* or *single-valued*. If there are any (state, input symbol) pairs such that a transducer can ‘choose’ from among a set of two or more states to transition to, or from among a set of two or more strings to output, then that transducer’s transition and/or output functions are (respectively) non-deterministic.

Following [Heinz \(2018, §6.2.5\)](#), we exclusively use the term *regular relations* (abbreviated ‘RR’) to refer to the most general class of string-to-string mappings definable using FSTs — i.e., the class that includes transducers which are not functional as just defined. We also follow [Heinz \(2018, §6.2.5\)](#) in using *non-deterministic regular functions* (‘ND’) to refer to the class of transducers that are functional, but that can have non-deterministic state and/or output functions. Such transducers can have temporarily and incrementally ambiguous input strings, but the point of disambiguation may be an unbounded distance away from the location of the read/write head at the moment of ambiguity. As [Elgot and Mezei \(1965\)](#) show, any regular relation can be decomposed into two subsequential functions going in opposite directions, as long as the first function in the composition is allowed to enlarge the input alphabet:

- (14) A function  $\tau: X^* \rightarrow Z^*$  is a regular relation iff there exist two subsequential functions  $I: X^* \rightarrow Y^*$  and  $O: Y^* \rightarrow Z^*$  such that  $O$  is either left- or right-subsequential,  $I$  is the other,  $X \subseteq Y$ , and  $\tau = O \circ I$ .

As mentioned in the previous subsection, the intuition is that the first, inner function can effectively perform unbounded lookahead for the second function by ‘marking up’ intermediate string with extra information (in

the form of extra symbols that are in *Y* but not in *X*) that the second, outer function can then use to behave in a manner that is incrementally deterministic.

To see this, consider a hypothetical, pathological example of ‘sour grapes’ harmony (Wilson 2003) that resembles the Turkish rounding harmony pattern in (2). In this hypothetical language that we refer to as ‘Zurkish’, [+round] spreads rightward from roots to suffixes, triggering alternations on [+high] vowels (15c). As in Turkish, [−high] vowels do not undergo harmony — but if a [−high] vowel occurs anywhere in the word, no harmony occurs (15d,e). In other words: if [+round] cannot spread to the right edge of the word, it does not spread at all.

(15) Zurkish ‘sour grapes’ rounding harmony

	[−round] root	[+round] root	
a.	ip-in-i	c.	jyz-yn-y
b.	ip-in-de	d.	jyz-in-de      *jyz-yn-de
		e.	jyz-in-...-de      *jyz-yn-...-de

Subsequential functions may, upon reading a given input symbol, delay writing an output. In these instances, the FST continues to read more of the input string to determine the appropriate output for the relevant symbol. Crucially, the number of states a transducer can ‘wait’ is bounded and this bound must be known *a priori* (Chandlee et al. 2014, 2018, Chandlee and Heinz 2018, Jardine 2016). In the Zurkish data, as shown in (15e), the distance between the blocking segment and the relevant high vowel is (by hypothesis) unbounded, preventing an analysis depending on bounded look-ahead.

Recall Jardine’s (2016) claim that there is a typological asymmetry between segmental and tonal phonology: while segmental patterns which require non-determinism appear to be rare (e.g. Yaka height harmony; Hyman 1998) or pathological (e.g. sour grapes harmony, as exemplified by Zurkish), tonal patterns which require non-determinism are relatively common. He thus argues that the minimum expressivity required for tonal phonology is strictly greater than that required for segmental phonology, and he observes a similar property across attested non-deterministic patterns, both segmental and tonal: *unbounded circumambience*. Jardine (2016, 249) defines the class of unbounded circumambient processes as in (16).

(16) An unbounded circumambient process is a process:

- a. whose application is dependent on information (e.g. the presence of a trigger or blocker) on both sides of the target; [= *circumambient*]
- b. in which, on both sides, there is no bound on how far this information may be from the target. [= *unbounded*]

The non-deterministic patterns captured by this definition can be analyzed using a single non-deterministic function, or, as demonstrated in Elgot and Mezei (1965), the composition of left- and right-subsequential functions using mark-up on the intermediate output to describe them. We walk through Jardine’s analysis of tonal spreading in Copperbelt Bemba to explicate this point.<sup>6</sup>

In Copperbelt Bemba (Bickmore and Kula 2013, Kula and Bickmore 2015), a high tone spreads unboundedly to the right word edge in phrase-final forms, as seen in (17a–d). However, if another high tone intervenes between the first high tone and the word edge, then bounded ternary spreading takes place instead, as in (17e–g). We schematize high tone spreading in Copperbelt Bemba in (18).<sup>7</sup>

<sup>6</sup>This analysis is depicted as two FSTs in Figures 5 and 6 in the Appendix.

<sup>7</sup>Ternary spreading is analyzed as two separate processes in Bickmore and Kula (2013) and Kula and Bickmore (2015). The motivation for their analysis and the constraints on ternary spreading are not germane for the present paper.



## (17) Copperbelt Bemba tone spreading

- |    |                            |                           |                               |
|----|----------------------------|---------------------------|-------------------------------|
| a. | /u-ku-tul-a/               | ù-kù-tùl-à                | ‘to pierce’                   |
| b. | /bá-ka-fik-a/              | bá-ká-fíká                | ‘they will arrive’            |
| c. | /bá-ka-mu-londolol-a/      | bá-ká-mú-lóondólól-á      | ‘they will introduce 3.SG’    |
| d. | /tu-ka-páapaatik-a/        | tù-kà-páápáátík-á         | ‘we flatten’                  |
| e. | /bá-ka-pat-a kó/           | bá-ká-pát-à kó            | ‘they will hate’              |
| f. | /bá-ka-londolol-a kó/      | bá-ká-lóondòlòl-à kó      | ‘they will introduce them’    |
| g. | /tu-ka-béleegj-el-an-a kó/ | tù-kà-bélééggj-él-àn-à kó | ‘we will read for each other’ |

## (18) Schema for Copperbelt Bemba tone spreading

- |    |   |  |
|----|---|--|
| a. | $L^n \rightarrow L^n$                                     | no H present   |
| b. | $\dots HL^m \rightarrow \dots HH^m$                       | unbounded H spreading in the absence of a following H            |
| c. | $\dots HL^n H \dots \rightarrow \dots HHHL^{n-2} H \dots$ | ternary H spreading in the presence of a following H, $n \geq 2$ |
| d. | $\dots HLH \dots \rightarrow \dots HHH \dots$             | binary spreading up to a following H                             |

In (19)–(24) we show ternary spreading with a running tape representation of an input string, /HLLLH/. The input-output mapping here is composed of a left-subsequential FST and a right-subsequential FST which reads the output of the left-subsequential FST as its input and outputs the actual attested form. The order between the two FSTs is not arbitrary: the first/inner, left-subsequential FST adds to mark-up that the second/outer, right-subsequential FST then takes advantage of.

Generally speaking, the left-subsequential FST outputs all L tones without modification until encountering a H tone, upon which the H ‘spreads’ to two following L tones. For all following L tones, the FST outputs a distinct symbol (notated as  $\Psi$  here) not contained in the input alphabet, because it cannot determine whether the output should be an L (or an H) tone until it does (or does not) encounter another H tone later in the word. Since a second H tone may in principle occur an unbounded distance from the first, the FST cannot ‘wait’ and so instead outputs the placeholder  $\Psi$  symbol, indicating that the preceding context matches the left-hand side of the structural description for unbounded spreading. The FST iteratively outputs all subsequent input L tones as  $\Psi$  until it encounters either an input H, which is output without modification, or the right word edge.

Now on with the input /HLLLH/. The inner, left-subsequential FST first reads the word-initial H tone and initiates ternary spreading to the immediately following two L tones, as shown in (19).

## (19) Copperbelt Bemba inner, left-subsequential FST, stage 1: ternary spread

input	×	H	L	L	L	H	×
output	×	H	H	H			

Next, the FST reads the third L tone from the input, but cannot determine at this point whether to output an H or an L tone, because the presence or absence of a following H tone is unknown. The FST thus outputs a new symbol not contained in the input alphabet,  $\Psi$ , as shown in (20). This new symbol will ultimately provide the outer, right-subsequential FST with the information necessary to determine all tone values for the word.

## (20) Copperbelt Bemba inner, left-subsequential FST, stage 2: mark-up

input	×	H	L	L	L	H	×
output	×	H	H	H	$\Psi$		

Progressing through the input string, the inner FST continues as described above. In (21), the FST reads a second input H tone and maps it faithfully, and then reaches the right edge.

## (21) Copperbelt Bemba inner, left-subsequential FST, stage 3: right edge reached

input	×	H	L	L	L	H	×
output	×	H	H	H	$\Psi$	H	×

The outer, right-subsequential FST now reads the output just produced by the inner, left-subsequential FST as its input and completes the input-output mapping. Generally speaking again, the outer FST outputs all input L and H tones without modification, but maps intermediate  $\Psi$  according to the previously-read context. Proceeding from right to left, if the FST reads an H tone, then it outputs all  $\Psi$  as L; if it does not encounter H, then it outputs all  $\Psi$  as H. The outer FST is able to discern whether a second H tone is present in the word, and uses the mark-up passed from the inner FST to determine whether ternary or unbounded spreading occurs.

This is shown in (22)–(24) for the more specific form under discussion. First, the outer FST reads the word-final H tone and outputs it faithfully.

- (22) Copperbelt Bemba outer, right-subsequential FST, stage 1: faithful H map

<i>initial input</i>	×	H	L	L	L	H	×
<i>intermediate</i>	×	H	H	H	$\Psi$	H	×
<i>final output</i>						H	×

Since an input H has been encountered, the FST outputs all  $\Psi$  as L in (23) and all H faithfully in (24). The result in this case is thus ternary rather than unbounded spreading from the initial H.

- (23) Copperbelt Bemba outer, right-subsequential FST, stage 2: mark-up resolution

<i>initial input</i>	×	H	L	L	L	H	×
<i>intermediate</i>	×	H	H	H	$\Psi$	H	×
<i>final output</i>					L	H	×

- (24) Copperbelt Bemba left-subsequential FST, stage 3: left edge reached; computation ends

<i>initial input</i>	×	H	L	L	L	H	×
<i>intermediate</i>	×	H	H	H	$\Psi$	H	×
<i>final output</i>	×	H	H	H	L	H	×

Unbounded circumambient processes like tone spreading in Copperbelt Bemba can thus be analyzed as regular non-deterministic maps, either in the form of a single non-deterministic FST, as in [Jardine \(2016\)](#), or as the composition of two subsequential functions that may use an enlarged alphabet of symbols not present in the initial input alphabet.

## 2.4 Summary

The three levels of expressivity defined and exemplified in this section can be summarized as follows. Subsequential regular functions can describe unidirectional processes with bounded lookahead (and unbounded ‘lookbehind’), as in Turkish harmony. Weakly deterministic functions are intended to be able to describe bidirectional processes where the first ‘pass’ is *not* allowed to behave as lookahead for, or otherwise affect the behavior of, the second pass, as in Akan. Non-deterministic regular functions can describe compositions of unidirectional processes going in opposite directions, where the first pass may serve as unbounded lookahead for, or otherwise affect the behavior of, the second pass, as in Tutrugbu.

[Jardine \(2016, §5.4\)](#) acknowledges two attested cases of apparent unbounded circumambience in segmental phonology, Sanskrit *n*-retroflexion ([Ryan 2017](#)) and Yaka height harmony ([Hyman 1998](#)). [Jardine](#) does not consider them equivalent to tonal patterns Copperbelt Bemba’s, however, claiming that such segmental patterns are ‘extremely rare.’ He suggests that the harmony patterns in Sanskrit and Yaka may not actually be unbounded, further speculating that the extent to which Yaka height harmony is unbounded may have been ‘generalized’ from a pre-existing unbounded circumambient tonal process. In the next section we describe ATR harmony in Tutrugbu, which further challenges the claim that segmental phonology is at most weakly deterministic. Looking ahead, in §5.1 we point to a number of other unbounded circumambient vowel harmony patterns as evidence that these patterns are more widely attested than previously thought.

### 3 Tutrugbu ATR harmony

Tutrugbu is a Ghana-Togo Mountain language (Kwa) spoken in southeastern Ghana. The data for this paper come from formal elicitation as well as a documentary corpus of natural speech. The same pattern of harmony detailed below is also attested in Tafi (Bobuafor 2013), a closely related language.

#### 3.1 Data

Tutrugbu has an inventory of nine oral vowels, /a ɔ o u ε e i i/, with contrasts in height, backness, rounding, and ATR.<sup>8</sup> Nasal vowels are also phonemic in the language, but they behave just like oral vowels with respect to ATR harmony. Harmony proceeds from right to left in Tutrugbu, from roots to prefixes. Suffixation is rare, and suffixes do not generally undergo vowel harmony, so they are not discussed below.

Observe the ATR pairings demonstrated by regressive harmony on noun class prefixes in (25). In (25a), prefixal /a/ alternates with /e/, while in (25b) /ɔ/ alternates with /o/. In (25c), /i/ alternates with /i/, and finally, in (25d) /u/ alternates with /u/. Note that /ε/ does not occur in affixes.

(25) ATR harmony in nouns

	Class	[−ATR]	Gloss	[+ATR]	Gloss
a.	1	a-ɲɪ	‘C1-man’	e-bú	‘C1-dog’
b.	3	ɔ-da	‘C3-copper’	o-pétē	‘C3-vulture’
c.	4	i-da	‘C4-copper’	i-pétē	‘C4-vulture’
d.	8	bɔ-wɪ	‘C8-axe’	bu-ju	‘C8-war’

Noun class prefixes undergo ATR harmony, but concatenating more prefixes to a nominal root is not possible. Verbal morphology on the other hand allows for more morphological complexity. In (26) we see that words with only [+high] prefix vowels show full harmony. In (26a–c), [−ATR] roots are preceded by [−ATR] prefixes, while in (26d–f), [+ATR] roots are preceded by [+ATR] prefixes.

(26) Harmony on [+high] prefixes

	[−ATR] root	Gloss		[+ATR] root	Gloss
a.	i-tí-bá	‘1S-NEG-come’	d.	i-tí-ɛē	‘1S-NEG-grow’
b.	bɔ-tí-bá	‘1P-NEG-come’	e.	bu-tí-ɛē	‘1P-NEG-grow’
c.	ki-tí-bá	‘C5-NEG-come’	f.	ki-tí-ɛē	‘C5-NEG-grow’

Full harmony also obtains when all prefix vowels are [−high], as shown in (27). In (27d–f), [+ATR] roots propagate their [+ATR] feature to the left edge of the word.<sup>9</sup>

(27) Harmony on [−high] prefixes

	[−ATR] root	Gloss		[+ATR] root	Gloss
a.	a-ba-bá	‘3S-FUT-come’	d.	e-be-ɛē	‘3S-FUT-grow’
b.	ka-ba-bá	‘C7-FUT-come’	e.	ke-be-ɛē	‘C7-FUT-grow’
c.	ɔ-bo-bá	‘2S-FUT-come’	f.	o-bo-ɛē	‘2S-FUT-grow’

In (26) and (27), all prefix vowels agree in height. Harmony in forms with prefix vowels of differing heights is shown in (28). As in (27), if the initial-syllable vowel is [−high], harmony obtains throughout the word.

<sup>8</sup>The high [−ATR] vowels, /u/ and /i/, surface as mid vowels, [ɔ] and [ε], but pattern as high vowels. To make their phonological status clear we will transcribe them as [u] and [i] throughout. For more discussion on abstractness in the Tutrugbu vowel inventory, see Essegbey and McCollum (2017) and McCollum and Essegbey (2018).

<sup>9</sup>Note also in (27c,f) that round vowels in the initial syllable trigger progressive labial harmony (Essegbey and McCollum 2017).

## (28) Harmony on varying-height prefixes and a [–high] initial prefix

	[–ATR] root	Gloss		[+ATR] root	Gloss
a.	a-tí-ba-bá	‘3S-NEG-FUT-come’	d.	e-tí-be-ǵē	‘3S-NEG-FUT-grow’
b.	ɔ-tí-bɔ-bá	‘2S-NEG-FUT-come’	e.	o-tí-bo-ǵē	‘2S-NEG-FUT-grow’
c.	ka-tí-ba-bá	‘C7-NEG-FUT-come’	f.	ke-tí-be-ǵē	‘C7-NEG-FUT-grow’

In contrast to the previous examples, [–high] vowels block harmony if the vowel in the initial syllable is [+high]. Regardless of the root’s ATR value, the vowel of the [–high] FUT prefix and all preceding vowels surface as [–ATR] under this condition, as shown in (29).

## (29) Disharmony on varying-height prefixes and a [+high] initial prefix

	[–ATR] root	Gloss		[+ATR] root	Gloss
a.	ɪ-ba-bá	‘1S-FUT-come’	d.	ɪ-ba-ǵē	‘1S-FUT-grow’
b.	bʊ-ba-bá	‘1P-FUT-come’	e.	bʊ-ba-ǵē	‘1P-FUT-grow’
c.	kɪ-ba-bá	‘C5-FUT-come’	f.	kɪ-ba-ǵē	‘C5-FUT-grow’

In words with a [+ATR] root, a [+high] initial-syllable vowel, and a medial [–high] prefix vowel (satisfying the two conditions necessary to block harmony), a [+high] prefix vowel intervening between the root and the medial [–high] vowel undergoes harmony, as shown in (30).

## (30) Partial harmony on [+high] prefixes

a.	ɪ-ba-di-wu	‘1S-FUT-ITIVE-climb’
b.	bʊ-ba-di-wu	‘1P-FUT-ITIVE-climb’
c.	kɪ-ba-di-wu	‘C5-FUT-ITIVE-climb’

The data above show that [–high] vowels are **CONDITIONAL BLOCKERS**: they block harmony only in the presence of an initial-syllable high vowel. Two [+high] vowel prefixes do not block harmony, as in (26), and two [–high] vowel prefixes do not block harmony, as in (27) and (28). It is only the combination of an initial-syllable [+high] vowel and a medial [–high] vowel that blocks harmony. In other words, the realization of a [–high] prefix vowel depends not only on the ATR value of the vowel in the immediately following morpheme (the root, or a prefix closer to the root), but also on the height of the initial-syllable vowel.

In (29) and (30), the initial-syllable [+high] vowel and the medial [–high] vowel are in adjacent syllables. In (31) below we see that harmony is blocked by the co-presence of these two conditions, even when separated by a number of intervening syllables.<sup>10</sup>

## (31) ATR harmony

	Context	Example	Gloss
a.	no #[+high] prefix	e-tí-wu	‘3S-NEG-climb’
b.	no [–high] prefix	ɪ-tí-wu	‘1S-NEG-climb’
c.	adjacent	ɪ-ba-wu	‘1S-FUT-climb’
d.	1σ gap	ɪ-tí-ka-wu	‘1S-NEG-PFV-climb’
e.	2σ gap	ɪ-tí-ka-a-wu	‘1S-NEG-PFV-PROG-climb’
f.	3σ gap	ɪ-tí-ka-a-ba-wu	‘1S-NEG-PFV-PROG-VENT-climb’
g.	4σ gap	ɪ-tí-ka-a-ba-ba-wu	‘1S-NEG-PFV-PROG-VENT-VENT-climb’
h.	no #[+high] prefix	e-tí-ke-e-be-be-wu	‘3S-NEG-PFV-PROG-VENT-VENT-climbs’

<sup>10</sup>The following are rough translations of the examples in (31): (a) “S/he does not climb”; (b) “I do not climb”; (c) “I will climb”; (d) “I no longer climb”; (e) “I am no longer climbing”; (f) “I am no longer coming to climb”; (g) “I am no longer coming to climb (defiant)”; (h) “S/he is no longer coming to climb (defiant)”.

Tutrugbu ATR harmony is blocked only when two conditions are met: one, the initial-syllable vowel is [+high], and two, another prefix vowel is [−high], as in (31c–g). When only one of these conditions is met — as in (31a,b,h) — harmony obtains. In (31c), the [+high] initial-syllable vowel and [−high] prefix vowel are adjacent, and harmony fails. In (31d–g), one, two, three, and four syllables intervene between these two interacting conditions on harmony. Thus, the blocking of regressive ATR harmony depends on decidedly non-local information — the height of the initial-syllable vowel — and the presence of a [−high] prefix vowel, which may occur a number of syllables from the initial syllable with no principled upper bound.

### 3.2 Analysis

This section presents a finite-state analysis of regressive ATR harmony and conditional blocking in Tutrugbu. We demonstrate that vowel harmony in Tutrugbu is as complex as tonal spreading in Copperbelt Bemba.

Since ATR harmony in Tutrugbu is regressive, the analysis begins with a right-subsequential FST. We first demonstrate a simple attempt at modelling ATR harmony below in (32)–(34), using the example [i-tí-wu] (31b). Since harmony is regressive and root-controlled in the language, all segments to the right of the root-prefix boundary (represented with  $\sqrt{\phantom{x}}$ ) are output without modification, as shown in (32).

- (32) Tutrugbu right-subsequential FST, stage 1: faithful root map

<i>input</i>	×	i	t	í	$\sqrt{\phantom{x}}$	w	u	×
<i>output</i>					$\sqrt{\phantom{x}}$	w	u	×

In (33), the FST reads a prefix vowel and outputs it according to the ATR value of the root.<sup>11</sup>

- (33) Tutrugbu right-subsequential FST, stage 2: prefix vowel map

<i>input</i>	×	i	t	í	$\sqrt{\phantom{x}}$	w	u	×
<i>output</i>	×		t	í	$\sqrt{\phantom{x}}$	w	u	×

The right-subsequential FST outputs further prefix vowels in the same way as in (33). The output ATR value for a given prefix vowel depends on the output ATR value of the nearest following output vowel. In (34), the realization of the 1S prefix /i-/ is, like the vowel of the negation prefix /tí-/ , output as [+ATR]. The next symbol read is the left word edge symbol, and the computation is complete.

- (34) Tutrugbu right-subsequential FST, stage 3: left edge reached; computation ends

<i>input</i>	×	i	t	í	$\sqrt{\phantom{x}}$	w	u	×
<i>output</i>	×	i	t	í	$\sqrt{\phantom{x}}$	w	u	×

The input-output mapping shown in (32)–(34) models harmony when both blocking conditions are not met. If, however, an initial [+high] prefix and a [−high] medial prefix co-occur, then this right-subsequential FST cannot on its own properly determine whether a given prefix vowel will be realized as [+ATR] based solely on the nearest vowel to the right. Compare the forms in (35). In (35a), the FUT prefix is [+ATR] before the root /wu/ ‘climb’, but in (35b), the same prefix is [−ATR] even though it occurs before the same root.

- (35) Near-minimal pair exemplifying blocking conditions

- a. e-be-wu ‘1S-FUT-climb’ all [−high] prefix vowels
- b. i-ba-wu ‘3S-FUT-climb’ initial [+high] and medial [−high] prefix vowels

Since the right-subsequential FST does not have access to information to the left of the target vowel (that is in principle an unbounded distance away), it cannot determine which allomorph of FUT should surface, [ba] or [be]. This indeterminacy precludes a subsequential analysis of Tutrugbu. If, however, we mimic

<sup>11</sup>The realization of consonants does not depend on ATR harmony, so prefix consonants are output without modification.

the structure of the analysis provided for Copperbelt Bemba in §2.3, where the first FST is able to introduce intermediate mark-up into the derivation, the Tutrugbu pattern is analyzable.

Generally speaking, the first, right-subsequential FST proceeds leftward, outputting all high vowels in accordance with the root's ATR value. If, after reading a [+ATR] root, the FST reads an input [−high] vowel, the transducer outputs  $\Psi$  and outputs all subsequent vowels as either  $\Psi$  (for further [−high] vowels) or  $\mathfrak{h}$  (for [+high] vowels). By doing so, this first FST passes information about the ATR value of the root and the presence of a potential blocker to the second, left-subsequential FST.

This analysis is illustrated in (36)–(38) using the example [i-ba-wu] (35b). First, the right-subsequential FST faithfully maps the root.

- (36) Tutrugbu inner, right-subsequential FST, stage 1: faithful root map

<i>input</i>	⌘	i	b	a	√	w	u	⌘
<i>output</i>					√	w	u	⌘

Next, this inner FST encounters a prefix with a [−high] vowel. Having not yet determined the height of the initial prefix vowel, the [−high] prefix vowel is output with the new symbol  $\Psi$  that encodes its height and leaves its ATR value to be determined by the outer, left-subsequential FST.

- (37) Tutrugbu inner, right-subsequential FST, stage 2: [−high] prefix vowel mark-up

<i>input</i>	⌘	i	b	a	√	w	u	⌘
<i>output</i>			b	$\Psi$	√	w	u	⌘

Next, the inner FST encounters a prefix with a [+high] vowel. It happens to be the initial prefix in this case, but as happens with all [+high] prefix vowels, it is output with the new symbol  $\mathfrak{h}$  that encodes its height and leaves its ATR value to be determined. The left edge is then reached and this FST's job is done.

- (38) Tutrugbu inner, right-subsequential FST, stage 3: [+high] prefix vowel mark-up; left edge reached

<i>input</i>	⌘	i	b	a	√	w	u	⌘
<i>output</i>	⌘	$\mathfrak{h}$	b	$\Psi$	√	w	u	⌘

The outer, left-subsequential FST then proceeds from left to right using the output of the inner FST as its input. Upon encountering a  $\mathfrak{h}$  in the initial syllable, the outer FST ‘knows’ that the conditions for blocking have been met, because initial  $\mathfrak{h}$  indicates both that the root is [+ATR] and that there is a medial [−high] prefix vowel. Because  $\mathfrak{h}$  is in the initial syllable, this FST outputs all  $\mathfrak{h}$  and  $\Psi$  as [i] and [a], respectively. In this way, this analysis of Tutrugbu parallels the non-deterministic analysis of Copperbelt Bemba in §2.3.

- (39) Tutrugbu outer, left-subsequential FST: mark-up resolved; right edge reached; computation ends

<i>initial input</i>	⌘	i	b	a	√	w	u	⌘
<i>intermediate</i>	⌘	$\mathfrak{h}$	b	$\Psi$	√	w	u	⌘
<i>final output</i>	⌘	i	b	a	√	w	u	⌘

As [Jardine \(2016\)](#) notes, it is possible to allow the FST to ‘wait’ a finite number of states; these allow the machine to look ahead and access information on the far side of the target vowel before deciding what to output for that target vowel. To preserve subsequentiality while introducing ‘waiting’, the look-ahead context must be of some *a priori* known bounded length. Given that the context to the left of a target [−high] vowel in Tutrugbu may be unboundedly long, then there is no single number of ‘waiting’ transition states that can model all of the possible data and capture the principled generalization describing it. In (31) we demonstrated that [−high] vowels could be separated from the initial syllable by a number of syllables with no principled upper bound. The generalization is that the realization of ATR on [−high] prefix vowels depends on both the ATR value of the root and the height value of the initial syllable. Furthermore, a given [−high] vowel may occur a potentially unbounded distance from both the root and the initial syllable. The conditional blocking pattern in Tutrugbu is thus unbounded circumambient, like the tonal processes analyzed in [Jardine \(2016\)](#).



## 4 Revising the definition of weak determinism

In this section, we show that [Heinz and Lai \(2013\)](#)’s definition of weakly deterministic regular functions can be satisfied by many of the functions it was intended to exclude. We then introduce an alternative definition centered on a notion in §4) of *interaction* between composed functions that addresses the inadequacies of [Heinz and Lai](#)’s definition and correctly differentiates patterns like ATR harmony in Akan from unbounded circumambient patterns like those found in Copperbelt Bemba and Tugrugbu.

To demonstrate these results, we start by explicating the existing definition of weakly deterministic regular functions and what we take its intended scope to be. Next, we show that the Tugrugbu and Copperbelt Bemba patterns — patterns outside the intended scope of weak determinism — can in fact be analyzed in a fashion that satisfies the letter of this definition. We then walk through a relatively general method, using phonotactically-illicit intermediate symbol sequences as ‘cheat codes’ for extra symbols, by which non-deterministic functions can be defined in a way that technically meets the [Heinz and Lai](#)’s (2013) definition of weak determinism. We argue that a definition that captures our interpretation of the intended scope of weak determinism (excluding e.g. compositions of subsequential functions using extra intermediate symbols, length-increasing codes, or phonotactic cheats) must forbid the inner function in the composition from affecting or overwriting (‘interacting with’) the behavior of the outer function. We conclude the section with an illustrative discussion of how this general notion of ‘interaction’ captures the difference in complexity between weakly deterministic harmony in Akan and non-deterministic harmony in Tugrugbu.

### 4.1 The existing definition of the weakly deterministic regular functions

[Heinz and Lai \(2013\)](#) mainly define the class of weakly deterministic regular functions as a restriction on the class of regular relations: any regular relation that can be defined as the composition of a left- and a right-subsequential function where the output alphabet of the function applied first is equivalent to its input alphabet falls within the class of weakly deterministic regular functions.

- (40) A regular string-to-string function  $\tau: X^* \rightarrow Y^*$  is weakly deterministic iff there exist two subsequential functions  $I: X^* \rightarrow X^*$  and  $O: X^* \rightarrow Y^*$  such that  $O$  is either left- or right-subsequential,  $I$  is the other, and  $\tau = O \circ I$ .

This restriction effectively bans ‘extra’ intermediate symbols that could be used to flag portions of the input for future computations. However, as [Heinz and Lai \(2013, §2.2\)](#) note, the restriction is not sufficient to exclude what it intends: the functionality of [Elgot and Mezei’s \(1965\)](#) extra intermediate alphabet symbols — ‘marking up’ the input string for the benefit of the second (‘outer’) function in the composition — can under relatively weak conditions be accomplished *without* extra symbols. That is, suppose we are given a properly non-deterministic regular function  $\tau: X^* \rightarrow Z^*$  such that

- (41) a.  $\tau = L \circ R$ , where  $L$  and  $R$  are left-subsequential and right-subsequential functions, and  
b.  $R$  uses an enlarged intermediate alphabet  $Y$  to mark up input for  $L$ .

By the assumption that  $\tau$  is properly non-deterministic, it is a function that *ought* to be excluded from the class of weakly deterministic regular functions. Nevertheless, as long as the input alphabet  $X$  contains more than two symbols, we can construct alternative functions  $L'$  and  $R'$  such that  $R': X^* \rightarrow X^*$  (i.e.  $R'$  does *not* use an enlarged alphabet),  $L': X^* \rightarrow Z^*$ , and  $\tau = L' \circ R'$ . To do this,  $R'$  need only *encode* each extra symbol in  $Y$  as a *sequence* of symbols over  $X$ , but can otherwise behave identically.

To exclude length-increasing codes for extra symbols, [Heinz and Lai](#) add an extra condition to the definition of the weakly deterministic regular functions:

- (42) A regular string-to-string function  $\tau: X^* \rightarrow Y^*$  is weakly deterministic iff there exist two subsequential functions  $I: X^* \rightarrow X^*$  and  $O: X^* \rightarrow Y^*$  such that  $O$  is either left- or right-subsequential,  $I$  is the other,  $I$  is not length-increasing, and  $\tau = O \circ I$ .

We take the intention of [Heinz and Lai’s \(2013\)](#) definition for weakly deterministic regular functions and its restrictions on expanded intermediate alphabets and length-increasing codes to be the exclusion of patterns requiring unbounded lookahead. That is, in both mark-up strategies prohibited by [Heinz and Lai \(2013\)](#), the inner function of the composition can provide unbounded lookahead for the outer function, and the outer function can then use that information to behave entirely deterministically. Below we show there are other such strategies not banned by the definition in (42), and then offer a revised definition intended to prohibit any kind of composition where one function INTERACTS WITH (‘affects’, ‘depends on’, ‘interferes with’) the other.

## 4.2 Mark-up without extra symbols or length-increasing codes

In addition to the non-deterministic right-subsequential transducer shown in Figure 7 in the [Appendix](#), and a composition of non-alphabet-preserving subsequential transducers noted in §3.2, we demonstrate here that the unbounded circumambient patterns in Tutrugbu and Copperbelt Bemba are analyzable using the composition of multiple subsequential transducers that do not introduce the kind of mark-up specifically prohibited by [Heinz and Lai \(2013\)](#) — additional intermediate symbols or length-increasing codes.

For Tutrugbu, we offer an analysis that, despite being both alphabet- and length-preserving, is capable of describing the data in (43), where full harmony in (43a) is contrasted with partial harmony in (43b).

- (43) Full harmony vs. partial harmony in Tutrugbu
- a. e-be-di-wu ‘1S-FUT-ITV-climb’
  - b. i-ba-di-wu ‘3S-FUT-ITV-climb’

First, a left-subsequential FST proceeds from the left edge of the word. (Note that this differs from our analysis in §3.2 above, which began with a right-subsequential FST proceeding from the root to prefixes.) If the initial-syllable vowel is [−high], all prefix vowels are output as [−ATR]; if the root is [+ATR], the second, right-subsequential FST will change all prefix vowels in this case to [+ATR]. If, however, the initial-syllable vowel is [+high], the first FST outputs all prefix [+high] vowels as [−ATR], but outputs all prefix [−high] vowels as [+ATR].<sup>12</sup> The operation of this first function is demonstrated in the running tape representations below. In (44), the initial vowel is [+high] and is output as [−ATR].

- (44) Tutrugbu inner left-subsequential FST, stage 1: initial-syllable [+high] vowel map

<i>input</i>	×	i	b	a	d	i	√	w	u	×
<i>output</i>	×	i								

Since the initial vowel is [+high], the FST rewrites the following [−high] vowel as [+ATR] in (45). (As before, consonanants are always output faithfully, and these steps in the derivation are ignored for brevity.)

- (45) Tutrugbu inner, left-subsequential FST, stage 2: medial [−high] vowel map

<i>input</i>	×	i	b	a	d	i	√	w	u	×
<i>output</i>	×	i	b	e						

The FST continues rightward in (46), now outputting the following [+high] vowel as [−ATR].

- (46) Tutrugbu inner, left-subsequential FST, stage 3: medial [+high] vowel map

<i>input</i>	×	i	b	a	d	i	√	w	u	×
<i>output</i>	×	i	b	e	d	i				

In its final stage, the left-subsequential FST encounters the root boundary, outputs all following input vowels faithfully, and reaches the right edge of the word.

<sup>12</sup>The result is thus in effect a ‘Duke of York’ derivation ([Pullum 1976](#)) for [−high] medial prefix vowels in the context of initial-syllable [+high] vowels: they start out as [−ATR] and become [+ATR], only to be ultimately output as [−ATR] again.

- (47) Tutrugbu inner left-subsequential FST, stage 4: faithful root map; right edge reached

<i>input</i>	×	i	b	a	d	i	√	w	u	×
<i>output</i>	×	i	b	e	d	i	√	w	u	×

The second, right-subsequential FST now takes this intermediate output and maps it to the actual attested final output form, [ɪbadiwu] (43b). Proceeding leftward from the right word edge, this FST determines the ATR value of each prefix vowel using the output of the vowel to its right. Crucially, though, all prefix [−high] vowels that have been rendered as [+ATR] by the first FST are output as [−ATR] by the second. Because the first FST only renders [−high] prefix vowels as [+ATR] if the initial-syllable vowel is [+high], the combined effect of these two FSTs is that [−high] vowels block ATR harmony just in case the initial vowel is [+high].

The right-subsequential FST starts from the right edge in (48). All segments to the right of the root boundary are output faithfully. Since the first prefix vowel encountered is [+high], the FST outputs it as [+ATR], agreeing with the ATR value of the root.

- (48) Tutrugbu outer, right-subsequential FST, stage 1: faithful root map; medial [+high] vowel map

<i>initial input</i>	×	i	b	a	d	i	√	w	u	×
<i>intermediate</i>	×	i	b	e	d	i	√	w	u	×
<i>final output</i>	×				d	i	√	w	u	×

When the FST encounters a [−high, +ATR] vowel, it rewrites it as [−ATR].

- (49) Tutrugbu outer, right-subsequential FST, stage 2: medial [−high] vowel map

<i>initial input</i>	×	i	b	a	d	i	√	w	u	×
<i>intermediate</i>	×	i	b	e	d	i	√	w	u	×
<i>final output</i>	×		b	a	d	i	√	w	u	×

As a result, all vowels to the left of this vowel will be [−ATR] in agreement with the output ATR value of the following vowel.<sup>13</sup> Then the initial-syllable vowel is output as [−ATR], in agreement with the [−ATR] feature of the following output vowel, as seen in (50).

- (50) Tutrugbu outer, right-subsequential FST, stage 3: initial [+high] vowel map; computation ends

<i>initial input</i>	×	i	b	a	d	i	√	w	u	×
<i>intermediate</i>	×	i	b	e	d	i	√	w	u	×
<i>final output</i>	×	i	b	a	d	i	√	w	u	×

The composition of the two FSTs described above, shown in Figures 8 and 9 in the Appendix, is able to model the unbounded circumambient nature of harmony in Tutrugbu without either of the mark-up strategies outlined in Heinz and Lai (2013). Instead, the first, left-subsequential FST creates a phonotactic sequence that effectively smuggles information from the left edge of the word into the second, right-subsequential FST. The first FST only outputs a [−high, +ATR] vowel if the initial vowel is [+high] and is followed at any distance by a prefix [−high] vowel. The only segment to block harmony for the second FST is the [−high, +ATR] prefix vowel closest to the root. By structuring the derivation in this way, the unbounded circumambience of Tutrugbu's harmony pattern receives an analysis that is compatible with Heinz and Lai's definition of weak determinism in (42). As we argue below, this points not to the pattern in Tutrugbu being weakly deterministic, but rather, to the need to revise the definition of weak determinism.

Before introducing our revised definition, we first extend the type of analysis pursued for Tutrugbu to tonal spreading in Copperbelt Bemba. Recall from (18), repeated below in (51), that a H tone spreads unboundedly if there is no following H tone (51b), but H tone spreads, at maximum, ternarily if another underlying H tone follows (51c,d).

<sup>13</sup>In other words, intermediate-level [−high, +ATR] vowels block ATR harmony under this analysis.

## (51) Schema of Copperbelt Bemba tone spreading

- a.  $L^n \rightarrow L^n$  no H present
- b.  $\dots HL^m \rightarrow \dots HH^m$  unbounded H spreading in the absence of a following H
- c.  $\dots HL^n H \dots \rightarrow \dots HH L^{n-2} H \dots$  ternary H spreading in the presence of a following H,  $n \geq 2$
- d.  $\dots HLH \dots \rightarrow \dots HHH \dots$  binary spreading up to a following H

To account for the basic Copperbelt Bemba facts, we use a left-subsequential FST that promotes ternary spreading, which is then followed by a right-subsequential FST that completes spreading in the absence of a second H tone. To see how the composition of these two FSTs works, consider the patterns below.<sup>14</sup>

## (52) Copperbelt Bemba tonal spreading

- a.  $/LHLLLL/ \rightarrow [LHHHHH]$  unbounded spreading
- b.  $/LHLLHL/ \rightarrow [LHHHLH]$  ternary spreading

In general, if the first transducer reads an input string that should exhibit unbounded spreading, it outputs ternary spreading and then codes the right edge of the word with a HL sequence. To ensure that the transducer can write a HL sequence at the end of the word, it ‘waits’ two symbols after encountering a H. Upon reading the right word boundary symbol, the transducer outputs a HL sequence if no other H tones were encountered. Since the output tape is two symbols behind the input tape, this final HL sequence does not increase the length of the string, obeying the limitations imposed on weakly deterministic functions in [Heinz and Lai \(2013\)](#). This word-final HL sequence serves one purpose — to inform the second transducer that the structural description for unbounded spreading has been met. This word-final HL sequence can accomplish this because there is no other input that would generate a final sequence of HL. Since H spreads rightward and is only blocked by a following H, this substring is ill-formed, like the medial [–high, +ATR] vowel in the Tutrugbu analysis above. Since the Tutrugbu vowel inventory is larger than the tonal inventory in Copperbelt Bemba, it was possible and simpler to use a single symbol to furtively introduce a form of mark-up. In the absence of a larger alphabet, we use a string of symbols below to serve that same purpose.

To model (52b), the first FST, proceeding from left to right, maps all input L tones without modification until it encounters a H tone, which it also maps faithfully.

## (53) Copperbelt Bemba inner, left-subsequential FST, stage 1: faithful map to first H

input	×	L	H	L	L	L	H	×
output	×	L	H					

After reading an H tone, two immediately following L tones trigger a two-symbol ‘wait’ phase for the FST. This waiting is indicated by  $\lambda$  symbols in the relevant cells of the output tape in (54), but note that these symbols technically represent the *absence* of an output for the corresponding input symbol. This technicality will be resolved when this output is given as input to the second FST in (57).

## (54) Copperbelt Bemba inner, left-subsequential FST, stage 2: two-symbol wait

input	×	L	H	L	L	L	H	×
output	×	L	H	$\lambda$	$\lambda$			

If a subsequent H tone is encountered, the FST outputs the necessary H tones to model bounded spreading. In (55) though, a third input L is read, and the FST outputs H. Since the FST only outputs a single symbol, the output tape is still two symbols behind the input tape.

(55) Copperbelt Bemba inner, left-subsequential FST, stage 3:  $L \rightarrow H$ 

input	×	L	H	L	L	L	H	×
output	×	L	H	$\lambda$	$\lambda$	H		

<sup>14</sup>The two FSTs used to model tonal spreading in Copperbelt Bemba are shown in Figures 5 and 6 in the [Appendix](#).

In the next step, the FST encounters a second H, and outputs HLH (56) — and since that H is final, the right edge is also reached. This accomplishes two things. First, by outputting three symbols, the output tape has caught up with the input tape. Second, the FST has modeled ternary spreading, by outputting the two L tones immediately following the first input H tone faithfully.

- (56) Copperbelt Bemba inner, left-subsequential FST, stage 4: H → HLH; right edge reached

<i>input</i>	×	L	H	L	L	L	H	×
<i>output</i>	×	L	H	λ	λ	H	HLH	×

The second, right-subsequential FST operates as follows. Reading from right to left, if a word-final H tone is read, then all intermediate outputs from the first FST are output without modification, completing the input-output mapping. This is precisely what happens in (57) below. Note that technicality of the ‘wait’ symbols of the first FST has been resolved here; these are not visible to the second FST, which only sees the string of H, L, and boundary symbols output by the first FST.

- (57) Copperbelt Bemba outer, right-subsequential FST: complete computation

<i>initial input</i>	×	L	H	L	L	L	H	×
<i>intermediate</i>	×	L	H	H	H	L	H	×
<i>final output</i>	×	L	H	H	H	L	H	×

The derivation above shows a representative instance of ternary spreading, (52b). In (52a), unbounded spreading occurs because there is only one underlying H tone in the word. To model this mapping, our first FST does two things: ternary spreading, and then coding the right edge of the word to indicate that a second underlying H was not encountered.

When comparing (52a,b), the only difference in these two forms is the word-final tone; it is a L tone in (18a), but a H tone in (52b). Thus, the first five tones encountered in the derivation below are output exactly as in (53)–(55). We skip these steps in the first FST below, since they are identical up to the tape shown in (55). In (58), the sixth tone read from the input tape is a L tone and another H tone is output.

- (58) Copperbelt Bemba inner, left-subsequential FST, stage 4: L → H, again

<i>input</i>	×	L	H	L	L	L	L	×
<i>output</i>	×	L	H	λ	λ	H	H	

The final step for this first FST, in (59), crucially involves the word boundary symbol. Upon reading input ×, the transducer outputs HL×. As a result, the input is output neither with ternary (LHHHLL) nor with unbounded spreading (LHHHHH), but instead with an illicit sequence with quaternary spreading.

- (59) Copperbelt Bemba inner, left-subsequential FST, stage 5: × → HL×

<i>input</i>	×	L	H	L	L	L	L	×
<i>output</i>	×	L	H	λ	λ	H	H	HL×

In (59) we saw that the second FST, upon reading a word-final H, outputs all of the first FST’s intermediate output symbols without modification. In (60), though, the second FST reads a word-final L and ‘waits’. A word-final L tone may correspond to one of two types of words in our analysis: words with no H tones, or words with a single H, which triggers unbounded spreading.

- (60) Copperbelt Bemba outer, right-subsequential FST, stage 1: one-symbol wait

<i>initial input</i>	×	L	H	L	L	L	H	×
<i>intermediate</i>	×	L	H	H	H	H	L	×
<i>final output</i>							λ	×

By waiting one symbol, the FST is able to determine which input form is being read. If the input has no H tones, then the second symbol will also be a L tone. However, if the initial input string contains a single H, the context for unbounded spreading, the second FST should read a penultimate H tone. This is what the transducer reads in (61) below. The word-final HL sequence is output as HH.

- (61) Copperbelt Bemba outer, right-subsequential FST, stage 2: word-final HL  $\rightarrow$  HH

<i>initial input</i>	×	L	H	L	L	L	H	×
<i>intermediate</i>	×	L	H	H	H	H	L	×
<i>final output</i>						HH	$\lambda$	×

The second FST now outputs each L tone as a H tone until it encounters another input H tone. After reading another H tone, the FST outputs the rest of the input string faithfully and the derivation is completed.

- (62) Copperbelt Bemba outer, right-subsequential FST, stage 3: left edge reached; computation ends

<i>initial input</i>	×	L	H	L	L	L	H	×
<i>intermediate</i>	×	L	H	H	H	L	H	×
<i>final output</i>	×	L	H	H	H	HH	$\lambda$	×

In both the Tutrugbu and Copperbelt Bemba analyses presented in this section, we have exploited the surface phonotactics of each language to introduce a syntagmatic form of mark-up that is both alphabet- and length-preserving. In Tutrugbu, we used a prefixal [–high, +ATR] vowel in forms with initial-syllable [+high] vowels to indicate that both blocking conditions had been met. In Copperbelt Bemba, we used an illicit HL sequence at the right edge of the word to indicate that the structural description for unbounded spreading had been met. In both of these examples, we used the first FST to share information via these final output-illicit sequences to mark up the string for the second FST.

If it is not clear at this point that similar strategies are exploitable for, as far as we know, any such unbounded circumambient pattern, let us reinforce our claim with one more brief example. Earlier we introduced a hypothetical language, Zurkish, with an unattested ‘sour grapes’ rounding harmony pattern. In Zurkish, a [–high] vowel anywhere to the right of a [+round] root will block harmony on all intervening [+high] vowels. One possible way to model this is shown below. In (63), our first, left-subsequential FST operates exactly like the sole FST described for actual Turkish in (2), spreading [+round] to all [+high] vowels following a [+round] root. In (63), the first FST outputs a string that would be grammatical in actual Turkish, with myopic rounding harmony from the root up to the blocking [–high] vowel.

- (63) Zurkish inner, left-subsequential FST

<i>input</i>	×	j	y	z	√	l	i	g	i	n	d	e	×
<i>output</i>	×	j	y	z	√	l	y	g	y	n	d	e	×

The second FST proceeds from right to left looking for a [–high] vowel. Upon reading an input [–high] vowel, it spreads [–round] regressively up to the root boundary. This is shown below in (64), and like the languages analyzed above, this complex pattern is analyzable without the introduction of new symbols or length-altering coding tricks.

- (64) Zurkish outer, right-subsequential FST

<i>initial input</i>	×	j	y	z	√	l	i	g	i	n	d	e	×
<i>intermediate</i>	×	j	y	z	√	l	y	g	y	n	d	e	×
<i>final output</i>	×	j	y	z	√	l	i	g	i	n	d	e	×

To sketch the general case: consider a properly non-deterministic composition of opposite-direction subsequential functions  $\tau = O \circ I$  where  $I : X^* \rightarrow Y^*$  uses a set of extra mark-up symbols  $M = Y - X$  to



behave as unbounded lookahead for  $O$ . If there are unused (‘phonotactically illicit’) substrings  $usv$ ,  $u, v \in X^*$  and  $s \in X$  such that  $usv$  doesn’t appear in any strings that are both in the codomain of  $I$  and the domain of  $O$ , then these substrings are potentially usable by alternative functions  $O'$  and  $I'$  as non-length-increasing codes for symbols in  $M$ : instead of depositing a mark-up symbol  $y \in M$  (or a length-increasing code for one) into the intermediate representation that is the input to the second function in the composition,  $I'$  can rewrite an occurrence of a symbol  $x \in X$  in context  $u\_v$  as a symbol  $s \in X$  that would otherwise never occur in that phonotactic context. The presence of  $s$  in the environment  $u\_v$  is then unambiguously decipherable by  $L'$  as information put there by a lookahead pass.

We have just demonstrated that Tzutugbu ATR harmony, Copperbelt Bemba tone spreading, and Zerkish sour grapes can be analyzed using the composition of two transducers that do not violate [Heinz and Lai \(2013\)](#)’s restrictions on weakly deterministic functions even though these two patterns exhibit the unbounded circumambience of the non-deterministic class of string-to-string mappings. In the next subsection we argue that weakly deterministic functions must be constrained such that they are ‘non-interacting’. Moreover, we demonstrate that ‘interaction’ subsumes both of the mark-up techniques discussed in [Heinz and Lai \(2013\)](#), and provides a basis for a definition of weak determinism that matches the intended scope of [Heinz and Lai](#).

### 4.3 (Non-)interacting compositions and weak determinism

Recall that [Heinz and Lai \(2013\)](#) define the weakly deterministic regular functions by starting with [Elgot and Mezei’s \(1965\)](#) strategy for defining regular relations — compositions with intermediate mark-up — and specifically exclude those compositions that use either of the two strategies for mark-up discussed above. The last subsection showed that there is at least one other strategy not reducible to either of those two. Rather than continuing to discover new strategies and modifying [Heinz and Lai](#)’s tentative definition to stipulatively ban each such strategy, we suggest defining the weakly deterministic regular functions by banning the general property or behavior shared by all three of these strategies, as well as others that may yet be discovered.

An informal statement of what underlies all three of the strategies for mark-up is that the observable behavior (output) of the second (‘outer’) function  $O$  to apply in such compositions  $\tau = O \circ I$  can depend on or be affected by the behavior (output) of the first (‘inner’) function  $I$  to apply. That is, in each such strategy, exactly what kind of change  $L$  makes to an input  $w$  to  $\tau = O \circ I$  can in general depend on which changes  $I$  makes to  $w$  in producing an intermediate representation that is the input to  $O$ . Informally, then, we suggest defining weakly deterministic regular functions as those regular functions which can be defined with a *non-interacting* composition of subsequential functions – i.e. where changes made by the outer function do not change or depend on changes made by the inner function. To formalize this intuition, we first introduce some background concepts and notation for describing the behavior and interaction of functions in a simpler, more general setting, and then relate them to string functions and transducer composition.

#### 4.3.1 Mutation points and interacting function compositions

Consider a function  $f: X \rightarrow X$ . For any such function, we distinguish those elements of  $f$ ’s domain of definition –  $\text{dom}(f)$  – where  $f$  behaves exactly like the identity function  $\text{id}_X(x) = x$  from those elements of  $f$ ’s domain that  $f$  changes. The first set of elements – compactly denoted by  $\text{fix}(f)$  – constitutes  $f$ ’s *fixed points* or *fixset*; unaware of a convenient existing term for the second set, we refer to its elements as  $f$ ’s *mutation points* or *mutation set* and denote it by  $\mu(f)$ :

$$(65) \quad \begin{aligned} \text{a. } \text{fix}(f) &= \{x \mid x \in \text{dom}(f) \wedge f(x) = x\} \\ \text{b. } \mu(f) &= \{x \mid x \in \text{dom}(f) \wedge f(x) \neq x\} \end{aligned}$$

For example, suppose we have a function on the natural numbers  $\text{doubleEven}: \mathbb{N} \rightarrow \mathbb{N}$  defined as

$$(66) \quad \text{doubleEven}(x) = \begin{cases} 2x & \text{if } x \bmod 2 = 0 \\ x & \text{otherwise} \end{cases}$$

Its mutation set, then consists of all and only the even natural numbers:

$$(67) \quad \mu(\text{doubleEven}) = \{x \in \mathbb{N} \mid x \bmod 2 = 0\} = \{2, 4, 6, 8, \dots\}$$

Its fixset consists of all other natural numbers  $\{1, 3, 5, \dots\}$ .

Equipped with these two notions, we can define a non-interacting composition of two functions as follows:

(68) Given  $f: X \rightarrow X$  and  $g: X \rightarrow X$ ,  $f$  and  $g$  do **not** interact when composed as  $g \circ f$  iff both

- $\forall x \in \mu(f), g \circ f(x) = f(x)$
- $\forall x \in \mu(g), g \circ f(x) = g(x)$

i.e.  $g$  is a *left-identity* of  $f$  for all of  $f$ 's mutation points and  $f$  is a *right-identity* of  $g$  for all of  $g$ 's mutation points.

That is, in a non-interacting composition, no change  $f$  can cause to an input  $x$  in  $f(x)$  can be rewritten by  $g$  and any change  $g$  would cause to an input (as  $g(x)$ ) either isn't interfered with by  $f$  in  $g \circ f$ , or  $f$  makes the same change  $g$  would have, given the opportunity to apply first or in isolation. From the definition of composition and the restrictions the definition of non-interaction places on composition, it follows that non-interacting composition is a *mutation-conserving* operation on pairs of functions:

(69) For any non-interacting composition  $g \circ f$ ,  $\mu(g \circ f) = \mu(g) \cup \mu(f)$ .

That is, for any non-interacting composition  $g \circ f$ , every change made by  $f$  is a change made by  $g \circ f$ , every change made by  $g$  is a change made by  $g \circ f$ , and there are no other changes that  $g \circ f$  makes.

While a more detailed and rigorous discussion of the properties of function interaction is a topic for future work, we conclude this introduction to defining interaction with some observations about ordering alluded to in §2.1. First, recall that composition is *ordered* and that interaction in  $g \circ f$  does not in general entail that  $f \circ g$  involves interaction. However, if neither of two functions  $f, g \in X \rightarrow X$  is a partial inverse of the other<sup>15</sup> and  $g \circ f = f \circ g$ , then this is a sufficient condition for concluding that *both* orders of composition are non-interacting. If either is a partial inverse of the other or  $g \circ f \neq f \circ g$ , then this is sufficient to conclude that at least one composition order involves interaction.

As an example, suppose we have a function  $\text{doublePrime}(x): \mathbb{N} \rightarrow \mathbb{N}$  defined as

$$(70) \quad \text{doublePrime}(x) = \begin{cases} 2x & \text{if } \text{prime}(x) \\ x & \text{otherwise} \end{cases}$$

There are four possible compositions involving  $\text{doubleEven}$  and  $\text{doublePrime}$ . In both of the compositions  $\text{doubleEven} \circ \text{doubleEven}$  and  $\text{doubleEven} \circ \text{doublePrime}$ , the outer function interacts with the inner function at every single mutation point of the inner function and nowhere else: there are *no* mutation points of the inner function that the outer function behaves as the identity for. In the composition  $\text{doublePrime} \circ \text{doublePrime}$ , every mutation point of the inner function is left unchanged by the outer function, and every mutation point of the outer function and every change the outer function would make to each mutation point is changed or made in exactly the same way by the inner function; accordingly, this is a non-interacting composition. Finally, in the composition  $\text{doublePrime} \circ \text{doubleEven}$ , every mutation point of the inner function is left unchanged by the outer function and every mutation point of the outer function is either unchanged by the inner function or (in the case of 2) the inner function does exactly what the outer function would; accordingly, this is also a non-interacting composition.

<sup>15</sup>I.e., neither function 'undoes' any changes made by the other.

### 4.3.2 Mutation-conserving compositions of string functions and weak determinism

The definition and examples of (non-)interaction given so far have been easy to describe and reason about, but have not been specific to the context of transducers, string functions, or phonology. The key difference between functions exemplified so far and string functions is that, in addition to determining *whether* a string  $w$  is changed by a string function  $f$  (i.e. whether  $w \in \mu(f)$ ), we can also describe *where* a string  $w$  is changed by  $f$ . Just as two functions  $f, g$  (not necessarily on strings) interact when composed iff the composition  $g \circ f$  does not conserve the mutation points of the two functions when considered in isolation, we intuitively want a finer-grained notion of interaction we can apply to the composition  $g \circ f$  of two string functions that identifies such a composition as interacting in this more specific sense iff the composition does not conserve the changes to particular locations in the string that the two functions would make in isolation. To distinguish the more general notion of interaction from this location-specific notion of interaction, we suggest the term FACTOR-INTERACTION. Note, however, that while factor-interaction entails interaction, the reverse is not the case: two string functions that only ever operate on non-overlapping regions of a string cannot (by definition) factor-interact but could, in general, interact. Throughout the rest of the paper we almost exclusively consider string functions and are only interested in factor-interaction, so we abbreviate ‘(non-)factor-interaction’ as ‘(non-)interaction’ henceforth.

The easiest kind of interaction among string functions to see involves ‘lined up’ pairs of output functions from composed transducers  $\tau_1 \circ \tau_0$ . For example, recall the tape representation of the full derivation of  $[\text{X} \text{ɪ} \text{b} \text{a} \text{d} \text{ɪ} \sqrt{\text{w}} \text{u} \text{X}]$  from (50), repeated in (71) below for convenience:

(71)	<i>initial input</i>	$\text{X}$	$\text{ɪ}$	$\text{b}$	$\text{a}$	$\text{d}$	$\text{ɪ}$	$\sqrt{\text{w}}$	$\text{u}$	$\text{X}$
	<i>intermediate</i>	$\text{X}$	$\text{ɪ}$	$\text{b}$	$\text{e}$	$\text{d}$	$\text{ɪ}$	$\sqrt{\text{w}}$	$\text{u}$	$\text{X}$
	<i>final output</i>	$\text{X}$	$\text{ɪ}$	$\text{b}$	$\text{a}$	$\text{d}$	$\text{ɪ}$	$\sqrt{\text{w}}$	$\text{u}$	$\text{X}$

In the fourth column from the left, the incremental output function of the first FST changes an input symbol,  $/\text{a}/ \rightarrow |\text{e}|$ , and then the incremental output function of the second FST proceeds to overwrite (‘interfere’ or ‘interact’ with) the change made by the first FST,  $|\text{e}| \rightarrow [\text{a}]$ . More generally, however, a change made by the inner function in a composition can cause the behavior of the outer function to change somewhere else — a relationship akin to ‘feeding’ or ‘bleeding’ on environment (McCarthy 1999, Baković 2011). To illustrate this and to formalize the richer notion of ‘changes’ we can associate with string functions, we first introduce a few toy string functions defined by rewrite rules and an example input string:

- (72)
- $s = cad$
  - $\rho_A: c \rightarrow d/\_\text{a}$
  - $\rho_B: a \rightarrow b/c\_\text{d}$
  - $\rho_C: d \rightarrow c/da\_\$
  - $\rho_D: d \rightarrow a/a\_\$

If  $\tau_A, \tau_B$ , etc. represent the functions corresponding to  $\rho_A, \rho_B$ , etc., then intuitively, the following compositions (among others) involve interaction

- (73)
- $\tau_B \circ \tau_A$  involves bleeding on environment.
    - $\tau_B(cad) = cbd$ , but  $\tau_B \circ \tau_A(cad) = \tau_B(dad) = dad$
  - $\tau_C \circ \tau_A$  involves feeding on environment.
    - $\tau_C(cad) = cad$ , but  $\tau_C \circ \tau_A(cad) = \tau_C(dad) = dac$

In contrast,  $\tau_D(cad) = caa$ ,  $\tau_A(cad) = dad$ , and the composition  $\tau_D \circ \tau_A = \tau_D(dad) = daa$  involves no interaction.

To capture these intuitions, we associate each string-to-string function  $\tau: X^* \rightarrow Y^*$  (with  $X \subseteq Y$ ) and potential input  $w = x_1 x_2 \dots x_j \dots x_n$  to  $\tau$  with the set of uniquely identifiable *factors* of  $w$ <sup>16</sup> that  $\tau$  would change in computing  $\tau(w)$  and information about how  $\tau$  would change them. We call this set of changed factors (and information about changes) ‘the  $\mu$ -factors of  $\tau$  in  $w$ ’ or ‘the  $\mu$ -factors of  $w$  with respect to  $\tau$ ’ and denote it with  $\mu(\tau, w)$ . Equipped with this information, we can calculate and compare for any composition of string functions  $g \circ f$  and input string  $w$

- (74) a.  $\mu(f, w)$   
 b.  $\mu(g, w)$   
 c.  $\mu(g \circ f, w)$

Echoing the more general definition, then, non-interacting compositions of string functions will be defined as all and only those compositions where the  $\mu$ -factors of the composition are identical (for all input words) to the union of the  $\mu$ -factors of the two component functions considered in isolation. In other words, non-interacting compositions of string functions consist of all and only mutation-conserving compositions.

To define the  $\mu$ -factors of a string function defined by a transducer, we (at present) require that the transducer be

- (75) a. *sequential*  
 b. *synchronous* (letter-to-letter)  
 c. equipped with *origin semantics*.

A sequential transducer can be defined as in the definition of a subsequential transducer given in §2.1, except a sequential transducer lacks a function  $\sigma: F \rightarrow \Delta^*$  mapping final states to strings – or has a trivial one mapping every state to  $\lambda$ , the empty string.<sup>17</sup> Synchronous transducers are distinguished from asynchronous transducers by their output function  $\omega$ : in synchronous transducers it maps each input symbol to (at most) a single output symbol, whereas the output function of asynchronous transducers can map each input symbol to a string of multiple output symbols. A transducer  $\tau$  with origin semantics (Bojány 2014) applied to an input string  $w = x_1 x_2 \dots x_j \dots x_n$  returns not only an output string  $\tau(w) = w' = y_1 y_2 \dots y_i \dots y_m$  but also a function  $\mathbb{O}: \mathbb{Z}^+ \rightarrow \mathbb{Z}^+$  mapping each index  $i$  of the output string  $y_1 y_2 \dots y_i \dots y_m$  to the index  $j$  of the symbol  $x_j$  in the input string that was rewritten to yield  $y_i$ . This information tremendously simplifies bookkeeping and exposition, though we do not expect it to be strictly necessary for defining interaction between string-to-string functions.<sup>18</sup>

For  $w \in \Sigma^*$  such that  $w = x_1 x_2 \dots x_j \dots x_n$ ,  $f: \Sigma^* \rightarrow \Delta^*$ , and  $f(w) = w' = y_1 y_2 \dots y_i \dots y_m$ , then, we define the  $\mu$ -factors of  $w$  with respect to  $f$  as

$$(76) \quad \mu(f, w) = \{(\mathbb{O}(i), x_{\mathbb{O}(i)}, y_i) \mid y_i \neq x_{\mathbb{O}(i)}\}$$

Applied to the examples above, this yields

$$(77) \quad \begin{array}{ll} \text{a. } \mu(\tau_A, cad) = \{(1, c, d)\} & \text{c. } \mu(\tau_B, \tau_A(cad)) = \emptyset \\ \text{b. } \mu(\tau_B, cad) = \{(2, a, b)\} & \text{d. } \mu(\tau_B \circ \tau_A, cad) = \{(1, c, d)\} \end{array}$$

<sup>16</sup>Every substrings of a string  $w$  is a *factor* of  $w$ .

<sup>17</sup>Subsequential transducers are slightly more expressive than sequential ones. Note, however, that in the context of finite-state models of natural language, the generally guaranteed presence of boundary symbols (e.g. a word-final marker like ‘ $\times$ ’) obviates the need for  $\sigma$  and permits straightforward conversion to an equivalent (if less compact) sequential transducer (Oncina et al. 1993, p. 450).

<sup>18</sup>In subsequent work we expect to offer a definition of interaction for pairs of functions where one or more of the conditions outlined in (75) can be relaxed and to elucidate how interaction can be determined during composition (i.e. without having to reason directly about the full mapping of strings to strings associated with the composition).

- |   |   |
|---|---|
| e. $\mu(\tau_C, cad) = \emptyset$                             | h. $\mu(\tau_D, cad) = \{(3, d, a)\}$                         |
| f. $\mu(\tau_C, \tau_A(cad)) = \{(3, d, c)\}$                 | i. $\mu(\tau_D, \tau_A(cad)) = \{(3, d, a)\}$                 |
| g. $\mu(\tau_C \circ \tau_A, cad) = \{(1, c, d), (3, d, c)\}$ | j. $\mu(\tau_D \circ \tau_A, cad) = \{(1, c, d), (3, d, a)\}$ |

Bleeding compositions are identified by the existence of some input word with one or more missing  $\mu$ -factors, feeding compositions by the existence of an input word with one or more additional  $\mu$ -factors, and non-interacting compositions by a conservation of  $\mu$ -factors for all input words:

- (78) a. Bleeding:  $\mu(\tau_B \circ \tau_A, cad) = \{(1, c, d)\} \neq \mu(\tau_B, cad) \cup \mu(\tau_A, cad)$   
 b. Feeding:  $\mu(\tau_C \circ \tau_A, cad) = \{(1, c, d), (3, d, c)\} \neq \mu(\tau_C, cad) \cup \mu(\tau_A, cad)$   
 c. Non-interacting:  $\mu(\tau_D \circ \tau_A, cad) = \{(1, c, d), (3, d, a)\} = \mu(\tau_D, cad) \cup \mu(\tau_A, cad)$

Accordingly, we offer the following definition of a non-interacting composition of string functions:

- (79) A composition  $g \circ f$  of functions  $f: X^* \rightarrow Y^*$  and  $g: Y^* \rightarrow Z^*$  such that  $X \subseteq Y$  and such that both  $f$  and  $g$  meet the conditions in (75) is NON-INTERACTING (or MUTATION-CONSERVING) iff  $\forall w \in X^*, \mu(g \circ f, w) = \mu(g, w) \cup \mu(f, w)$ .

As an alternative to the definition of weakly deterministic regular functions offered by [Heinz and Lai \(2013\)](#), then, we propose the following:

- (80) A regular function  $\tau: X^* \rightarrow Z^*$  is WEAKLY DETERMINISTIC iff there exist two sequential functions  $I: X^* \rightarrow Y^*$  such that  $X \subseteq Y$  and  $O: Y^* \rightarrow Z^*$  such that  $O$  is either left- or right-subsequential,  $I$  is the other,  $\tau = O \circ I$  and  $O \circ I$  is a non-interacting composition.

This definition excludes any composition that uses extra intermediate symbols, length-increasing codes, phonotactic codes of the sort outlined in §4.2, or any other means by which the behavior of the inner and outer functions in a composition can guide, interfere with, or interact with each other.

#### 4.3.3 A note on rule-ordering and interaction

The derivational nature of the composed functions used throughout this paper does not entail that all phonological phenomena analyzed with rule ordering — e.g. opaque interactions between SPE-style rules — are also non-deterministic.<sup>19</sup> For example, [Chandlee et al. \(2018\)](#) demonstrate that a large number of opaque interactions are analyzable using INPUT STRICTLY LOCAL FUNCTIONS, a highly restricted subset of the subsequential functions. In these cases, by ‘waiting’ an *a priori* known number of states a single input strictly local function can successfully account for multiple interacting processes. Crucially, however, the types of patterns discussed in [Chandlee et al. \(2018\)](#) are all non-iterative in nature. In contrast, as [Jardine \(2016\)](#) notes, in unbounded circumambient processes there is no *a priori* known upper bound on the number of ‘waiting’ states necessary to successfully map inputs to their corresponding outputs, rendering a waiting strategy inapplicable to such cases.

Formally speaking, to determine the complexity of the composition of two specific interacting phonological mappings (whether defined by opaque SPE-style rule interactions or not), the relevant algebraic question is what class the composition of those two particular mappings is in. More generally, this question can be framed or explored as:

- (81) Which subclasses of the regular relations are *closed under composition*?

To summarize existing closure results about classes mentioned in this paper:

- (82) a. Regular relations are closed under composition.

<sup>19</sup>Such orderings *do* necessarily involve interaction, however.

- b. [Elgot and Mezei \(1965\)](#)’s previously mentioned result shows that the set containing both left- and right-subsequential functions is not closed under composition, and that the class’s closure is the set of regular relations.
- c. As noted previously, however, when considered separately each of the left- and the right-subsequential functions are closed under composition ([Mohri 1997](#)).
- d. Input strictly local functions are closed under composition ([Chandlee et al. 2018](#)).

Where are the weakly deterministic regular functions? First, recall that our proposed revised definition of this class defines them as non-interacting compositions of subsequential functions. While a more rigorous exploration of the properties of classes of (non-)interacting string functions (subsequential or not) lies beyond the scope of the present work, we expect future work to show that the weakly deterministic regular functions are not in general closed under composition. The intuition behind this expectation is that the composition of a non-interacting composition of functions with another non-interacting composition is not, in general, non-interacting. To see this, recall from the end of §4.3.1 that  $\text{doublePrime} \circ \text{doublePrime}$  and  $\text{doublePrime} \circ \text{doubleEven}$  are non-interacting compositions, but observe that one of their compositions –  $(\text{doublePrime} \circ \text{doubleEven}) \circ (\text{doublePrime} \circ \text{doublePrime})$  – involves interaction.

#### 4.4 Illustrating interaction: Akan and Tutrugbu

Here we make use of the notion of ‘interaction’ introduced above to shed light on what differentiates unbounded circumambient harmony in Tutrugbu from weakly deterministic harmony in Akan.

In both Akan and Tutrugbu, accounting for harmony requires either a non-deterministic function or (equivalently) a composition of two subsequential functions that proceed in opposite directions. In Akan, one such function proceeds rightward, iteratively rewriting suffix vowels in accordance with the ATR value of the root. The second function proceeds leftward, iteratively rewriting prefix vowels in accordance with the ATR value of the root. Crucially, these two functions *cannot* interact — one only ever operates on suffixes while the other only ever operates on prefixes, and the changes each function makes are dependent *only* on information about the original input string that neither ever alters. In contrast, because the first and second functions in Tutrugbu both operate on prefixes, they are able to – and do, in fact – interact. That is, in Tutrugbu, for the second function to behave deterministically, it *must* have information from an earlier rightwards proceeding lookahead function — specifically, information about the height of the initial-syllable vowel (i.e. at the left end of the input string) — to determine how to spread ATR from the root leftwards. If the second function does not have information about this, there is no way to properly determine the realization of a [–high] prefix vowel. Recall that a similar inheritance of information was necessary for the analysis of Copperbelt Bemba: the second function only ‘knows’ to rewrite word-final [HL] as [HH] because the rightward function uses that particular output string, [HLX], to indicate that the structural description for unbounded spreading has been met.

In sum, a proper notion of ‘interaction’ subsumes the two constraints on subsequential FST composition laid out in [Heinz and Lai \(2013\)](#).

## 5 Discussion

In this section we point to additional examples of non-deterministic segmental phenomena in §5.1, discuss the role of learnability in the typology of phonological patterns in §5.2, and briefly discuss the potential role of substance in formal language theory in §5.3.

### 5.1 Additional unbounded circumambient vowel harmony patterns

We contend that existing literature has been too quick to dismiss the possibility of complex segmental phenomena and to conclude that there exists a relative paucity of complex segmental patterns compared to tonal phenomena, and consequently that conclusions drawn about computational distinctions between segmental and tonal phenomena have been too strong. In support of this contention, we offer five arguments.



First, we argue that a categorical boundary between the complexity of segmental and tonal phenomena delineated by the upper boundary of the weakly deterministic functions is not tenable because the segmental pattern of ATR harmony in Tutrugbu transgresses this boundary. While [Heinz \(2018\)](#) and [Jardine \(2016\)](#) acknowledge two examples of segmental processes that are plausibly unbounded circumambient — Sanskrit *n*-retroflexion ([Ryan 2017](#)) and Yaka height harmony ([Hyman 1998](#)) — they marginalize these examples, claiming that they may not be unbounded. However, in the previous section we demonstrated that regressive ATR harmony in Tutrugbu clearly satisfies [Jardine’s \(2016\)](#) definition of an unbounded circumambient mapping, making it equivalent in complexity to the tonal patterns discussed in [Jardine \(2016\)](#) and difficult to dispute on the same grounds that has led previous work to marginalize Sanskrit *n*-retroflexion and Yaka height harmony.

Second, we argue that claims about the frequency of complex segmental phenomena relative to tonal phenomena are at best premature. As further support for a computational distinction between segmental and tonal processes, [Heinz \(2018\)](#) and [Jardine \(2016\)](#) claim that unbounded circumambient tonal processes are relatively common and equivalent segmental mappings are at most ‘extremely rare.’ To this point, [Jardine](#) adduces nine potential examples of such tonal processes but only two attested unbounded circumambient segmental processes. However, we submit that the difference between two and nine examples is not persuasive evidence of a difference in typological frequency. As [Piantadosi and Gibson \(2014\)](#) contend, outside particular cases of carefully framed questions and statistical reasoning, linguists do not currently nor will soon have descriptions of enough languages to make strong conclusions about what properties are categorically impossible (e.g. non-deterministic segmental phonological patterns) for natural languages to have on the basis of simple comparison of frequency counts. Even when some properties have an observed count of zero, such claims are not justifiable because a similar set/subset problem that children face in attempting to infer properties of a particular natural language from a limited number of positive examples is also faced by linguists in inferring properties of all human languages from an even smaller number of positive examples.

Third, even if the foregoing were not the case, of the nine tonal patterns marshalled as evidence in [Jardine \(2016\)](#), three of these (Digo, Xhosa, and Zulu) are not actually unbounded circumambient (see [Figure 10](#) and the immediately preceding text in the [Appendix](#) for an analysis of Digo). Fourth, besides Sanskrit *n*-retroflexion, Yaka height harmony, and Tutrugbu ATR harmony, we have identified at least six additional examples of unbounded circumambient segmental patterns. One of these is the previously-mentioned harmony pattern in Tafi ([Bobuafor 2013](#)), identical to the pattern in closely-related Tutrugbu. We discuss these patterns not to make typological claims about the frequency of these patterns relative to similarly complex tonal processes, but instead to demonstrate that they are not atypical and must be dealt with seriously and not swept aside as outliers. We briefly discuss one of these patterns below, and point to several others. Fifth and finally, we note that there are even reasons to believe that segmental phonology requires the full expressivity of regular relations — or more. We now discuss these final arguments in more detail.

Besides Tutrugbu and Tafi, Turkana also features a vowel harmony pattern that requires non-determinism to describe ([Dimmendaal 1983](#), [Baković 2000](#)).<sup>20</sup> In Turkana, ATR spreads bidirectionally from the root, as shown below in (83). Affix vowels generally alternate in accordance with the ATR value of root vowels, but the behavior of /a/ differs depending on its prefix vs. suffix status: prefixal /a/ is opaque to harmony and does not alternate (as shown by the gerundial prefix in these examples), while suffixal /a/ alternates between the [+low, –ATR] vowel [a] and the [–low, +ATR] vowel [o], with [a] occurring after [–ATR] roots and [o] occurring after [+ATR] roots (as shown by the epipatetic vowel in these examples, glossed ‘EPI’).

(83) Turkana ATR harmony

- a. a-k-ɪnɔk-a-kɪn-ɪ ‘GER-K-light fire-EPI-DAT-V’
- b. a-k-ibus-o-kin-ɪ ‘GER-K-drop-EPI-DAT-V’

<sup>20</sup>See also [Noske \(1996, 2001\)](#) for more on Turkana vowel harmony.

In addition to dominant root vowels, there are dominant suffixes in Turkana. Some of these are [−ATR], and when a dominant [−ATR] suffix co-occurs with a [+ATR] root, the realization of the epipatetic vowel between them is neither [+low, −ATR] [a] nor [−low, +ATR] [o], but rather [−low, −ATR] [ɔ]. This is shown in (84). As above, in (84a) the epipatetic vowel surfaces as [o] after the [+ATR] root /ibus/ when no dominant [−ATR] suffix vowel follows. However, in (84b), a dominant [−ATR] suffix occurs to the right of the epipatetic vowel. In this word, the epipatetic vowel surfaces as [ɔ].

- (84) Unbounded circumambience in Turkana ATR harmony
- a. e-ibus-o-kin-ɪ ‘3-drop-EPI-DAT-V’
  - b. ɛ-ibʊs-ɔ-km-a ‘GER-K-drop-EPI-DAT-VOI’

The surface quality of a suffixal low vowel is thus not determinable based solely on the ATR value of the root, which may precede the suffix at any distance, but also depends on the presence or absence of a dominant suffix, which may follow at any distance. In other words, the realization of a suffixal low vowel depends on information a potentially unbounded distance away in both directions. ATR harmony in Turkana is thus unbounded circumambient, and therefore requires non-determinism to describe.

To reinforce this point, consider the propagation of ATR rightward from the root in (83–84). Upon encountering the epipatetic vowel, there are three options: one, map input /a/ faithfully onto output [a] (83a); two, map input /a/ onto output [o] (83b, 84a); and three, map input /a/ onto output [ɔ] (84b). Option one is chosen if the root is [−ATR]. Options two and three, though, depend on both the root value of ATR and the presence of a dominant suffix. In the presence of a [+ATR] root or a dominant [+ATR] suffix, [o] surfaces, but in the presence of a [+ATR] root and a dominant [−ATR] suffix, [ɔ] surfaces. Although the unbounded circumambient dependency differs in certain ways from that in Tutrugbu and Tafi, the realization of a suffix low vowel in Turkana is non-deterministic, as shown in Figures 11 and 12 in the Appendix.

In addition to Sanskrit, Yaka, Tutrugbu, Tafi, and Turkana, we are aware of four other unbounded circumambient segmental patterns, which we briefly list here. The pattern described above for Turkana is also attested in Karimojong, a related Nilotic language (Novelli 1985, Lesley-Neuman 2012), and a similar pattern is found in another Nilotic language, Toposa (Schröder and Schröder 1987). Finally, comparable patterns are attested in the Bantu language Liko (Wit 2015) and in the Dogon language Bondu-So (Hantgan and Davis 2012). In sum, Jardine’s claim that non-deterministic segmental patterns are rarer than their tonal counterparts appears to have been premature.

In fact, there is even evidence that segmental patterns may require strictly *more* expressivity than the non-deterministic regular functions. As Heinz (2018, §6.2.5) notes, optional processes require non-deterministic mappings that are, unlike the Copperbelt Bemba and Tutrugbu patterns, non-functional (recall the discussion at the beginning of §2.3). Optional processes thus require the more expressive class of regular relations. Although Heinz (2018) downplays the significance of these potential counterexamples to the subregular hypothesis, patterns of optionality like those listed in Vaux (2008) and others like iterative optionality in Icelandic umlaut (Anderson 1974) are another source of evidence against any strong claim that segmental phonology is categorically subregular. On top of this, some recent work suggests that segmental phonology can require *more* expressivity than the regular relations: Bowler (2013) demonstrates from a corpus of Warlpiri nouns that disharmonic nouns exhibit optional ‘majority rules’ harmony (Lombardi 1999, Baković 2000), a pattern that numerous linguists have claimed to be unattested, impossible, and pathological, given that it requires even more expressivity than the regular relations (Riggle 2004, Heinz and Lai 2013, Hulden 2017, Heinz 2018, Lamont 2018). Whatever conclusion future work comes to about the significance of these last two pieces of evidence, their present significance is that there is nothing clearly separating the complexity of segmental from tonal phonology and no clear categorical or subregular cap on the complexity of phonology.

The first counterexamples to the subregular hypothesis as articulated in Heinz (2011a, b, 2018) came from the tonal patterns in Jardine (2016), which require the expressivity of non-deterministic regular functions. To uphold the subregular hypothesis, Jardine (2016) adds a qualification: *segmental* phonology is subregular,

but *tonal* phonology is not constrained in the same way. We have noted a comparable number of segmental phenomena requiring the same level of expressivity as (or possibly more than) the tonal patterns in [Jardine \(2016\)](#), adding more counterexamples to any proposed subregular upper bound on the complexity of phonology relative to the existing classes that have been identified and discussed so far.

## 5.2 Explaining the typology of phonological complexity

Existing work on the apparent (sub)regular complexity of phonology has also noted that syntax is clearly more complex, requiring expressivity comparable to that of MILDLY CONTEXT-SENSITIVE grammars ([Joshi 1985](#)). Such work has posited that the putative (sub)regular complexity of phonology and difference in complexity relative to syntax could be explained by distinct learning mechanisms applied to each domain, and that the mechanism applied to phonology has a categorical cap on the complexity of patterns it can learn ([Heinz and Idsardi 2011, 2013, Lai 2015](#)).

The evidence we have presented most directly bears on the exact position of any claimed cap on the (empirically observed) complexity of phonology: it is higher than previously thought. We also believe it raises other questions about the currently offered explanation for the complexity of phonology, and we offer an alternative we believe avoids many of these questions and better explains empirical observations.

First, the current hypothesis that there exists a distinct learning mechanism with an arbitrary categorical cap does not predict or explain why we ought to expect distinct learning mechanisms for (or applied to) phonology vs. syntax. Second, it does not predict or explain why we ought to expect, supposing there is a distinct learning mechanism applied to phonology, that it ought to be categorically capped in what it can learn, capped at a lower level than syntax, or capped at any particular level of complexity. Third, whatever the nature of the complexity bound on what phonological patterns humans can learn, a *capacity* to learn patterns of a certain complexity does not in any way predict or explain what causal processes or mechanisms gives rise to data of any particular complexity level for a learner to observe. Finally, this hypothesis says nothing about why attested phonological patterns are distributed across the lower complexity classes the way that they are. [Chandlee and Heinz \(2018\)](#) note that 95% of phonological analyses in the PBase ([Mielke 2007](#)) are input strictly local, one of the simplest classes of functions in the subregular hierarchy (see Figure 1), and it appears that examples of non-deterministic patterns are far less numerous (though more numerous than previously claimed). As stated, the claim that phonology is subregular — or if our claim that phonology is more expressive than that is accepted, the claim that phonological patterns exist in any particular class — is inadequate to describe their skewed distribution, which heavily favors simpler patterns.

We suggest instead that gradient rather than categorical learning biases underlie the gradient distribution of phonological patterns (or simply ‘patterns’ more generally) across the hierarchy of formal languages and functions on them: simpler patterns are likely easier to learn (or to learn to the same level of accuracy) with the same quantity of data, all else being equal. Together with future work on the causal processes underlying morphophonological change, this does not require any commitment to or explanation of a distinct learning mechanism or a particular categorical cap, and it offers a promising account of the gradient, inverse relationship between complexity and observed frequency that is otherwise in need of explanation.

## 5.3 The role of substance

[Pater \(2018\)](#) critiques [Jardine \(2016\)](#) and, by extension, the pursuit of work in formal language theory as an alternative to existing phonological theories. While noting that formal language theory usefully distinguishes different complexity classes, [Pater](#) contends that this approach lacks the substantive restrictions imposed by constraint-based theories like Optimality Theory ([Prince and Smolensky 2004](#)). Specifically, [Pater \(2018, 154\)](#) proposes that tonal constraints can differ substantively from segmental constraints for reasons “presumably related to the shared phonetic substance of lexical tone and phrasal intonation.”

As we hope to have made clear in the foregoing pages, no clear distinction between tonal and segmental patterns is empirically justified, and [Pater](#)’s specific concern with the analysis of unbounded circumambience

in Copperbelt Bemba is moot. More generally, however, we agree with [Pater \(2018\)](#) that the role of substance is an area for future work in formal language theory. At present, much of this work is completely formal and substance-free. Should phonetically-based generalizations, alongside insights from the study of typology, acquisition, and psycholinguistics, be incorporated into the formalism? (And if so, how can they be?) Furthermore, current work in formal language theory is often segment- rather than feature-based, losing many of the representational insights provided by existing phonological theories. [Jardine \(2016, §4.4\)](#) discusses the relationship between string-based and autosegmental representations. The closest equivalent at present are tier-based representational structures ([Heinz et al. 2011](#), [McMullin 2016](#), [Heinz 2018](#)). Given both the historical successes of enriched representational structures in phonology, [McMullin \(2016, 113\)](#) notes that “[i]t may . . . be desirable to integrate certain aspects of phonological theory into the [tier-based strictly local] account of long-distance phonotactics in order to further limit the range of possible patterns.”

## 6 Conclusion

[Jardine \(2016\)](#) claims that the minimum expressivity required to describe tonal patterns exhibiting *unbounded circumambience* is categorically higher than that needed to represent any patterns of segmental phonology, which are claimed to require at most weakly deterministic regular functions. Existing evidence for potentially unbounded circumambient segmental processes in Yaka and Sanskrit has been minimized and these patterns have been construed as aberrant ([Heinz 2018](#), [Jardine 2016](#)). Empirically, we have shown that ATR harmony in Tutrugbu clearly satisfies the definition of unbounded circumambience according to the criteria laid out in [Jardine \(2016\)](#). Moreover, we have noted a variety of other languages that exhibit similarly complex patterns, providing further evidence that complex segmental patterns should not be marginalized or discounted. Altogether, there is strong evidence that segmental patterns are no less complex than tonal patterns.

In conjunction with these empirical findings, we have shown that the original definition of weakly deterministic regular functions offered by [Heinz and Lai \(2013\)](#) does not properly separate the unbounded circumambient patterns it was intended to exclude from the simpler patterns it was intended to include. We have offered a revised definition of the weakly deterministic regular functions based on a notion of ‘interaction’ between functions. We see these two issues, empirical grounding and formal precision, as crucial to describing and accounting for the typology of complexity in phonology.

## References

- Anderson, S. R. (1974). *The organization of phonology*. Academic Press.
- Baković, E. (2000). *Harmony, dominance and control*. PhD thesis, Rutgers, The State University of New Jersey.
- Baković, E. (2011). Opacity and ordering. In Goldsmith, J. A., Riggle, J., and Yu, A. C. L., editors, *The Handbook of Phonological Theory*, pages 40–67. Wiley Blackwell, Malden, Mass., 2nd edition.
- Bickmore, L. S. and Kula, N. C. (2013). Ternary spreading and the OCP in Copperbelt Bemba. *Studies in African Linguistics*, 42(2):101.
- Bobuafor, M. (2013). *A grammar of Tafi*. LOT: Utrecht.
- Bojányzyk, M. (2014). Transducers with origin information. In *ICALP 2014: Automata, Languages, and Programming*, pages 26–37.
- Bowler, M. (2013). Majority rules effects in warlpiri vowel harmony. unpublished ms., UCLA.
- Chandlee, J. (2014). *Strictly local phonological processes*. PhD thesis, University of Delaware.

- Chandlee, J., Athanasopoulou, A., and Heinz, J. (2012). Evidence for classifying metathesis patterns as subsequential. In *The Proceedings of the 29th West Coast Conference on Formal Linguistics*, pages 303–309.
- Chandlee, J. and Heinz, J. (2012). Bounded copying is subsequential: Implications for metathesis and reduplication. In *Proceedings of the Twelfth Meeting of the Special Interest Group on Computational Morphology and Phonology*, pages 42–51. Association for Computational Linguistics.
- Chandlee, J. and Heinz, J. (2018). Strict locality and phonological maps. *Linguistic Inquiry*, 49(1):23–60.
- Chandlee, J., Heinz, J., and Jardine, A. (2018). Input strictly local opaque maps. *Phonology*, 35(2):171–205.
- Chandlee, J., Jardine, A., and Heinz, J. (2014). Learning repairs for marked structures. In *Proceedings of the 2014 Annual Meeting on Phonology*.
- Chomsky, N. and Halle, M. (1968). *The sound pattern of English*. Harper & Row.
- Dimmendaal, G. J. (1983). *The Turkana Language*. Foris Pubns USA.
- Dolphyne, F. A. (1988). *The Akan (Twi-Fante) language: Its sound systems and tonal structure*. Ghana Universities Press.
- Elgot, C. C. and Mezei, J. E. (1965). On relations defined by generalized finite automata. *IBM Journal of Research and Development*, 9(1):47–68.
- Essegbey, J. and McCollum, A. G. (2017). Initial prominence and Tutrugbu labial harmony. *Ms, University of Florida and University of California-San Diego*.
- Gainor, B., Lai, R., and Heinz, J. (2012). Computational characterizations of vowel harmony patterns and pathologies. In *The Proceedings of the 29th West Coast Conference on Formal Linguistics*, pages 63–71. Citeseer.
- Hantgan, A. and Davis, S. (2012). Bondu-so vowel harmony: a descriptive analysis with theoretical implications. *Studies in African Linguistics*, 41(2):1–26.
- Heinz, J. (2011a). Computational phonology–part I: Foundations. *Language and Linguistics Compass*, 5(4):140–152.
- Heinz, J. (2011b). Computational phonology–part II: Grammars, learning, and the future. *Language and Linguistics Compass*, 5(4):153–168.
- Heinz, J. (2018). The computational nature of phonological generalizations. *Phonological Typology*, 23:126.
- Heinz, J. and Idsardi, W. (2011). Sentence and word complexity. *Science*, 333(6040):295–297.
- Heinz, J. and Idsardi, W. (2013). What complexity differences reveal about domains in language. *Topics in Cognitive Science*, 5(1):111–131.
- 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.
- Heinz, J., Rawal, C., and Tanner, H. G. (2011). Tier-based strictly local constraints for phonology. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers-Volume 2*, pages 58–64. Association for Computational Linguistics.

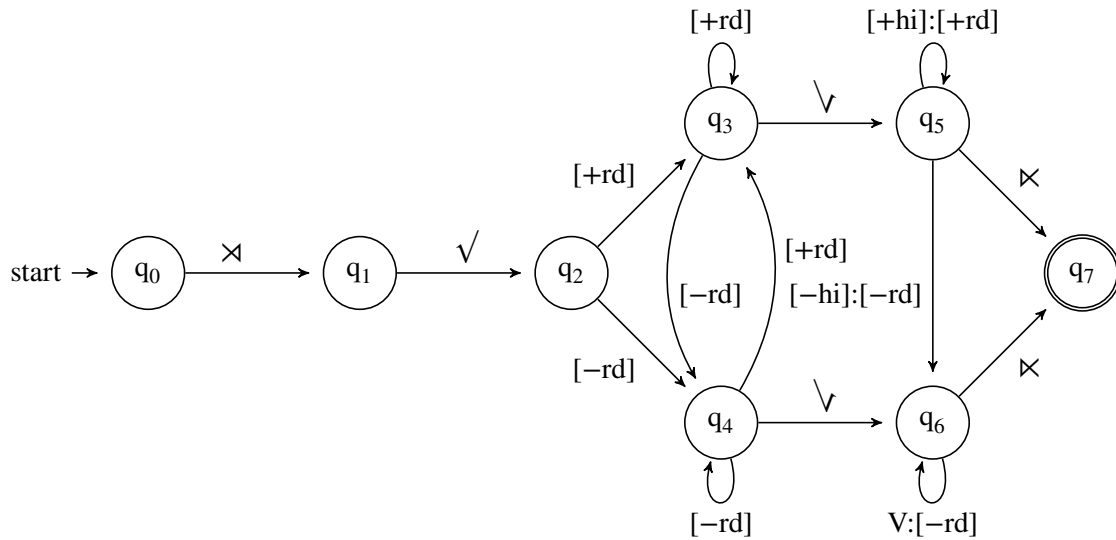
- Hulden, M. (2017). Formal and computational verification of phonological analyses. *Phonology*, 34(2):407–435.
- Hyman, L. M. (1998). Positional prominence and the 'prosodic trough' in Yaka. *Phonology*, 15(1):41–75.
- Jardine, A. (2016). Computationally, tone is different. *Phonology*, 33(2):247–283.
- Johnson, C. D. (1972). *Formal aspects of phonological description*. Mouton The Hague.
- Joshi, A. K. (1985). Tree adjoining grammars: How much context-sensitivity is required to provide reasonable structural descriptions? In Dowty, D. R., Karttunen, L., and Zwicky, A. M., editors, *Natural Language Parsing*, page 206–250. Cambridge University Press, Cambridge.
- Kaplan, R. M. and Kay, M. (1994). Regular models of phonological rule systems. *Computational linguistics*, 20(3):331–378.
- Kisseberth, C. and Odden, D. (2014). Tone. In Nurse, D. and Philippson, G., editors, *The Bantu Languages*, pages 59–71. Routledge.
- Kula, N. C. and Bickmore, L. S. (2015). Phrasal phonology in Copperbelt Bemba. *Phonology*, 32(1):147–176.
- Lai, R. (2015). Learnable vs. unlearnable harmony patterns. *Linguistic Inquiry*, 46(3):425–451.
- Lamont, A. (2018). Majority rule in harmonic serialism. *Poster presented at AMP 2018, San Diego*.
- Laughren, M. (1984). Tone in zulu nouns. In Clements, G. and Goldsmith, J., editors, *Autosegmental studies in Bantu tone*, pages 183–234. Foris.
- Lesley-Neuman, D. (2012). Morpho-phonological levels and grammaticalization in Karimojong: A review of the evidence. *Studies in African Linguistics*, 41(1):99–169.
- Lombardi, L. (1999). Positional faithfulness and voicing assimilation in optimality theory. *Natural Language & Linguistic Theory*, 17(2):267–302.
- McCarthy, J. J. (1999). Sympathy and phonological opacity. *Phonology*, 16(3):331–399.
- McCollum, A. G. and Essegbey, J. (2018). Unbounded harmony is not always myopic: Evidence from Tutrugbu. In *35th West Coast Conference on Formal Linguistics*, pages 251–258. Cascadilla Proceedings Project.
- McMullin, K. J. (2016). *Tier-based locality in long-distance phonotactics: learnability and typology*. PhD thesis, University of British Columbia.
- Mielke, J. (2007). P-base. *A database of sound patterns in 500+ languages*.
- Mohri, M. (1997). Finite-state transducers in language and speech processing. *Computational Linguistics*, 23(2):269–311.
- Nevins, A. (2010). *Locality in vowel harmony*, volume 55. MIT Press.
- Noske, M. (1996). [ATR] harmony in Turkana. *Studies in African Linguistics*, 25(1):61–99.
- Noske, M. (2001). [ATR] harmony in Turkana: A case of faith suffix » faith root. *Natural Language & Linguistic Theory*, 18(4):771–812.



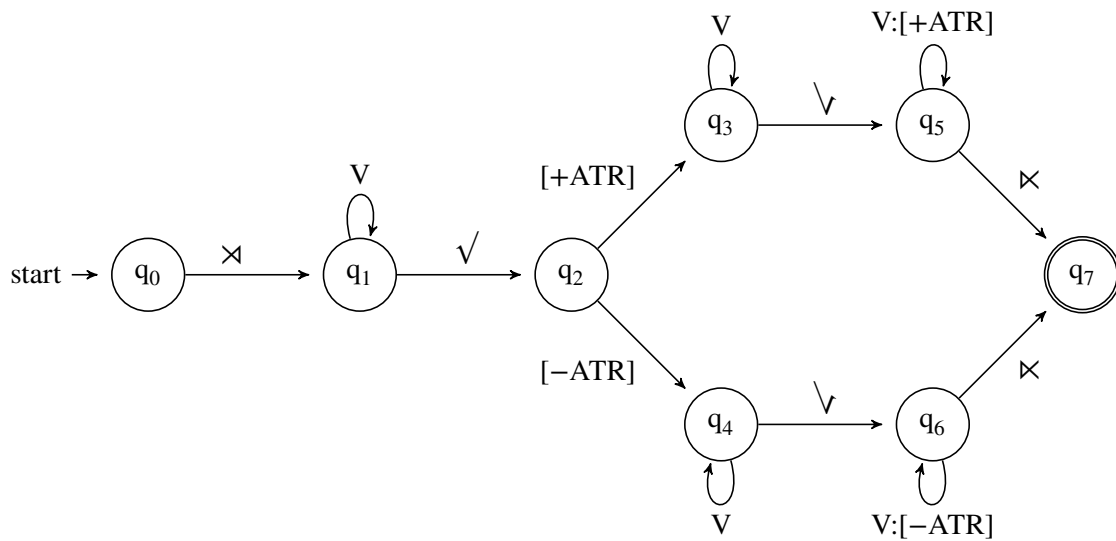
- Novelli, B. (1985). *A grammar of the Karimojong language*, volume 7. Reimer.
- Oncina, J., García, P., and Vidal, E. (1993). Learning subsequential transducers for pattern recognition interpretation tasks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(5):448–458.
- Pater, J. (2018). Substance matters: a reply to Jardine (2016). *Phonology*, 35(1):151–156.
- Payne, A. (2017). All dissimilation is computationally subsequential. *Language*, 93(4):e353–e371.
- Piantadosi, S. T. and Gibson, E. (2014). Quantitative standards for absolute linguistic universals. *Cognitive Science*, 38(4):736–756.
- Prince, A. and Smolensky, P. (2004). *Optimality Theory: Constraint interaction in generative grammar*. Wiley-Blackwell.
- Pullum, G. K. (1976). The Duke of York gambit. *Journal of linguistics*, 12(1):83–102.
- Riggle, J. A. (2004). *Generation, recognition, and learning in finite state Optimality Theory*. PhD thesis, UCLA.
- Roche, E. and Schabes, Y. (1997). *Finite state language processing*. MIT Press.
- Ryan, K. (2017). Attenuated spreading in Sanskrit retroflex harmony. *Linguistic Inquiry*, 48(2):299–340.
- Schröder, H. and Schröder, M. (1987). Vowel harmony in Toposa. *Afrikanistische Arbeitspapiere*, 12:27–35.
- Vaux, B. (2008). Why the phonological component must be serial and rule-based. In Vaux, B. and Nevins, A., editors, *Rules, Constraints, and Phonological Phenomena*, pages 20–60. Oxford University Press, Oxford.
- Walker, R. (2010). Nonmyopic harmony and the nature of derivations. *Linguistic Inquiry*, 41(1):169–179.
- Wilson, C. (2003). Analyzing unbounded spreading with constraints: marks, targets, and derivations. *Unpublished manuscript, UCLA*.
- Wit, G. d. (2015). *Liko phonology and grammar: a Bantu language of the Democratic Republic of the Congo*. LOT, Utrecht.

## Appendix: FSTs and an analysis of tone spreading in Digo

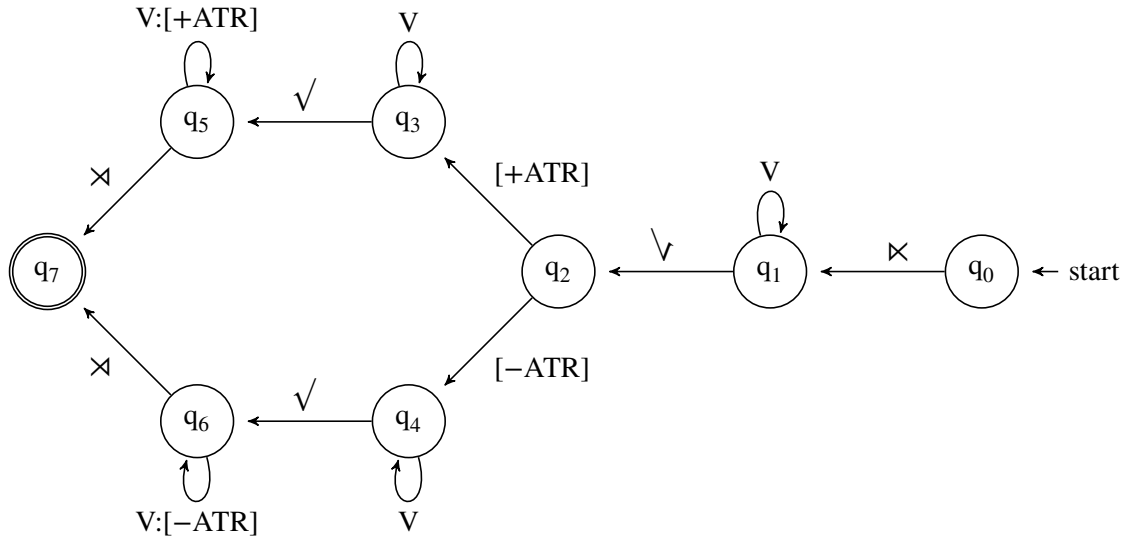
Since consonants do not play a role in the patterns discussed in the paper, all consonants are ignored in the transducers below.



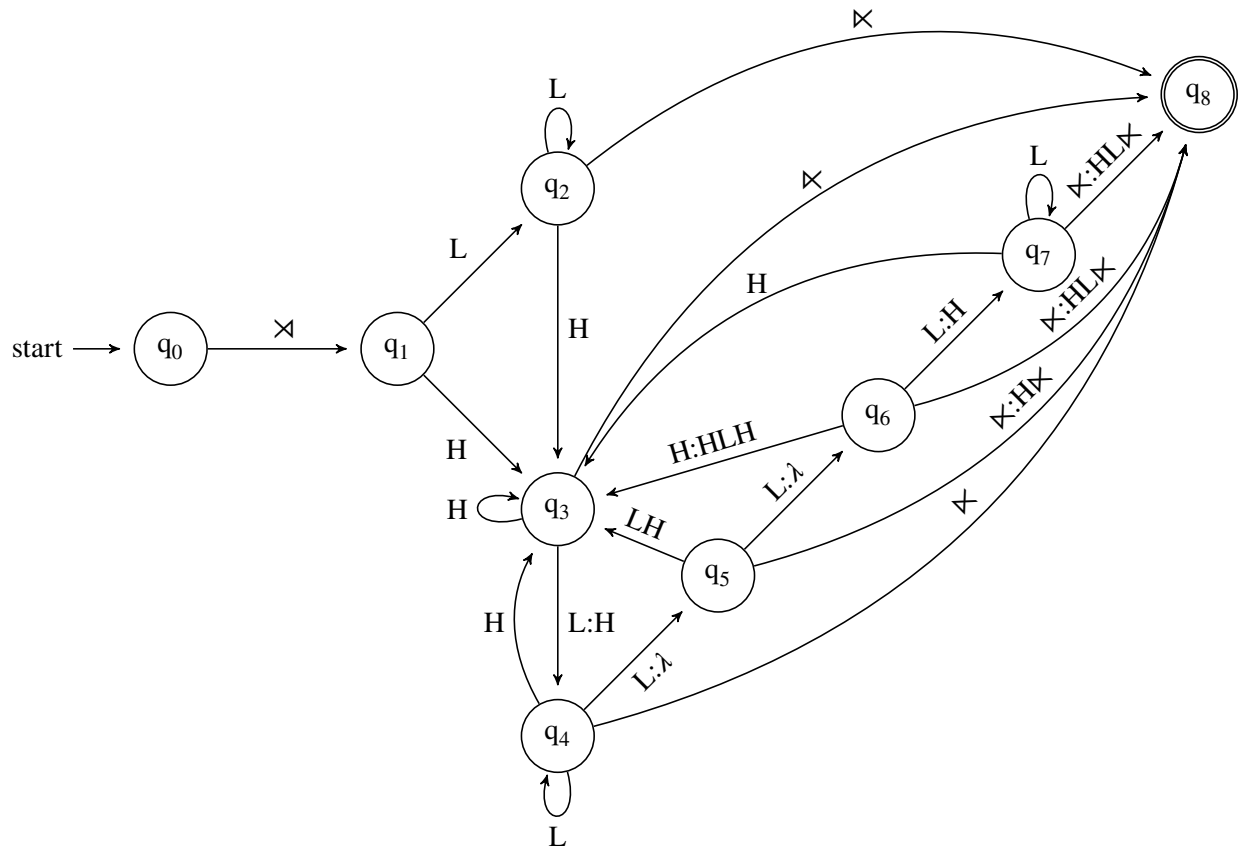
**Figure 2** Left-subsequential FST for Turkish progressive rounding harmony



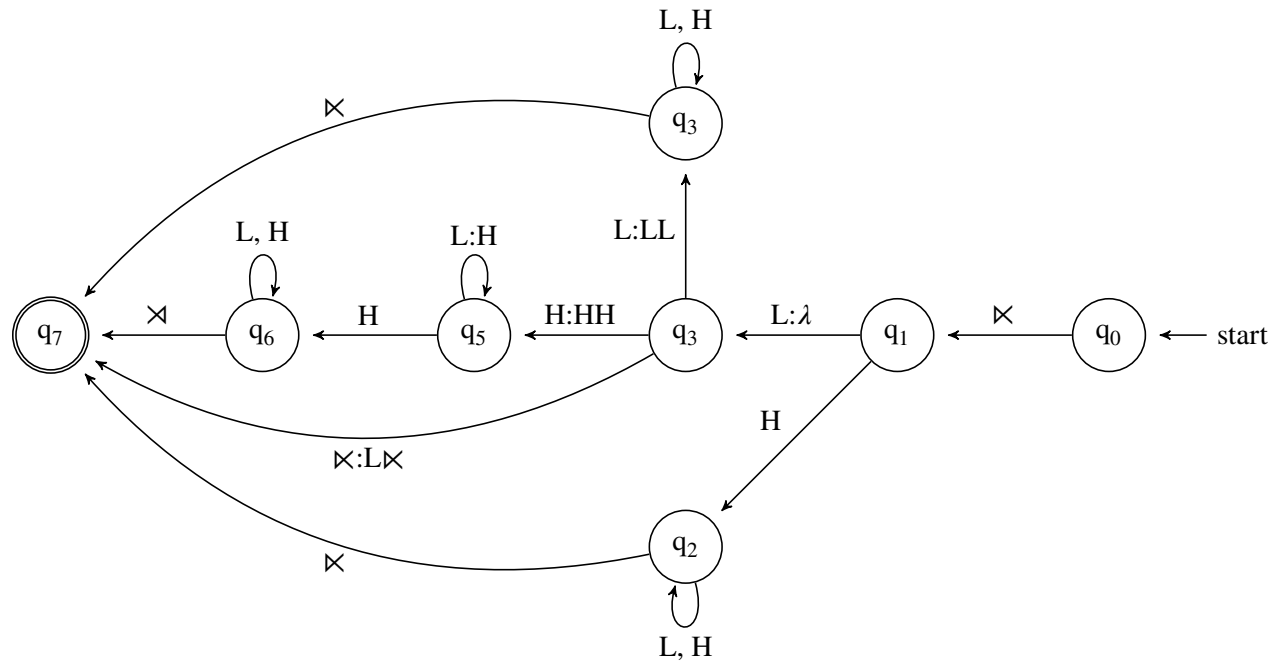
**Figure 3** Left-subsequential FST for Akan



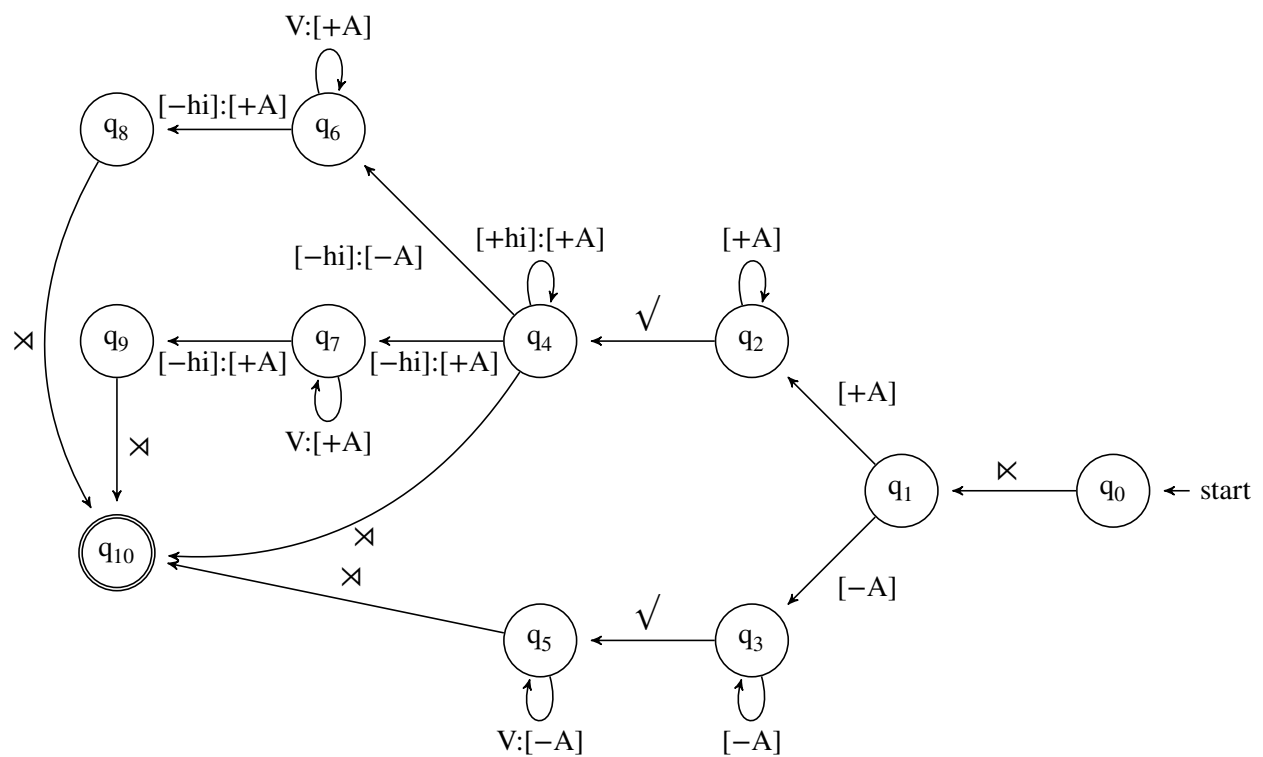
**Figure 4** Right-subsequential FST for Akan



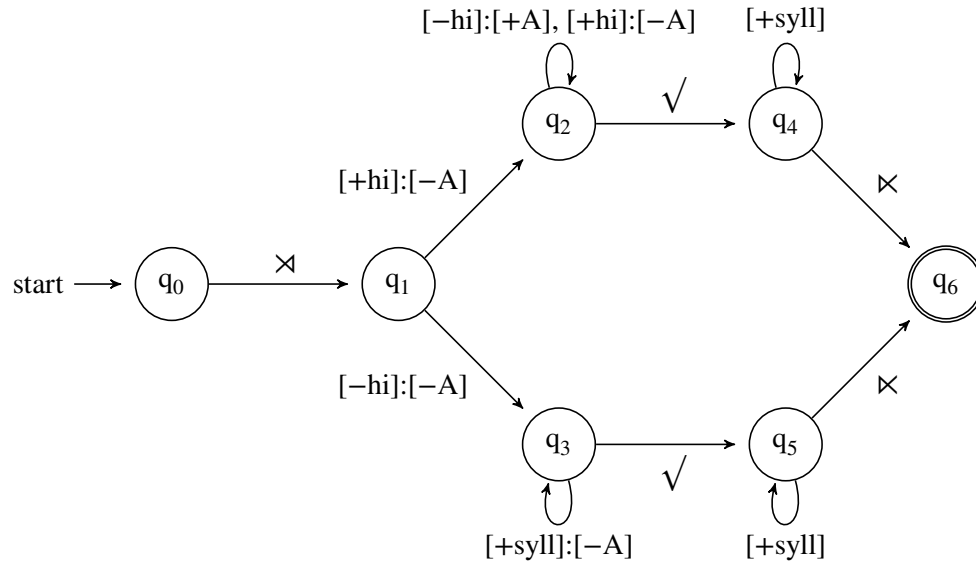
**Figure 5** Left-subsequential FST for Copperbelt Bemba tone spreading



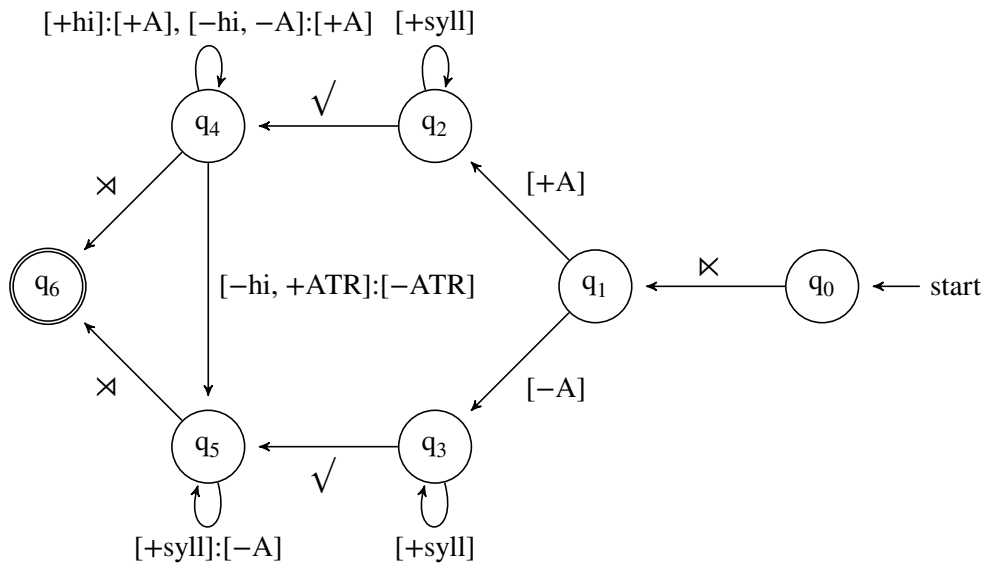
**Figure 6** Right-subsequential FST for Copperbelt Bemba tone spreading



**Figure 7** Non-deterministic FST of ATR harmony in Tutrugbu ( $\sqrt{\quad}$  = root boundary)



**Figure 8** Left-subsequential FST for Tutrugbu ATR harmony



**Figure 9** Right-subsequential FST for Tutrugbu ATR harmony

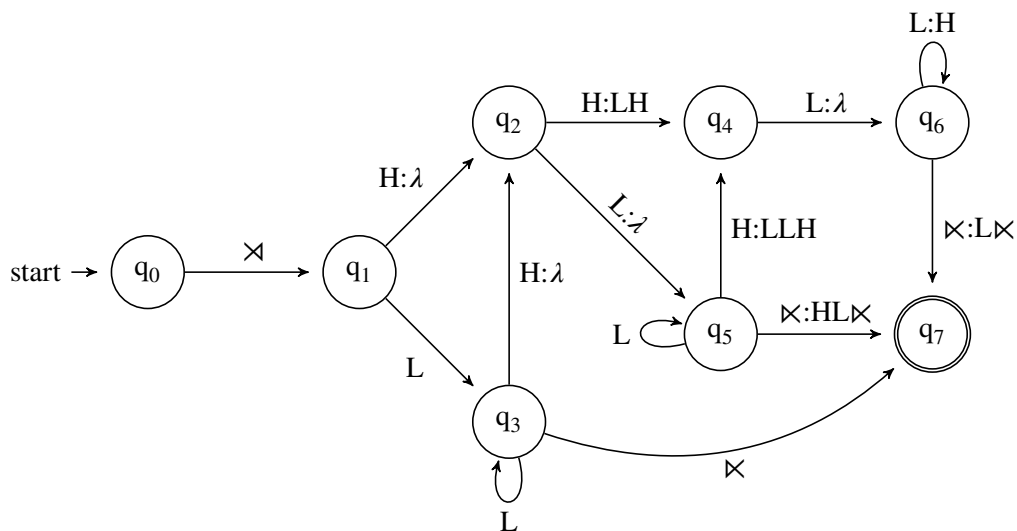
In the Digo data below, a single underlying H shifts to the penult (85b). Note that in (85b) the final two syllables are realized as a rising-falling sequence. Below we treat them as a HL sequence. In a word containing two underlying H tones, the first H shifts to the penult and the second H initiates spreading up to the penult (85c).

(85) Tonal spreading in Digo (underlying H are underlined)

- a. ni-na+tsukur-a “I am taking”
- b. ni-na+a-tsukur-â “I am taking them”
- c. a-na+â-tsúkúr-â “s/he is taking them”

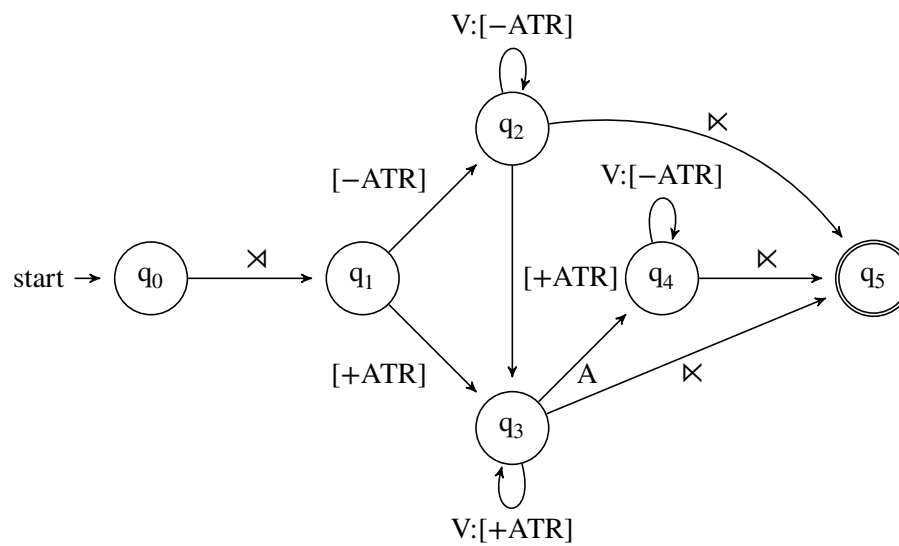
Jardine (2016, §2.2.2) claims that the tonal pattern in Digo is unbounded circumambient, and as a result, non-deterministic. However, the pattern is analyzable as left-subsequential, as we show in Figure 10. The transducer reads the input tape from left to right, and spreads H from the second input H until the penultimate syllable. The analysis only requires ‘waiting’ two symbols upon reading an input H to ensure that the target for tone shifting (if only a single H occurs), or the final target for tone spreading (if two H tones occur) is the penult.

Moreover, we see no convincing evidence that the patterns in Xhosa and Zulu are non-deterministic, either. In both languages, the descriptions note that these patterns are bounded. In Xhosa, (Kisseberth and Odden 2014, p.68) explicitly notes that spreading occurs only when two H tones are adjacent, thus making the triggering context bounded. As for Zulu, (Laughren 1984, p.221) describes the pattern as bounded plateauing (as noted in (Jardine 2016, p.255)). Thus, neither of these patterns appear to qualify as unbounded circumambient.

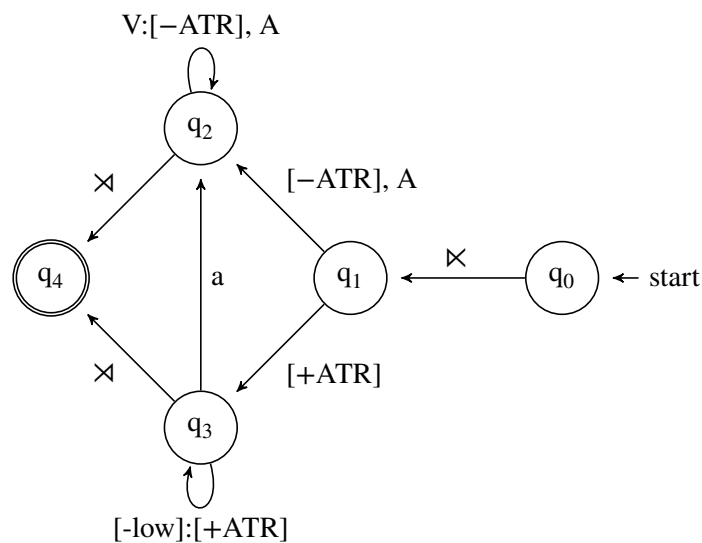


**Figure 10** Subsequential FST of Digo (assuming that the rise-fall sequence on the final 2 TBUs is a HL sequence)





**Figure 11** Left-subsequential FST for Turkana ATR harmony (A = dominant  $[-ATR]$  suffix)



**Figure 12** Right-subsequential FST for ATR harmony in Turkana (A = dominant  $[-ATR]$  suffix)