Weak determinism and the computational consequences of interaction

Eric Meinhardt,* Anna Mai,* Eric Baković,* Adam G. McCollum†
*UC San Diego Linguistics, †Rutgers University Linguistics

Abstract

Recent work has claimed that (non-tonal) phonological patterns are *subregular* (Heinz 2011a, b, 2018, Heinz and Idsardi 2013), occupying a delimited proper subregion of the regular functions — the *weakly deterministic* (WD) functions (Heinz and Lai 2013, Jardine 2016). Whether or not it is correct (McCollum et al. 2020a), this claim can only be properly assessed given a complete and accurate definition of WD functions. We propose such a definition in this article, patching unintended holes in Heinz and Lai's (2013) original definition that we argue have led to the incorrect classification of some phonological patterns as WD. We start from the observation that WD patterns share a property that we call *unbounded semiambience*, modeled after the analogous observation by Jardine (2016) about non-deterministic (ND) patterns and their *unbounded circumambience*. Both ND and WD functions can be broken down into compositions of deterministic (subsequential) functions (Elgot and Mezei 1965, Heinz and Lai 2013) that read an input string from opposite directions; we show that WD functions are those for which these deterministic composands do not *interact* in a way that is familiar from the theoretical phonology literature. To underscore how this concept of interaction neatly separates the WD class of functions from the strictly more expressive ND class, we provide analyses of the vowel harmony patterns of two Eastern Nilotic languages, Maasai and Turkana, using bimachines, an automaton type that represents unbounded bidirectional dependencies explicitly. These analyses make clear that there is interaction between deterministic composands when (and only when) the output of a given input element of a string is simultaneously dependent on information from both the left and the right: ND functions are those that involve interaction, while WD functions are those that do not.

Keywords: phonology, computational phonology, subregular hierarchy, interaction

1 Introduction

Over the past decade, research at the intersection of computational phonology and formal language theory has advanced the SUBREGULAR HYPOTHESIS, which claims that phonological patterns "occupy some area strictly smaller than the regular languages" (Heinz 2011a, p. 147) and that "the study of the typology of the attested transformations [= input-output maps] in the light of the existing categories yields similar conclusions" (Heinz 2018, p. 198). This work has usefully defined various types of subregular functions relevant to the description of phonological transformations, all properly included within the regular functions in terms of their computational expressivity.[1] A recent snapshot of the most significant of these subregular function types, and their hierarchical inclusion relationships with respect to one another, is shown in Fig. 1.

[1] What we are calling 'regular functions' here and throughout are more precisely termed the *rational* functions in the larger context of formal language theory from the last few decades, where the term 'regular' has begun to be reserved for more complex functions that can be expressed by e.g. *two-way* deterministic finite-state transducers; see e.g. Gauwin (2020, Ch. 2). Because we have no need to make use of this distinction, we keep with older usage and with most of the existing literature in computational phonology and persist in using the term 'regular' throughout. We thank a reviewer for advocating that we be upfront about this terminological issue.

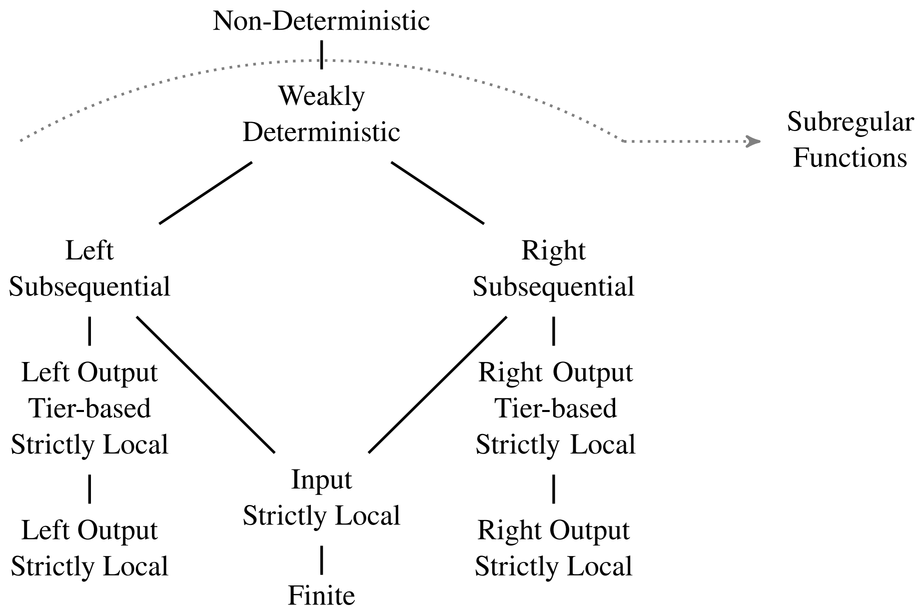

Figure 1 A hierarchy of regular functions, based on Heinz (2018) and Aksënova et al. (2020). All classes define *functional* string transductions, and everything below the non-deterministic regular functions are examples of *subregular* string transductions.

One of the aims of this body of work is to assess more precisely the minimum degree of computational expressivity required to describe some otherwise well-defined phonological patterns, leading to results such as 'local noniterative assimilation patterns are input strictly local' (Chandlee 2014, Chandlee and Heinz 2018), 'unidirectional iterative assimilation patterns are output strictly local' (Chandlee et al. 2015), and 'bidirectional iterative assimilation patterns are weakly deterministic' (Heinz and Lai 2013). These results lead in turn to readily testable hypotheses about the relative complexity and learnability of different types of phonological patterns. Any future results derived from testing hypotheses generated by previous results are of course only as reliable as the results on which the hypotheses are based, which are in turn only as reliable as the formal definitions of any subregular function types relevant to those hypotheses. It is thus imperative that these definitions be as meaningfully accurate as possible.

From the perspective of theory development, good mathematical definitions are insightful, may crystalize linguistic intuitions, and often suggest future directions for research or offer an unexpected unification of previously disparate results and hypotheses. Some examples from the relatively recent literature include Oakden (2020), demonstrating "that two competing feature-geometric models of tonal representation are notationally equivalent"; Jardine et al. (2021), showing "that Q-Theory and Autosegmental Phonology are equivalent in terms of the constraints they can express"; and Bennett and DelBusso (2018), establishing that some apparently different constraint sets in the Agreement by Correspondence framework "not only result in the same extensional predictions; they also generate them in identical ways."

Our focus in this article is on the WEAKLY DETERMINISTIC (WD) functions, at the outer edge of the set of subregular functions depicted in Figure 1. Heinz and Lai (2013) propose the WD class to demarcate the expressivity boundary between amply attested bidirectional vowel harmony patterns and apparently unattested patterns involving e.g. non-myopic spreading (Wilson 2003, 2006) or majority rules spreading (Lombardi 1999, Baković 2000). Jardine (2016) builds on Heinz and Lai (2013) by identifying attested tonal spreading patterns requiring greater expressivity than that afforded by WD functions, leading to a conjecture that the WD class also demarcates the expressivity boundary between non-tonal and tonal phonological patterns. McCollum et al. (2020a) respond by identifying attested non-tonal feature spreading patterns that require crossing the WD expressivity boundary; McCollum et al. (2020b) discuss some examples in greater depth.

The intuition behind Heinz and Lai's (2013) definition is clear enough to allow some work to proceed and some statements to be made about the class of WD functions. However, Heinz and Lai (2013) only conjecture that the formal definition they offer is distinct from the strictly more expressive NON-DETERMINISTIC (ND) functions, and subsequent work has identified pathological and counterintuitive consequences of the letter of that definition that clashes with its intended intuitive scope (Graf 2016, McCollum et al. 2018, O'Hara and Smith 2019, Smith and O'Hara 2019, Lamont et al. 2019), effectively allowing ND functions to satisfy the letter of the definition even when they clash with its intended scope.

The WD–ND boundary is thus clearly an important one to define properly, given that it serves as the dividing line between amply attested patterns of bidirectional vowel harmony and other patterns ranging from the comparatively rare, to the exclusively tonal, to the apparently unattested. We demonstrate how Heinz and Lai's (2013) original formal definition of weak determinism does not accurately delimit the set of functions that it is meaningfully intended to delimit, and we offer a revised definition that accurately delimits the intended set, based on a definition of *interaction* closely related to the notion familiar from the phonological literature. Our empirical focus is on the vowel harmony patterns of two Eastern Nilotic languages, Maasai (Tucker and Mpaayei 1955) and Turkana (Dimmendaal 1983), where the Maasai pattern can be described with a WD function as it is intended to be defined while the Turkana pattern requires the greater expressivity of an ND function and yet is incorrectly classified as WD by Heinz and Lai's (2013) definition.

The article is structured as follows. First, in §2, we provide some necessary formal background for understanding the problems with Heinz and Lai's (2013) definition of weak determinism and our approach to solving them. Then, in §3, we describe and analyze the vowel harmony patterns of Maasai and Turkana, with special attention to the differences that lead them to be on different sides of the intended WD function boundary, exposing both the intent and the limits of Heinz and Lai's (2013) definition. In §4 we introduce a more precise definition of interaction between string functions, clarify its role in adjudicating the WD–ND boundary, and introduce our proposed definition of weak determinism that properly distinguishes the pattern in Maasai from the pattern in Turkana. We end the article with a discussion of the computational and linguistic naturalness of our revised definition of weak determinism in §5 and with a brief conclusion in §6.

2 Formal background

2.1 Scope of relevant claims

It is important to begin here with some clarification of what various results reported in the subregular hypothesis literature mean and what they don't mean. There is understandable confusion here, due to the fact that the subregular hypothesis itself is intended to be a claim about 'phonology' generally — 'phonology is subregular' — and the scope of 'phonology' in this claim is typically not made sufficiently clear. Reported results within this broad hypothesis often reference individual phenomena (e.g., that bidirectional iterative assimilation is weakly deterministic; Heinz and Lai 2013) or substantive classes thereof (e.g., that non-tonal phonology is subregular while tonal phonology is not; Jardine 2016). However, the statement of these reported results should — in themselves — *not* be taken as claims about the complexity of phonology *in toto*.

It is perhaps simplest and most directly useful to clarify this point as follows. Individual phonological processes — however they may be identified (e.g., as the mapping performed by a single rule) — are *not* (typically) the true objects of analysts' claims within this area of computational phonology. Rather, claims about the complexity of phonology are claims about the nature of the computations required to map any single valid underlying representation (UR) to its grammatical surface representation (SR). The fact that an overall UR→SR map may be broken down into (possibly) simpler intermediate steps does not impact this notion of complexity. It is for precisely this reason that results from this approach are of interest to a wide range of phonologists: different analysts may favor different ways of breaking down phonological patterns, but results concerning the minimal necessary computational complexity of the overall UR→SR map are fixed.

To see this more clearly, suppose for example that the functional description of some well-defined set of input-output mappings in a given language is shown to require the expressivity afforded by a defined class of functions F_1. Suppose further that this set of input-output mappings has a phonological analysis in terms of two (or more) independently motivated processes, and that the set of input-output mappings individually described by each of those processes can be shown to require the expressivity afforded by some other defined class of functions F_2 that is strictly less expressive than F_1 ($F_1 \supset F_2$). The F_1 class is still required to express the entire input-output map in such an example; the availability of a phonological analysis whose individual components only require the expressivity afforded by F_2 does not change this fact, regardless of the independent linguistic motivation for the componential breakdown in that analysis.

Indeed, this is what we will see in our discussions of the relevant patterns in Maasai and Turkana in §3. Most if not all previous phonological analyses of both patterns have involved the postulation of (at least) two harmonic feature spreading processes, at least one applying from left to right and another applying from right to left, with some amount of independent linguistic motivation for each. Similarly, our finite-state analyses break both patterns down into two CONTRADIRECTIONAL functions, one reading the input string from left to right and the other from right to left. Each of these functions is SUBSEQUENTIAL, requiring strictly less expressivity than the composition of it with its contradirectional counterpart. The main question that we address in this paper is: what property of these compositions makes some WD and others ND? We argue that the significant difference between finite-state analyses of WD and ND patterns comes down to whether the two contradirectional functions INTERACT. For brevity and exposition, we give a relatively informal definition of interaction in Def. 1, to be followed up by a more formal definition in §4.2.

Definition 1. An input-output map ($X^* \rightarrow Y^*$), broken down into two contradirectional functions l and r, requires that l and r *interact* if there exists some input symbol $x_i \in X$ for which the determination of the corresponding output string $y_i \in Y$ is dependent on information from both sides of x_i, where information from the left is uniquely accessible to l and information from the right is uniquely accessible to r.

As we show in §§3–4 below, the contradirectional functions describing the harmony pattern in Maasai do *not* interact, and this makes their composition a WD function. By contrast, the contradirectional functions describing the harmony pattern in Turkana *do* interact, making their composition an ND function.

2.2 Regular functions, composition, and weak determinism: the view 'from below'

In general, any regular function can be broken down into two contradirectional subsequential functions (Elgot and Mezei 1965, Prop. 7.4, p. 60), with one (the 'outer' function) composed with and applying to the result of the other (the 'inner' function). We refer to all such broken-down regular functions as ELGOT-MEZEI (EM) COMPOSITIONS. Heinz and Lai (2013) define the subregular WD functions in terms of a restriction on EM compositions. In their initial, informal definition, they state that a WD function "can be decomposed into [an inner] subsequential and [an outer, contradirectional] subsequential function *without the [inner] function marking up its output in any special way*" (Heinz and Lai 2013, p. 52, emphasis added). To keep these restricted compositions distinct from EM compositions, we refer to them as HEINZ-LAI (HL) COMPOSITIONS.

There are four main, interrelated things to note about EM compositions and HL compositions.

The first thing to note is that each of the two composand functions is subsequential.[2] A subsequential function can be understood as reading its input string incrementally, and as able to remember some (finite) amount of what it has already read in the string unboundedly far in the past but only to look boundedly far ahead in the string to aid in deciding what to write to its output at any given point. The subsequential class of functions is strictly less expressive than the WD or ND classes, as can be seen in Fig. 1, but the EM or HL

[2] Similar to our discussion of 'rational' vs. 'regular' in fn. 1, we are using the older term 'subsequential' that is also still more prevalent in the computational phonology literature. There is a reasonable argument to be made that the name and meanings of 'sequential' and 'subsequential' should be swapped, and some movement towards this swap has begun in recent years elsewhere. See e.g. Roche and Schabes (1997, pp. 45-46), Bojańczyk and Czerwiński (2018, Ch. 13, fn. 1), or Heinz and Lai (2013, fn. 3).

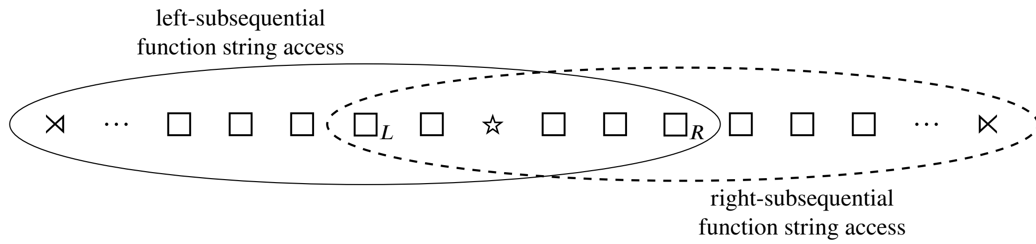

Figure 2 Information access of an arbitrary string from an arbitrary position '☆' by two contradirectional subsequential functions. At position '☆', a right-subsequential function would have access to information about all string positions enclosed within the dashed-line oval, and a left-subsequential function would have access to information about all string positions enclosed in the solid-line oval.

composition of the two contradirectional subsequential functions results in an overall function of one or the other of the strictly more expressive WD or ND classes.

The second thing to note is that the two functions are contradirectional: one is left-subsequential, reading its input string incrementally starting from the left, and the other is right-subsequential, reading its input string incrementally starting from the right.[3] The information in the input string to which each of these functions has access differs accordingly: the left-subsequential function has access to information unboundedly far to the left and bounded lookahead access to the right, while the right-subsequential function has access to information unboundedly far to the right and bounded lookahead access to the left. This difference in information access is schematically represented in Fig. 2.

In this figure, the end points of the string are represented by a boundary on the left (labeled '⋊') and a boundary on the right (labeled '⋉'); the position in the string of focal interest is represented as '☆', and other positions in the string are represented as '□'. When reading the position labeled '☆', the left-subsequential function may have access from the left string boundary '⋊' to some position '$\square_R$' to the right of '☆', and to none of the positions further to the right. When reading the same position '☆', the right-subsequential function may have access from the right string boundary '⋉' to some position '$\square_L$' to the left of '☆', and to none of the positions further to the left.[4] Each function otherwise has unbounded information access from its initial starting point through to the bounded lookahead access position. The information access that each of the functions has is represented by the labeled ovals. String positions enclosed in the solid-line oval and not also in the dashed-line oval are ones that are uniquely accessible to the left-subsequential function (recall Def. 1), while positions enclosed in the dashed-line oval and not also in the solid-line oval are ones that are uniquely accessible to the right-subsequential function. As a succinct and direction-agnostic shorthand, borrowing from the (finite-state) stochastic processes literature, we will refer to the side of the string 'already read' by a given function at any given point in its traversal of the input word as the CAUSAL PAST of the function at that point, and we will refer to the side of the string that the function 'has not yet read' as its CAUSAL FUTURE.

The third thing to note is that the two functions of the composition are (assumed to be) ORDERED with respect to one another: one is the 'inner' function I, applying first, and the other is the 'outer' function O, applying to the result of I. In other words, I and O are COMPOSED, conventionally represented as $O \circ I$.

The fourth thing to note is the additional restriction of HL compositions: that the inner function I does not "[mark] up its output in any special way". The intention of this restriction is to prevent I from providing information to the outer function O about what has previously been read on I's unbounded, causal past

[3]Indeed, Heinz and Lai (2013, p. 52) use "left" and "right" to characterize the two subsequential functions that comprise an HL composition. We use "inner" and "outer, contradirectional" because we find it signals more clearly that the first function to apply may be either left-subsequential or right-subsequential so long as the second function to apply is contradirectional.

[4]In fact, the maximum distance $\square_R$ or $\square_L$ can be from ☆ under any circumstances must be a constant that is known *a priori*. In the figure, this constant is (arbitrarily) 3 for the left-subsequential function and (again, arbitrarily) 2 for the right-subsequential function.

side, which is O's bounded, causal future side. If I is able to provide such information to O, then O would essentially have unbounded access to information on *both* sides of the string, effectively rendering O more expressive than subsequential, and thus rendering the composition $O \circ I$ in general more expressive than an alternative composition in which I did not mark up its output in any special way.

In the discussion leading up to their formal definition of the WD functions, Heinz and Lai (2013, pp. 54–55) explicitly identify just two forms of inner function markup to be precluded. One is the addition of *novel symbols* to the output of the inner function I, symbols that aren't in the original alphabet of I's input. If I is able to write novel symbols, then those symbols can be strategically deployed in such a way that crucial information about I's unbounded side is communicated to the outer function O. A function that does not add novel symbols in this way is ALPHABET-PRESERVING, and a function that *does* add novel symbols is ALPHABET-INCREASING. The other form of explicitly precluded markup in Heinz and Lai (2013) is based on an "important observation" by an anonymous reviewer: "*coding* these new symbols as strings formed over the original alphabet" (Heinz and Lai 2013, p. 55, emphasis in the original). In other words, novel sequences of symbols can also be strategically deployed to communicate crucial information to O about I's unbounded side. A function that effectively encodes new symbols in this way will necessarily be LENGTH-INCREASING: in general, the length of the output string will be no shorter than the input string and sometimes longer as a result of the insertion of substrings that effectively encode new symbols. A LENGTH-PRESERVING function, in contrast, cannot use this particular strategy for encoding novel symbols.

We claim here that what unifies alphabet-increasing and length-increasing inner functions in compositions of contradirectional subsequential functions $O \circ I$ is not that both allow particular forms of markup *per se*, but rather that both enable information from the unbounded, causal past side of I — information that is uniquely accessible to I — to be crucially communicated to O, changing O's behavior by effectively providing O with unbounded access to both sides of the string. When one function changes its contradirectional counterpart's behavior in this way, the two functions interact as defined in Def. 1. Information from the unbounded, causal past side of I is information that is uniquely accessible to I; giving O access to that information (via markup or otherwise) enables the output of a given input symbol to be synergistically dependent both on that information and on information uniquely accessible to O.

Note that, in general, when two functions interact, the order of their composition is important, and — typically — when the order of composition does not matter, they do not interact. Whether any markup introduced by I matters depends on whether it is crucially used by O: if I is alphabet- or length-increasing but O does not make crucial use of that markup, then the composition $O \circ I$ may still be WD in the sense intended by Heinz and Lai (2013). Conversely, if I is alphabet- and length-preserving but nevertheless I and O interact, then their composition is ND — indeed, this is what we argue is the case for the ATR harmony pattern of Turkana in §3.2. In other words, whether or not a composition of two contradirectional subsequential functions is a WD or an ND function depends on whether the two functions interact, and not *per se* on whether the inner function introduces some form of markup.

2.3 Unbounded circumambience, weak determinism, and the view 'from above'

As discussed above, Elgot and Mezei (1965) characterize the ND functions in terms of contradirectional compositions of subseqential functions where the inner function changes the behavior of (= interacts with) the outer function based on information from unboundedly far in the causal future of the outer function. Jardine (2016) offers what is essentially a characterization of properly ND functions in terms of what information is needed from where to determine the output associated with each input symbol. We present a revised definition of weak determinism that centers Jardine's (2016) framing over Elgot and Mezei's (1965).

One of Jardine's (2016) key results is that the minimum level of computational expressivity required to describe UNBOUNDED CIRCUMAMBIENT patterns is that afforded by the class of ND functions, at the outer edge of the class of regular functions — and thus strictly more expressive than the class of WD functions (see now also McCollum et al. 2020a, Hao and Andersson 2019, and Koser and Jardine 2020). A definition of unbounded circumambience, based on Jardine (2016, p. 249), is given in Def. 2.[5]

Definition 2. In an *unbounded circumambient* pattern,

a. the output of at least one input symbol (= a *target*) is dependent on information on both sides of the target; and [= CIRCUMAMBIENT]

b. on both sides, there is no principled *a priori* bound on how far this information may be from the target. [= UNBOUNDED]

A regular function describing an unbounded circumambient pattern can be broken down into a composition of two contradirectional subsequential functions, with each properly identifying critical information at an *a priori* unbounded distance on the side of the target in the function's causal past. Unlike a proper HL composition corresponding to a WD function, however, the inner function of this EM composition must somehow communicate to the outer function whether the critical information is present on the side of the target in the function's causal future in order to correctly describe the unbounded circumambient process. Explained in terms of EM compositions, that is why these patterns require greater computational expressivity than that afforded by the WD functions.

While the literature on weak determinism and its relation to the expressivity of phonology has to date centered on Heinz and Lai (2013)'s initial Elgot and Mezei-derived framing, we suggest instead that Jardine's definition readily motivates the following weakly deterministic parallel to unbounded circumambience:

Definition 3. In an *unbounded semiambient* pattern,

a. the output of any given input symbol (= any *target*) is determined by information *from at most one side* of the target; and [= SEMIAMBIENT]

b. on any determining side, there is no principled *a priori* bound on how far the determining information may be from the target. [= UNBOUNDED]

Crucially, this phenomenological analogue to weak determinism does *not* require that the side of the target containing critical information always be the same from word to word or even the same from target to target within the same word; such restrictions hold for the strictly even less expressive subsequential functions.

We argue in §4 that in addition to correctly capturing the intended scope of Heinz and Lai's (2013) definition of weak determinism, this framing of weak determinism in terms of what information is needed from where is not only equivalent but often simpler to understand and work with than one explicitly grounded in compositions of contradirectional functions. Furthermore, as we will show in §4, our phenomenological identification of unbounded semiambience with weak determinism is computationally well-founded.

Equipped with this background, we can delve into the details of the vowel harmony patterns of Maasai and Turkana in §3. As we'll see, the Turkana pattern is unbounded circumambient, dependent on information on both sides of some targets of harmony, while the Maasai pattern is unbounded semiambient, dependent on information from only one side of every target of harmony (though the side may vary from target to target). Thus, as will be shown in §4, the Maasai pattern is weakly deterministic, and the Turkana pattern is not.

3 Maasai and Turkana

Maasai (Tucker and Mpaayei 1955, Hall et al. 1973, Archangeli and Pulleyblank 1994, McCrary 2001, Guion et al. 2004, Quinn-Wriedt 2013) and Turkana (Dimmendaal 1983, Vago and Leder 1987, Albert 1995, Noske 1996, 2000) are closely related Eastern Nilotic languages. Both languages exhibit patterns of ATR harmony operating over an inventory of nine contrastive vowels, four [+ATR] /i e o u/ and five [−ATR] /ɪ ɛ ɑ ɔ ʊ/.

[5] Jardine's (2016) definition of unbounded circumambience references 'processes' rather than the 'patterns' that we reference in Def. 2. We have made this wording change in our effort to head off misunderstandings about the scope of relevant claims, as discussed in §2.1 — specifically, the misapprehension that unbounded circumambience is (strictly) a property of individual phonological rules.

In both languages, [+ATR] spreads bidirectionally from root and suffix vowels. Spreading vowels in this type of harmony pattern are called DOMINANT, and alternating vowels are called RECESSIVE.

Note that there are four direct harmonic [+ATR]∼[−ATR] pairings among the non-low vowels: i∼ɪ, e∼ɛ, o∼ɔ, and u∼ʊ. Significantly, the [−ATR] low vowel /ɑ/ does not have a direct harmonic counterpart. In certain [+ATR] contexts, underlying /ɑ/ is raised (and also rounded) to become [o], an indirect harmonic relationship dubbed RE-PAIRING in Baković (2000, 2002). This is where the similarities between the Maasai and Turkana vowel harmony patterns end. The key difference between them is the exceptionally dominant behavior of a set of [−ATR] suffix vowels present in the Turkana pattern but absent from the Maasai pattern, with significant consequences for the surface manifestation of re-paired /ɑ/. The presence of these exceptionally dominant [−ATR] suffix vowels in the Turkana case makes the overall Turkana pattern unbounded circumambient and thus non-deterministic, which can only be broken down into an interacting pair of contradirectional subsequential functions. Conversely, the absence of exceptionally dominant [−ATR] vowels in the Maasai case makes the overall Maasai case unbounded *semi*ambient and thus weakly deterministic, which can be broken down into a non-interacting pair of contradirectional subsequential functions.

Note that in the discussion that follows, we assume that (non-exceptional) dominant vowels are underlyingly specified for the spreading value of the harmonic feature, [+ATR], and that alternating recessive vowels are underlyingly specified for the opposite value, [−ATR]. Recessive vowels could alternatively be assumed to be underlyingly unspecified for a value of the harmonic feature, and in fact such an analysis has particular appeal in the case of Turkana because [−ATR] can then be reserved for the exceptionally dominant [−ATR] suffix vowels (Noske 1996, 2000). For the purposes of our argument about the differing levels of expressivity necessary to describe these two patterns, however, nothing hinges on this matter of representation.

3.1 Maasai

Examples of [+ATR] spreading from roots in Maasai are shown in (1) below. Morpheme boundaries are indicated with hyphens, and roots are further indicated with underlining.[6] The contrast between (1a,b) shows that [+ATR] spreads leftward from dominant root vowels to preceding recessive prefix vowels, and the contrast between (1c,d) shows that [+ATR] spreads outward from dominant root vowels both to preceding recessive prefix vowels and to following recessive suffix vowels.

(1) Root-controlled [+ATR] spreading in Maasai

a.	/mɪ-kɪ-rɑɲ/	[mɪ-kɪ-rɑɲ]	'NEG-1PL-sing'
b.	/mɪ-kɪ-itoki/	[mi-k-intoki]	'NEG-1PL-do.again'
c.	/kɪ-ɪdɪm-ʊ/	[k-ɪdɪm-ʊ]	'1PL-be.able-PRES'
d.	/kɪ-ɲorr-ʊ/	[ki-ɲorr-u]	'1PL-love-PRES'

Some suffixes also cause [+ATR] spreading. Examples of suffix-induced harmony are seen in (2). Observe that both the root meaning 'wash' and the intransitive suffix surface as [−ATR] in (2a), but in the presence of the dominant applied suffix, all root and prefix vowels surface as [+ATR] in (2b). In the same manner, the dominant instrumental suffix transforms the [−ATR] root and prefixes in (2c) to [+ATR] in (2d).

(2) Suffix-controlled [+ATR] spreading in Maasai

a.	/ɪsʊj-ɪʃɔ/	[ɪsʊj-ɪʃɔ]	'wash-INTRANS'
b.	/ɪsʊj-ɪʃɔ-re/	[isuj-iʃo-re]	'wash-INTRANS-APPLIED'
c.	/ɛ-tɛ-bɛl-ɑ/	[ɛ-tɛ-bɛl-ɑ]	'3SG-PF-break-PF'
d.	/ɛ-tɛ-bɛl-ie/	[e-te-bel-ie]	'3SG-PF-break-INST'

[6]Note also that we omit the transcription of tone on vowels, both because tone does not play a role in the harmony patterns of Maasai and Turkana as well as to visually clarify the contrast between [+ATR] /i/ and [−ATR] /ɪ/.

We focus now on the special behavior of the recessive low vowel /ɑ/. When a [+ATR] dominant vowel occurs to the right of /ɑ/, the low vowel blocks spreading and is output faithfully as [ɑ]. This is evident in (3). Observe that the first-person plural prefix /kɪ/ undergoes harmony under the influence of the following dominant vowel in the root /dot/ 'pull' in (3a), but when the past tense prefix /tɑ/ occurs between the two (3b), the low vowel /ɑ/ both fails to undergo harmony and prevents the more peripheral /kɪ/ prefix from undergoing harmony. Comparing (3c,d), we see the same behavior from an /ɑ/ in the root, which blocks leftward spreading originating from the dominant vowel of the applied suffix in (3d). Lastly, in (3e,f) we see that an /ɑ/ in a less peripheral suffix also blocks leftward spreading of [+ATR] from the applied suffix.

(3) Blocking behavior of /ɑ/ in Maasai

a.	/kɪ-dot-ʊn-ie/	[ki-dot-uɲ-ie]	'1PL-pull-MT-APPLIED'
b.	/kɪ-tɑ-dot-ʊn-ie/	[kɪ-tɑ-dot-uɲ-ie]	'1PL-PAST-pull-MT-APPLIED'
c.	/ɪ-duŋ-ɪʃɔ-re/	[i-duŋ-iʃo-re]	'2SG-cut-INTRANS-APPLIED'
d.	/ɪ-ɑs-ɪʃɔ-re/	[ɪ-ɑs-iʃo-re]	'2SG-do-INTRANS-APPLIED'
e.	/ɑ-ɪpʊt-ɑkɪn-ie/	[ɑ-ɪpʊt-ɑkiɲ-ie]	'3SG-fill-DATIVE-APPLIED'
f.	/ɛ-ɪpʊt-ɑ-rɪ-ie/	[ɛ-ɪpʊt-ɑ-ri-jie]	'3SG-fill-MA-N-APPLIED'

However, suffix /ɑ/ alternates depending on whether a dominant [+ATR] vowel occurs to its *left*. When following other recessive [−ATR] vowels, /ɑ/ is output faithfully, as in (4a,b). When following dominant [+ATR] vowels, though, /ɑ/ undergoes harmony and is output re-paired as [o], as in (4c,d). Similarly, when /ɑ/ is flanked on either side by dominant [+ATR] vowels, as in (4e,f) — compare (3e,f) above — it is output re-paired as [o] due to the presence of the dominant [+ATR] vowel to its left.[7]

(4) Re-pairing behavior of /ɑ/ in Maasai

a.	/ɪn-lɪpɔŋ-ɑ/	[ɪn-lɪpɔŋ-ɑ]	'FEM.PL-female-PROD'
b.	/ɔl-mɛn-ɑ/	[ɔl-mɛn-ɑ]	'MASC.SG-despise-PROD'
c.	/ɪn-mudoŋ-ɑ/	[in-mudoŋ-o]	'FEM.PL-relative-PROD'
d.	/ɛŋ-komon-ɑ/	[eŋ-komon-o]	'FEM.SG-pray-PROD'
e.	/ɑ-duŋ-ɑkɪn-ie/	[ɑ-duŋ-okiɲ-ie]	'3SG-cut-DATIVE-APPLIED'
f.	/ɛ-isud-ɑ-rɪ-ie/	[e-isud-o-ri-jie]	'3SG-hide-MA-N-APPLIED'

The harmony pattern in Maasai can be analyzed with two very similar rules spreading [+ATR] in opposite directions, as in Archangeli and Pulleyblank (1994): a left-to-right rule spreading [+ATR] to all following vowels (and re-pairing /ɑ/ to [o]), and a right-to-left [+ATR] spreading rule that is blocked by /ɑ/. These rules do not interact, and thus do not need not be ordered with respect to one another, because the left-to-right rule only affects vowels to the right of a dominant [+ATR] vowel and the right-to-left rule only affects vowels to the left of a dominant [+ATR] vowel. Consider a hypothetical underlying vowel string /ɛ ɑ ʊ i ɛ ɑ ɪ/, with the /i/ being the sole dominant [+ATR] vowel. Left-to-right spreading affects only the final /ɛ ɑ ɪ/ sequence, outputting it as [e o i], with re-pairing of /ɑ/ to [o]; it does not affect the realization of the initial sequence of vowels. Similarly, right-to-left spreading outputs the initial /ɛ ɑ ʊ/ as [ɛ ɑ u], with blocking by /ɑ/, but has nothing to say regarding the realization of the final sequence of vowels.

The situation is only superficially different when recessive vowels lie *between* dominant [+ATR] vowels. When a non-low recessive vowel is flanked by dominant vowels, it is output as [+ATR]. This is consistent with the effect of *either* of the two rules; in this context, we cannot determine which dominant vowel is responsible for the harmonic assimilation of the non-low vowel. When the low recessive vowel /ɑ/ is flanked

[7] Citing the example /ɛn-ʧoribɑ/ → [en-ʧoribɑ] 'rafters made of sticks', Quinn-Wriedt (2013, p. 158) concludes that low vowels in roots do not alternate regardless of their orientation with respect to dominant [+ATR] vowels. This is technically inconsistent with the description of the Maasai pattern that we are working with in the text, but can be straightforwardly accommodated in the analysis.

by two dominant vowels, it is output as [o], but in this context we can uniquely determine the dominant vowel responsible for the alternation: the one to the left, because /ɑ/ blocks spreading from the right. Since /ɑ/ blocks right-to-left spreading, in a flanking sequence like /e ɑ i/, the /i/ to the right will not cause assimilation (and re-pairing) of the /ɑ/. The /e/ to the left, though, can spread [+ATR] to /ɑ/, and so the string is ultimately output as [e o i] — regardless of the order of application between the two spreading rules.

In sum, the output quality of every vowel in Maasai depends on two sources of information. The first source of information is the vowel's underlying ATR specification, which determines whether it is a dominant (= spreading) or recessive (= alternating) vowel. In the latter case, the second source of information is the output ATR specification of *either* the preceding *or* the following vowel.[8] Recessive vowels to the left of a dominant vowel depend on the output ATR value of the following vowel while recessive vowels to the right of a dominant vowel depend on the output ATR value of the preceding vowel. Crucially, no individual vowel's output quality depends on the ATR specifications of vowels on *both* sides. In other words, the bidirectional ATR harmony pattern in Maasai is *not* unbounded circumambient, but rather unbounded *semi*ambient — recall §2.3, Defs. 2 and 3 above. This is of course what makes the Maasai harmony pattern fit comfortably within Heinz and Lai's (2013) class of WD functions: it can be broken down into an HL composition of two contradirectional subsequential functions, neither one of which marks up its output in such a way that the other function has crucial access to that markup. Or, in the stricter terms that we advocate in this article: the two functions do not interact and can even be composed in either order, because for any given input symbol, its output is determined by information on one side *or* the other, and never both simultaneously.

3.2 Turkana

As noted at the outset of this section, the bidirectional ATR harmony pattern in Turkana is in relevant respects identical to the pattern in Maasai except for the existence and consequences of a set of exceptionally dominant [−ATR] suffix vowels. We first establish the commonalities with the Maasai pattern. The contrast in (5a,b) shows that [+ATR] spreads leftward from dominant root vowels to preceding recessive prefix vowels, and the contrast in (5c,d) shows that [+ATR] spreads outward from dominant root vowels both to preceding recessive prefix vowels and to following recessive suffix vowels.

(5) Root-controlled [+ATR] spreading in Turkana

a.	/ɛ-kɔrɪ/	[ɛ-kɔrɪ]	'MASC.SG-ratel'
b.	/ɛ-kori/	[e-kori]	'MASC.SG-giraffe'
c.	/ɛ-ɛm-ɪ/	[ɛ-ɛm-ɪ]	'3SG-fear-ASP'
d.	/ɛ-los-ɪ/	[e-los-i]	'3SG-go-ASP'

Dominant suffix vowels also trigger [+ATR] spreading. The voice suffix /o/ and the gerundial suffix /e/ in (6b,d) cause all other, recessive vowels in the word to be output as [+ATR].

(6) Suffix-controlled [+ATR] spreading in Turkana

a.	/ɛ-ɑ-kɔkɔ-ʊn-ɪ/	[ɛ-ɑ-kɔkɔ-ʊn-ɪ]	'3-TNS-steal-VEN-ASP'
b.	/ɛ-kɔkɔ-ʊn-ɪ-o/	[ɛ-koko-un-i-o]	'3-steal-VEN-ASP-VOI'
c.	/ɑ-kɪ-dɔk/	[ɑ-kɪ-dɔk]	'INF-K-climb'
d.	/ɛ-dɔk-ʊn-e/	[e-dok-un-e]	'MASC.SG-climb-VEN-GER'

The final parallel between Maasai and Turkana is the behavior of the low vowel /ɑ/ in [+ATR] contexts. Examples (7a,b) show that the low vowel of the infinitive prefix is output faithfully regardless of the ATR value of the root vowel. In examples (7c,d), we see that the low vowel of the root 'beat' is also output faithfully regardless of the presence of the dominant gerundial suffix vowel /e/ in (7d), blocking spreading to the

[8]The output specifications of the adjacent vowels is what matters, not their underlying specifications, due to the iterativity of spreading. In other words, each of the two harmony rules is, strictly speaking, OUTPUT STRICTLY LOCAL (Chandlee et al. 2015).

preceding recessive vowel /ɛ/ of the masculine singular prefix. Examples (7e,f) further show that the low vowel of the habitual suffix blocks spreading from the dominant nominalizing suffix vowel /u/.

(7) Blocking behavior of /ɑ/ in Turkana

a.	/ɑ-dɔk-ʊn/	[ɑ-dɔk-ʊn]	‘INF-climb-VEN’
b.	/ɑ-lim-ʊn/	[ɑ-lim-un]	‘INF-tell-VEN’
c.	/ɑ-kɪ-ram/	[ɑ-kɪ-ram]	‘INF-K-beat’
d.	/ɛ-ram-e/	[ɛ-ram-e]	‘MASC.SG-beat-GER’
e.	/ɑ-pɛg-ɑan-u/	[ɑ-pɛg-ɑan-u]	‘GER-deny-HAB-NOM’
f.	/ɑ-dɑk-ɑan-u/	[ɑ-dɑk-ɑan-u]	‘GER-graze-HAB-NOM’

However, when a suffix low vowel follows a dominant [+ATR] root vowel, as shown by the contrasts in (8a–d), or when it is flanked on both sides by dominant [+ATR] vowels (8e,f), harmony (and re-pairing) obtains, and /ɑ/ is output as [o].[9] Given that leftward harmony is blocked by low vowels, we must assume that rightward spreading from the root is what causes /ɑ/ to be output as [o] in (8e,f); compare (7e,f) above.

(8) Re-pairing behavior of /ɑ/ in Turkana

a.	/ɛ-tɪm-ɑt/	[ɛ-tɪm-ɑt]	‘FEM.SG-shoe-SG’
b.	/ɛ-pur-ɑt/	[e-pur-ot]	‘MASC.SG-beer-SG’
c.	/ɛ-pɛg-ɑan-n-ɑ/	[ɛ-pɛg-ɑan-n-ɑ]	‘3SG-argue-HAB-SG-VOI’
d.	/ɛ-pup-ɑan-n-ɑ/	[e-pup-oon-n-o]	‘3SG-obey-HAB-SG-VOI’
e.	/ɑ-tur-ɑan-u/	[ɑ-tur-oon-u]	‘GER-agile-HAB-NOM’
f.	/ɑ-pup-ɑan-u/	[ɑ-pup-oon-u]	‘GER-obey-HAB-NOM’

Up to this point, the harmony patterns in Maasai and Turkana appear to be identical. However, there is one crucial difference between the two: there are exceptionally dominant [−ATR] suffix vowels in Turkana in addition to dominant [+ATR] suffix vowels. We assume that these exceptional suffixes (or the vowels themselves) are lexically specified as dominant and henceforth circle them in cited examples to distinguish them from recessive [−ATR] suffix vowels. When an exceptionally dominant [−ATR] suffix vowel co-occurs with a dominant [+ATR] root vowel, the exceptionally dominant vowel both resists becoming [+ATR] and imposes its own [−ATR] value on all preceding vowels, in a pattern that Baković (2000) refers to as ‘dominance reversal’.

Observe in (9a) that the (non-exceptional) dominant vowels of the root /ido/ ‘give birth’ spread [+ATR] rightward to the recessive vowels of the ventive and aspect-marking suffixes /ʊn/ and /ɪt/, causing these suffixes to be output as [un] and [it], respectively. In the presence of the exceptionally dominant [−ATR] instrumental-locative suffix /ɛt/ (9b), the recessive vowel of the ventive suffix as well as the dominant [+ATR] vowels of the root are all output as [−ATR]. The remaining examples in (9) illustrate what happens when the low vowel /ɑ/ of the epipatetic suffix is in a fully [−ATR] context (9c), surfacing faithfully as [ɑ]; when it is preceded by dominant [+ATR] vowels (9d), surfacing re-paired as [o]; and when it is flanked by dominant [+ATR] vowels to its left and an exceptionally dominant [−ATR] suffix to its right (9e), surfacing as [ɔ] — the expected harmonic counterpart of [o] — rather than faithfully as [ɑ], which would also be in harmonic agreement with the exceptionally dominant [−ATR] vowel in this case.

(9) Dominance reversal (exceptional suffix-controlled [−ATR] spreading) in Turkana

a.	/ido-ʊn-ɪt/	[ido-un-it]	‘give.birth-VEN-ASP’
b.	/ɑ-k-ido-ʊn-ɛt/	[ɑ-k-ɪdɔ-ʊn-ɛt]	‘GER-K-give.birth-VEN-INST.LOC’

[9]In his discussion of the Turkana pattern, van der Hulst (2018, p. 355) states that “[w]ithin morphemes, [the] distribution [of the low vowel] is limited in such a way that it never occurs following an advanced vowel”, and cites the form /ɛ-kɑlees/ → [ɛ-kɑlees] ‘SG-ostrich’ showing that a low vowel may occur *preceding* a [+ATR] vowel within a morpheme. This is consistent with our description of the Turkana pattern that we are working with in the text; cf. the inconsistency noted in fn. 7 for Maasai.

c. /ɑ-k-ɪmɔk-ɑ-kɪn-ɪ/ [ɑ-k-ɪmɔk-ɑ-kɪn-ɪ] 'GER-K-light.fire-EPI-DAT-V'
d. /ɛ-ibus-ɑ-kɪn-ɪ/ [e-ibus-o-kin-i] '3-drop-EPI-DAT-V'
e. /ɛ-ibus-ɑ-kɪn-ⓐ/ [ɛ-ɪbʊs-ɔ-kɪn-ⓐ] 'GER-drop-EPI-DAT-VOI'

There are some further details to be noted about the behavior of these exceptionally dominant [−ATR] suffix vowels, as well as some differences in the relevant data cited by Dimmendaal (1983) and Noske (2000). First, Dimmendaal (1983, p. 25–26) notes that if a suffix with an exceptionally dominant [−ATR] mid vowel (/ɔ/ or /ɛ/) is affixed directly to a root with a dominant [+ATR] mid vowel (/o/ or /e/), a harmony-blocking glide [j] is inserted between them: /ɑ-k-item-ɛt/ → [ɑ-k-item-jɛt] 'attempt'; cf. (9b), where a (recessive) suffix intervenes between the same exceptionally dominant [−ATR] suffix and another root with a dominant [+ATR] mid vowel.[10] Second, Noske (2000, p. 781–782) claims that only mid vowels, and not high vowels, undergo (leftward) [−ATR] spreading. Noske thus transcribes the form in (9b) as [ɑ-k-ido-un-ɛt], explicitly noting (fn. 8, p. 781) that this differs from Dimmendaal (1983); the critical form in (9e) — not cited by Noske — would presumably be [e-ibus-o-kin-ⓐ].[11] We ascribe this difference between Dimmendaal's (1983) transcriptions and Noske's to a dialect difference, and stick with Dimmendaal's. Lastly, Noske (2000, p. 782, (38)) cites examples with an exceptionally dominant [−ATR] suffix vowel followed by a dominant [+ATR] suffix vowel, e.g. [ɛ-k-ɪlɔt-ɑr-e] 'way of washing'. Noske states that this is "[t]he only construction type in which the two types of [dominant] suffixes occur regularly", but it is at least circumstantial evidence that [−ATR] does not spread *rightward* from exceptionally dominant [−ATR] suffix vowels. The fact that [+ATR] also does not spread leftward in this example could be due either to the invariance of exceptionally dominant [−ATR] vowels or to the fact that this particular exceptionally dominant [−ATR] vowel is /ɑ/ and thus independently expected to block [+ATR] spreading; for concreteness, we assume that both reasons apply.

Two further things are also worth noting here. First, there is no readily apparent phonological information to independently determine whether a given [−ATR] vowel is recessive or exceptionally dominant; compare, for example, the recessive voice-marking suffix /ɑ/ in (7g,h) with the distinct, exceptionally dominant voice-marking suffix /ⓐ/ in (9e). Second, note in (9e) that the exceptionally dominant [−ATR] vowel may be separated from the affected low vowel by another syllable; we assume that this distance is in principle unbounded, just as we assume — following the extensive literature on the analysis of vowel harmony — that the distance between any trigger and potential target of harmony is in principle unbounded.

Analytically speaking, Maasai and Turkana both exhibit bidirectional [+ATR] spreading, and in addition Turkana exhibits [−ATR] spreading. This additional part of the pattern in Turkana has significant computational repercussions for the analysis. Recall that the output of every alternating vowel in Maasai is dependent on two sources of information: that vowel's input specification, and the output specification of a neighboring vowel on *either* side. In Turkana, the output of every alternating vowel depends on additional information: whether there is a preceding dominant [+ATR] vowel *and* whether there is a following exceptionally dominant [−ATR] vowel. This is the key distinction between the Maasai and Turkana patterns. The output of a recessive low vowel in the input in Turkana is not able to be determined based solely on information about vowels to its left *or* about vowels to its right; it depends on information from *both* sides, *simultaneously*.

Consider a hypothetical underlying vowel string /i ɛ ɑ ʊ ⓔ/, where /i/ is a dominant [+ATR] vowel and /ⓔ/ is an exceptionally dominant [−ATR] vowel. The output of the vowels in the medial string of recessive vowels /ɛ ɑ ʊ/ depends on both of these flanking sources of information. If only the dominant [+ATR] vowel were present to the left, and the exceptionally dominant [−ATR] vowel were *not* also present to the right, then the vowels in this medial string would all be output as [+ATR], [e o u], with the /ɑ/ re-paired to [o]. But because the exceptionally dominant [−ATR] vowel *is* also present to the right, all vowels to its left

[10] Glides in general disrupt harmony, such that following vowels "may be [+ATR] or [−ATR] (as with vowels following the vowel /ɑ/)" and preceding vowels "are [+ATR]" (Dimmendaal 1983, p. 20). This accounts for a notable exception to van der Hulst's (2018) generalization in fn. 9 that low vowels never follow [+ATR] vowels within morphemes: the form [ŋi-turkana] 'the Turkana people' has a "more frequently used form" [ŋi-turkwana] with a harmony-disrupting glide (Dimmendaal 1983, p. 20).

[11] Noske (2000, p. 778) also cites the form in (9a), but transcribes the root as [edo] in this case rather than as [ido].

are instead output as [−ATR], [ɪ ɛ ɔ ʊ]. The fact that both sources of information are critical is made most obvious by the fact that the /ɑ/ is output as [ɔ]: the fact that it is [−low] is due to the dominant [+ATR] vowel to the left, and the fact that it is [−ATR] is due to the exceptionally dominant [−ATR] vowel to the right.

Furthermore, both sources of information could in principle be at an unbounded distance from any of the vowels between them. This dependency is thus consistent with the unbounded circumambient class of patterns; recall §2.3, Def. 2. The Turkana pattern should thus *not* be able to be described with an HL composition and to be classified as weakly deterministic in the way that the Maasai pattern is, and yet an EM composition analysis of the Turkana pattern exists in which the first, inner function of the composition is neither alphabet- nor length-increasing, consistent with Heinz and Lai's (2013) definition.

We briefly sketch this analysis here before presenting it in more technical detail in §4.3. The inner, left-subsequential function spreads [+ATR] from underlying dominant [+ATR] vowels from left to right, blocked only by following exceptionally dominant [−ATR] vowels. Underlying /ɑ/ is also re-paired by this function to intermediate |o|, which of course is representationally identical to an intermediate |o| from underlying /o/ or /ɔ/. The outer, right-subsequential function then acts on this intermediate representation, spreading [+ATR] from dominant [+ATR] vowels and now also [−ATR] from exceptionally dominant [−ATR] vowels, this time from right to left. Spreading of [−ATR] is unimpeded, and spreading of [+ATR] is blocked both by low vowels and by exceptionally dominant [−ATR] vowels.[12] All instances of intermediate |o|, whether from underlying /ɑ/, /o/, or /ɔ/, will surface as [ɔ] if and when this function spreads [−ATR] to them.

The inner function thus increases neither the size of the alphabet nor the length of the string, but the two functions do crucially interact. This is most obvious in the case of flanked /ɑ/: it must first be re-paired to |o| by the inner, left-subsequential function, which crucially enables it to be output as [ɔ] by the outer, right-subsequential function. This is why we maintain that Heinz and Lai's (2013) reliance on the absence of markup in their definition of weak determinism is insufficient, and that instead the absence of function interaction is key. When the output of a given element in a string depends on information *either* unboundedly far to the left *or* unboundedly far to the right, as it does in the Maasai pattern, no interaction between composed contradirectional subsequential functions is necessary; when it depends on information *both* unboundedly far to the left *and* unboundedly far to the right, as it does in the Turkana pattern, there *is* interaction between composed contradirectional subsequential functions.

4 Revising the definition of weak determinism

In the previous section we demonstrated that the nature of the dependencies in the harmony patterns of Maasai and Turkana are distinct: that the Turkana pattern is more computationally expressive due to its unbounded circumambience, and that the Maasai pattern is less computationally expressive due to its unbounded semiambience. Despite this difference, Heinz and Lai's (2013) formal definition of weak determinism treats the two patterns as equivalent since they can both be broken down into two contradirectional subsequential functions that involve neither alphabet- nor length-increasing markup.

In this section, we formalize a definition of weak determinism (§4.4, Def. 6) that correctly distinguishes between unbounded semiambient and unbounded circumambient processes and which centers the concept of interaction, defined less formally in §2.1, Def. 1 above and to be defined more formally in §4.2, Def. 5 below. We present a formulation of the WD functions 'from above' defined directly in terms of a restriction on ND functions (recall §2.3), in contrast with Heinz and Lai (2013), who attempt to define the WD functions 'from below' in terms of restricted compositions of contradirectional subsequential functions following Elgot and Mezei's (1965) result relating the ND functions to compositions of subsequential functions (recall §2.2). In doing so, as we shall see, the presence of interaction in ND function maps and their absence in WD function maps becomes more clear.

[12]But recall from the discussion below (9) that there are no examples (known to us, at least) that crucially require [+ATR] spreading to be blocked by non-low exceptionally dominant [−ATR] vowels.

In general, an ND function may use information from an unbounded distance in *both* directions to determine the output for any given input symbol of a given input string. Our proposed restriction for the definition of WD functions requires that for any given output position, the function only ever needs unbounded access in *one* direction, but that direction may vary from one point in any given string to another. For example, two iterative phonological processes proceeding in opposite directions where neither process changes the conditions or result of (nonvacuous) application of the other (as in the analysis of Maasai) will never result in an input string $w = x_1 x_2 \dots x_i \dots x_n$ where there is some x_i whose associated output depends on information from both directions: either one process will apply, the other will apply, neither process will apply, or if both apply, those applications must result in the same outcome.[13] We formalize this concept using an automaton called a *bimachine* (Schützenberger 1961, Eilenberg 1974), use it to introduce a more precise definition of interaction, show how interaction cleanly distinguishes the Maasai and Turkana patterns, and then discuss how this relates to previous bottom-up formulations of weak determinism centered on 'markup' and interaction of contradirectional subsequential functions.

4.1 Non-deterministic regular functions and bimachines

A BIMACHINE is the canonical machine representation for regular functions (Schützenberger 1961, Eilenberg 1974, Reutenauer and Schützenberger 1991). Briefly, a bimachine is defined by two deterministic finite-state automata, $\mathcal{L}$ and $\mathcal{R}$, that read the input from opposite directions and a simple output function ω that, given a tuple containing an input symbol x_i and the state of each automaton before it reads x_i ($q_j^{\mathcal{L}}$ for $\mathcal{L}$ and $q_k^{\mathcal{R}}$ for $\mathcal{R}$), maps that tuple to an output string w_i. Equivalently, bimachines can be viewed as explicitly determinized versions of functional (one-way) non-deterministic finite-state transducers (NDFSTs)[14] where determinization is made possible by the summary of information from both directions provided by the contradirectional automata. That is, for a given bimachine, every "joint state" $(q_j^{\mathcal{L}}, q_k^{\mathcal{R}})$ of the bimachine — at any given point for any given input word — corresponds to some state of the bimachine's corresponding NDFST (Mihov and Schulz 2019, Ch. 6, §2, Lhote 2018, §3.1).[15] In this sense, bimachines provide expressivity equal to that of the NDFSTs used in much of the computational phonology literature. Crucially, for the purposes of this article, bimachines provide clarity and ease of use that are notoriously not offered by NDFSTs, and this is why we employ bimachines here.

As we have argued, assessing the necessity of unbounded circumambient information is critical to determine membership in the WD vs. the ND classes of functions. Therefore, because bimachines make explicit the directionality of all information necessary to determine the output associated with each input symbol, we adopt them here as the clearest representation for defining weak determinism and differentiating it from non-determinism. Furthermore, bimachines offer not only a correct definition but also a clearer definition of the intended scope of the class of WD functions and how it relates to previous work on the boundary between ND and WD (Heinz and Lai 2013, Jardine 2016). Finally, bimachines are a theoretical tool that offer a deterministic representation of regular functions that obviates consideration of new and inventive forms of markup when considering the complexity of a string function.

We now provide a more explicit characterization of bimachines and the notation we will adopt here.

Definition 4. A bimachine $(\mathcal{L}, \mathcal{R}, \omega)$ that calculates a regular function $f : X^* \to Y^*$ is defined by a left-to-right automaton $\mathcal{L}$, a right-to-left automaton $\mathcal{R}$, and an output function $\omega : Q_{\mathcal{L}} \times X \times Q_{\mathcal{R}} \to Y^*$ that maps triples of a state from $\mathcal{L}$ ($Q_{\mathcal{L}}$), a symbol of the input X, and a state from $\mathcal{R}$ ($Q_{\mathcal{R}}$) to a substring of the output Y^*.

[13] These last cases meet the definition because it can be determined that the process applies without considering information *only* obtainable from both directions.

[14] A one-way NDFST is FUNCTIONAL if it maps each complete input string to at most one output string; such a transducer may nevertheless exhibit incremental non-determinism in which state(s) it transitions to as it reads a string from one edge to the other and what the output might be, but by defintion of being functional, by the end of the input string at most a single unique output string must be picked out. Except where noted, every reference to an NDFST henceforth will be to a functional one-way NDFST.

[15] In general, multiple joint-states of a bimachine may correspond to a given state of an equivalent NDFST.

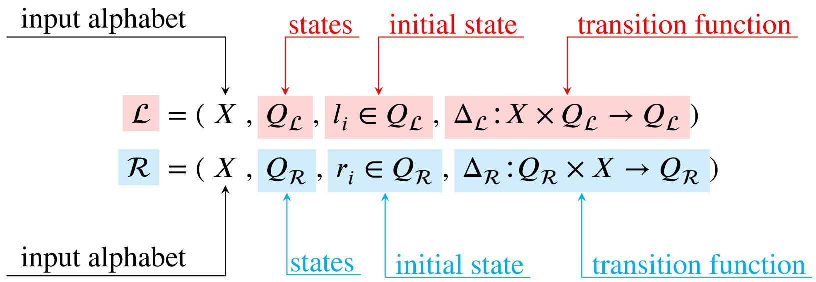

Figure 3 Definitions of left ($\mathcal{L}$) and right ($\mathcal{R}$) automata. Each automaton is defined over an input alphabet (X), a set of states ($Q_{\mathcal{L}/\mathcal{R}}$), a designated initial state ($q_i^{\mathcal{L}}/q_i^{\mathcal{R}}$), and a state transition function ($\Delta_{\mathcal{L}/\mathcal{R}}$). Note that we follow a common stylistic convention for bimachines where the right automaton's transition function has its argument order flipped ($\Delta_{\mathcal{R}} : Q_{\mathcal{R}} \times X \to Q_{\mathcal{R}}$) relative to the left automaton's transition function ($\Delta_{\mathcal{L}} : X \times Q_{\mathcal{L}} \to Q_{\mathcal{L}}$).

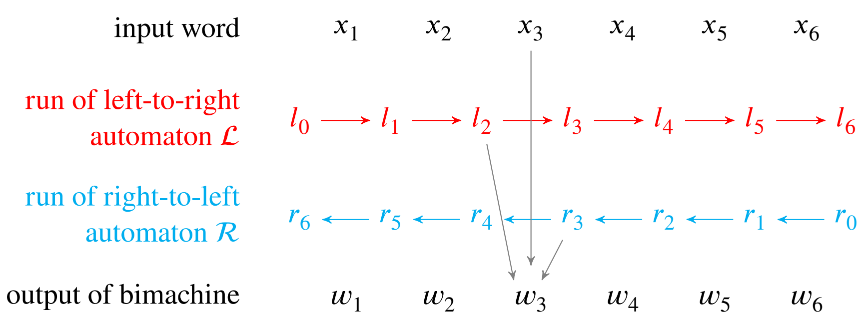

Figure 4 An example run of a bimachine over an arbitrary string. The i-th output string (w_i) is the value of the output function on (1) the state of left-to-right automaton after reading input symbols $x_{<i}$, (2) the i-th input symbol x_i, and (3) the state of right-to-left automaton after reading input symbols $x_{>i}$. Adapted with permission from Bojańczyk and Czerwiński (2018).

An inventory of the components that define $\mathcal{L}$, $\mathcal{R}$, and their transition functions is given in Fig. 3.[16]

The computation of an input-output mapping by a bimachine is illustrated schematically in Fig. 4, and proceeds in the following way. For an input word $w = x_1 x_2 \ldots x_i \ldots x_{n-1} x_n$, first the series of state transitions of each of the two automata is calculated in the usual fashion. Then, the state sequences from each of the two automata are aligned into pairs. Finally, using the definition of the output function ω, the appropriate output string is emitted for each pair of automata states and their associated input symbol. That is, for an associated input symbol x_i, if before reading x_i from the left, the left automaton $\mathcal{L}$ is in state $q_j^{\mathcal{L}}$, and before reading x_i from the right, the right automaton $\mathcal{R}$ is in state $q_k^{\mathcal{R}}$, then the output of x_i is determined by $\omega\left(q_j^{\mathcal{L}}, x_i, q_k^{\mathcal{R}}\right)$.

We close this introduction to bimachines and their interpretation with three observations: one on the correspondence between bimachines and more familiar NDFSTs, one on the interpretation of the output function, and one on the relevance of bimachines for formal theories of phonology that do not obviously make use of superficially similar representations.

First, as briefly mentioned earlier, the relationship between bimachines and NDFSTs is straightforward. Each pair of automata states of the bimachine can be interpreted as a single state of an NDFST, and the

[16]Note that final states are not included here among the components that define a bimachine's automata. This is because (without loss of generality) we assume Schützenberger (1961)-style bimachines that implement total functions, meaning that they are defined for every input string. It is conventional in this setting to either omit discussion of final states or to (equivalently) declare that every state is a final state.

bimachine's output function performs a function analogous to the NDFST's transition relation/output function (see e.g. Mihov and Schulz 2019, Ch. 6, §2 and Lhote 2018, §3.1 for details). This correspondence between bimachines and NDFSTs highlights that every state of a functional NDFST can be factored into information about PREFIXES of the input (= substrings originating at the left edge of the input) plus information about SUFFIXES of the input (= substrings originating at the right edge of the input), relative to the position associated with the current state after an NDFST reads some partial input string.[17] This intuition will be useful when considering the relationship of bimachines to EM compositions and hence also when considering the relationship of the definition of the WD functions presented here to the approaches of previous definitions.

Second, note that the output function of a bimachine $\omega : Q_{\mathcal{L}} \times X \times Q_{\mathcal{R}} \rightarrow Y^*$ implementing a regular function provides a 'local' summary of that function's behavior — it describes everything we need to know about what output each input symbol gets mapped to under what CONTEXT, described by a pair of states $(q_j^{\mathcal{L}}, q_k^{\mathcal{R}})$. Because each left-automaton state corresponds to a set of prefixes and each right-automaton state corresponds to a set of suffixes, each of the state pairs that can actually occur together in a given bimachine and define a context for ω corresponds to a set of *circumfixes* (= contexts). In more linguistically familiar terms, the output function ω of a bimachine is essentially a carefully orchestrated *set* of rewrite rules defining the local, contextualized behavior of the regular function associated with the bimachine. Each state of a reachable state pair corresponds directly to the regular expressions of symbols (usually feature vectors) that define a lefthand (or righthand) context in a traditional rewrite rule like $A \longrightarrow B \;/\; C \_\_ D$.

We close by summarizing the significance of bimachines for phonological formalisms that do not appear to use anything resembling a finite-state transducer (or bimachine) to model string functions. First, recall from §2.1 that any phonological formalism that involves a function mapping an input string to an output string (usually a UR→SR map) has some presentation-independent complexity indicating the minimum expressivity required to implement that function; recall also from §1 that complexity is organized in terms of *classes* of functions — e.g. input strictly local, subsequential, etc. Each of these classes usually has one or more MACHINE-INDEPENDENT or EXTENSIONAL CHARACTERIZATIONS and usually has one or more CANONICAL machines that characterize any function in that class. Briefly, a *machine-independent* or *extensional characterization* of a function $f : X^* \rightarrow Y^*$ in some class C depends only on properties of the mapping f between input string and output string — not on any specific representation (e.g. phonological formalism) or implementation of that input-output mapping. A *canonical* machine of some type for f is usually unique, has a minimal number of states relative to all machines in class C that implement f, and has some relatively transparent connection between the definition of its components and at least one machine-independent characterization of f. A machine implementing some function f is a concrete representation for working with or analyzing f, and a machine-independent characterization may be unwieldy: a canonical machine for f offers some of the best of both worlds — concreteness, uniqueness, and interpretability. In the present context, every regular function has an associated canonical bimachine, generally comparable to the status that e.g. a (minimal) deterministic one-way finite-state automaton has for a regular language and that an onward subsequential transducer has for a subsequential function (Oncina et al. 1993, §3; Choffrut 2003; Heinz and Lai 2013, §2.1).[18]

4.2 Interaction

In this subsection we discuss a general definition of *interaction* between regular functions inspired by the linguistic usage of the term, and describe the role of interaction in ordered compositions of contradirectional subsequential functions (EM compositions) that can describe unbounded circumambient patterns.

[17] For output-sensitive functions, this would be a combination of information about prefixes of the output plus information about suffixes of the output, relative to the current state's position.

[18] See Heinz (2018, §2) for explicit discussion of the relationship between phonological formalism, formal language theory, and intensional vs. extensional characterizations of a language class or function class; compare this with e.g. McCollum et al. (2020a, §6) for a differing view of the role and significance of formal language theory in phonological theory with respect to explanation (as opposed to theory *qua* formalism) and typology.

Figure 5 Three of many possible ways the input word *ab* could be mapped by some function f to the output word *ab*.

In an interacting composition the first function f to apply to a word w causes a *change in the behavior* of the second function g to apply relative to what would have occurred if g had simply been applied to w instead of to $f(w)$; the net result is an ND function $g \circ f$ whose behavior is 'more than the sum of its parts', i.e. not simply the union of what one would expect from applying g to w and applying f to w.

4.2.1 Interaction in bimachines

As noted at the end of §4.1, the output function $\omega : Q_{\mathcal{L}} \times X \times Q_{\mathcal{R}} \to Y^*$ of a bimachine summarizes the local input-output behavior of the bimachine in any given context, where 'context' is completely summarized by the pair of states of the bimachine's automata reading the string context on each side of a given position within the input string. Recall also that a canonical machine for a given function f offers a concrete, unique, and interpretable representation of f. For these reasons, bimachines provide us with precisely the right machinery to operationalize the concept of interaction. Consequently, as we argue, bimachines enable us to define weak determinism in a way that correctly reflects the difference between unbounded circumambience and unbounded semiambience and that captures the linguistic intuition that an interacting composition involves a 'change in behavior' of $g \circ f$ relative to the behavior of g and the behavior of f individually. Furthermore, this approach has computational merit, facilitating future exploration of the properties held by the output functions of canonical bimachines, due to its machine-independent foundation, which we describe now.

Bojánczyk (2018, Thm. 2.1, fn. 3) shows that if we have a specific kind of what a phonologist would call a correspondence relation for a regular function f, then we can define a machine-independent characterization — and therefore a canonical bimachine — for that function. Briefly, a function f augmented with ORIGIN SEMANTICS is a normal string function equipped with a specific kind of correspondence relation: an ORIGIN FUNCTION $\mathbb{o}_{w,f(w)} : \mathbb{N} \to \mathbb{N}$ which maps, for every (input word, output word) pair, each position j of the output word $f(w)$ to the position i of the symbol in the input word that caused position j to have the symbol it does (see e.g. Dolatian and Heinz 2020 for discussion in a linguistic setting).

For example, consider the behavior of an unknown regular function f on an input word $w = ab$ such that $f(w) = ab$. Each of the correspondence relations ('origin graphs') in Fig. 5 represents a possible origin semantics for how $f(ab) = ab$ could be the case for some function f equipped with origin semantics. On the left in Fig. 5, f associates each input symbol with an output symbol in a one-to-one fashion. In the middle, f associates the first input symbol a with ab and the other input symbol b with no output symbol (the 'empty string' ϵ). On the right, f associates the input a with no output symbol but emits ab for the input b.

For every regular function, then, if we have an origin semantics for it, we can define a canonical bimachine with a canonical output function $\omega : Q_{\mathcal{L}} \times X \times Q_{\mathcal{R}} \to Y^*$ driven by an associated set of canonical state pairs (Bojánczyk 2018, Thm. 2.1 and fn. 3). This means that for every regular function equipped with origin semantics, we have a straightforward path to a machine-independent characterization of that function, which we can manipulate through a canonical bimachine. This is accomplished through the observation that every regular function f augmented with origin semantics necessarily has a unique finite set of equivalence classes of two-sided contexts, and each of these equivalence classes has a function that maps an input symbol x to an output (Bojánczyk 2018, fn. 3; see also the presentation of origin semantics in Filiot 2015, §5.4). For example, one of the equivalence classes of a palatalization pattern would be a context with a high vowel to the right of the input symbol, and the function associated with this equivalence class would map undergoers of palatalization to their palatalized form and would map non-undergoers faithfully. The total set of equivalence classes of contexts corresponds to the reachable state pairs of a canonical bimachine for f, which can be used to define the output function $\omega_f : Q_{\mathcal{L}} \times X \times Q_{\mathcal{R}} \to Y^*$ for the canonical bimachine of f. In short, the behavior of an ND function f is summarized by the output function ω and the set of reachable state pairs of f's canonical bimachine (per §4.1).

Accordingly, we can define a NON-INTERACTING composition $g \circ f$ as one where the composition's behavior is exactly the sum (union) of its composands, and an INTERACTING composition as one where something is added or removed relative to the behavior of its composands.

Definition 5. Given length-preserving functions $g, f, g \circ f : X^* \to X^*$ with origin-semantics and canonical bimachines whose output functions are given by ω_g, ω_f, and $\omega_{g \circ f}$, the composition $g \circ f$ is a *non-interacting* composition iff $\omega_{g \circ f}$ reflects all and only the changes that ω_g makes (in the contexts ω_g makes them) plus the changes that ω_f makes (in the contexts ω_f makes them). If instead for $g \circ f$ there exists at least one context (p, s) and input symbol x where the condition above fails to hold, the composition $g \circ f$ is an *interacting* composition at x in the context (p, s).

While the number of unique contexts (p, s) that exists is in principle infinite, bimachines partition the set of contexts into a finite number of equivalence classes, as described above. In this way, the finiteness of bimachines is critical to the operationalization of this definition of (non-)interaction. Once contexts are mapped to the reachable state pairs of some canonical bimachine for the composition, then the reachable state pairs of the composition can be related to the reachable state pairs of each of its composands by applying the definition of bimachine composition (see Peikov 2007, §4 or Mihov and Schulz 2019, Ch. 6, §4). The details of this construction lie beyond the scope of this article; importantly, it is precisely the use of bimachines that allows this definition of interaction to be effectively operationalized, casting a solid foundation for the definition of weak determinism that we propose.

4.2.2 Interaction in EM compositions

In the specific setting of EM compositions we are interested in here, as we have noted throughout this article so far, the interaction of contradirectional compositions of subsequential functions — the change in behavior of g as a result of information encoded by f about the causal future of g's input in an intermediate string — is exactly what enables the composition to express functions in a class (ND) that is more expressive than that of either of its composands. We now elaborate on the relationship of the notion of interaction between regular string functions expressed in Def. 5 to the notion of interaction between the composands of an EM composition, paying specific attention to the features of EM compositions that allow them to express ND functions — even in a setting where new symbols and length-increasing codes for new symbols are forbidden, i.e. a setting that meets Heinz and Lai's (2013) unintentionally flawed definition.

In more detail, consider a schematic example composition $l \circ r(w)$ of two length-preserving and alphabet-preserving functions applied to an arbitrary input string w: a composition of a left-subsequential function $l : X^* \to X^*$ and a right-subsequential function $r : X^* \to X^*$. For the sake of illustration, we assume that r does nothing except behave as lookahead for l when it is applied to w. If we break w down into an arbitrary 'prefix' p and 'suffix' s, then the essense of what r must do is classify the suffix s from the right and then transform p in such a way that when l reads the output of $r(w) = p's$, on the basis of p', it can transform $r(w)$ in a way that reflects knowledge about s that l could not have inferred from reading w itself or could not have inferred early enough (far enough to left) in p to change its output behavior for s. This change in behavior of l as a result of changes made by r is precisely why an EM composition expressing an ND function necessarily exhibits interaction of the kind identified by Def. 5.

So how does our definition of interaction allow us to identify EM compositions that are actually WD and not ND? We begin by discussing how certain distributions of information in strings of the input language L can enable our example lookahead function r to interact with the contradirectional function l. Recall from earlier in this subsection that regular functions with origin semantics classify the circumfixes of the input language into a finite number of equivalence classes: left-subsequential functions classify their input language into a finite number of prefix equivalence classes, and right-subsequential functions classify their input language into a finite number of suffix equivalence classes.[19] For L to have some distribution of information in its strings that will enable interaction, then, there first must be some finite partition π_0 of strings of L that cannot be expressed by a left-subsequential function (i.e. as a finite partition of prefixes of L). That is, this partition must either be something that no left-subsequential function could ever capture about L at all, or more specifically a partition of L that no left-subsequential function could both capture and alter its behavior soon enough to act on that information. Thus, to work around the illegibility of π_0 to a left subsequential function, r's effect on L must be to make possible a partition of prefixes π_1 that is contingent on the content of the suffixes of L. In this way, the partition of prefixes π_1 must allow a subsequential function l to infer which partition of π_0 every whole string belongs to using just information in the prefix that has been modified by r.

The first step of what r must accomplish — classifying s from the right — will be reflected in the states of a canonical subsequential transducer for r (Heinz and Lai 2013, §2.1): to encode the simplest possible kind of lookahead — some binary classification of the suffix — there needs to be a two-way partition of the states of r's transducer reflecting suffixes that fall on one side of this classification vs. the other.

The next requirement is that r must 'remember' this classification of s for a distance with no principled upper bound. In a subsequential transducer for r, this would manifest as loops within at least one of the two sets of states previously mentioned as reflecting the classification of s.

While it traverses p from the right and remembers the classification of s, r must then classify the rightmost portion of the prefix p to decide when it is time to alter the prefix in some way that can be used by l. Operationally, this means that a subsequential transducer for r would need to distinguish between different possible leftward continuations of the suffix s — those where it is not yet time to make a change vs. those where the next step will be making a change. Once a subsequential transducer for r has identified that it has finished reading the rightmost region of the prefix p and that it is now time to make an edit, the edit r makes to p must be dependent on the classification of s that r arrived at earlier. For example, as expanded on in §5.2, a markup strategy that does not require new symbols or length-increasing codes is what we term *phonotactic codes*. Such a strategy uses intermediate strings that are not in general part of the input language (i.e. they are phonotactically illicit in all contexts) or which at least do not appear in input contexts for l outside of those created by r. In this way, r can deposit information that a left-subsequential function will deduce could only have been created by a preceding right-subsequential lookahead function.

Regardless of the specific markup strategy (if any) employed by r, the left-subsequential transducer l must now read $r(w)$ from the left and be able to classify some prefix of $r(w)$ and as a result act as though it knew which partition r assigned to s before l could have inferred that itself by reading w from the left. Such a change in the behavior of l as a result of changes made by r is exactly where an EM composition (of canonical bimachines) involves interaction as described in Def. 5. As we will show in subsequent sections, such interaction is far easier to diagnose when functions are viewed as bimachines.

4.3 The computational distinction between the Maasai and Turkana harmony patterns

We now show how bimachines bear out the difference between the unbounded semiambience of the Maasai pattern and unbounded circumambience of the Turkana pattern, and in doing so, we demonstrate how bimachines accentuate the necessity of interaction that is core to the Turkana pattern.

The lefthand, red column of Fig. 6 defines the Maasai harmony pattern as an EM composition for which the order of composition does not matter. No matter which order of composition is chosen, in neither case does the first function to apply cause a change in behavior of the other in the sense introduced in §4.2, and hence there is no interaction in the EM composition. Translation of the EM composition to an equivalent

[19] See e.g. the summaries in Heinz and Lai (2013, §2.1) or Gauwin (2020, Ch. 5 §§1.1-1.3) of the machine-independent characterization of subsequential functions and related expositional context.

bimachine, shown in the righthand column, is straightforward in this case. We can simply remove information about outputs from the subsequential FSTs in the lefthand column to yield the automata $\mathcal{L}$ and $\mathcal{R}$ given in the righthand, blue column. Additionally, we can define the bimachine's output function ω in the natural way suggested by the original FSTs, and shown in (10). In conclusion, there are two aspects of the output function for Turkana that demonstrate unbounded circumambience: the explicit unbounded bidirectional dependence in line (11a) and the necessary ordering of (11c–d) before (11c–f).

(10) Output function of a bimachine for Maasai harmony

$$\omega(Q_{\mathcal{L}}, x, Q_{\mathcal{R}}) = \begin{cases} [\text{o}] & \text{if } x = /\text{ɑ}/ \quad \wedge Q_{\mathcal{L}} = l_2 & \text{(a)} \\ [\text{+ATR}] & \text{if } x \in [-\text{low}] \wedge Q_{\mathcal{L}} = l_2 & \text{(b)} \\ [\text{+ATR}] & \text{if } x \in [-\text{low}] \wedge Q_{\mathcal{R}} = r_2 & \text{(c)} \\ x & \text{otherwise} & \text{(d)} \end{cases}$$

In these bimachine representations, the automata states keep track of whether the information from their causal past (prefixes for the left-to-right machine $\mathcal{L}$, suffixes for the right-to-left machine $\mathcal{R}$) licenses a change or not. The states labeled $q_2^{\mathcal{L}}$ (for $\mathcal{L}$) and $q_2^{\mathcal{R}}$ (for $\mathcal{R}$) are the states that license spreading; other states bookkeep whether the machine has encountered a spreading trigger, a blocker, or word boundaries. We then define the output function ω in a piecewise fashion. Crucially, note that the output symbol never depends on *both* information from the left *and* information from the right. That is, any conjunction within the conditions of the piecewise-defined output function can be defined using states from at most *one* of the two automata. The implicit ordering of components of a piecewise function definition means, for example, that line (10c) is only used if the conditions for lines (10a–b) fail to apply to the current input — i.e. if the negation of conditions for all lines prior to (10c) hold. This may superficially appear to suggest that line (10c) requires information not only about the righthand side of the string, but also the lefthand side; however, lines (10a–c) of the output function can be re-ordered without changing the behavior of the string function, demonstrating that information from both sides of the string is not in fact necessary to determine any output string.

In contrast, recall that in the Turkana harmony pattern, the realization of a (suffix) low vowel depends on information on both sides of the vowel: whether there is a preceding [+ATR] vowel (in the root or in a dominant [+ATR] suffix), *and* whether there is a following exceptionally dominant [−ATR] suffix vowel. The lefthand column of Fig. 7 shows an EM composition implementing the analysis sketched earlier in §3.2.

The first, left-to-right pass spreads [+ATR] (if present) to non-low vowels, and /ɑ/ is re-paired to |o|. Exceptionally dominant [−ATR] suffix vowels (represented in the figure as 'V') block [+ATR] spreading, but do not spread [−ATR] in this direction (recall the discussion after the examples in (9) in §3.2). The subsequent right-to-left pass takes advantage of the behavior of the first pass to effectively behave as though it was able to 'look ahead' unboundedly leftward. This is because the ATR value of the final vowel of the intermediate string provides effectively unambiguous information about the rest of the string. In this sense, the Turkana pattern does not meet the definition for non-interaction described in §4.2.

If the final vowel of the intermediate string is [+ATR], then there must be a dominant [+ATR] root or suffix vowel in the input string (possibly, but not necessarily, the final vowel itself). In this case, the right-to-left pass spreads [+ATR] leftward, beyond the leftmost source of [+ATR] spreading encountered by the first pass, subject to blocking by low vowels and exceptionally dominant [−ATR] suffix vowels. If either type of blocking vowel is encountered, all vowels further to the left are output as [−ATR]. In the case of a low vowel, this is because there cannot have been a dominant [+ATR] vowel further to the left; if there had been, the prior left-to-right pass would have re-paired /ɑ/ to |o|. In the case of an exceptionally dominant [−ATR] suffix vowel, this is because [−ATR] spreads from that exceptionally dominant vowel leftward, changing otherwise dominant [+ATR] vowels to [−ATR].

Alternatively, if the final vowel of the intermediate string is [−ATR], then one of two conditions must

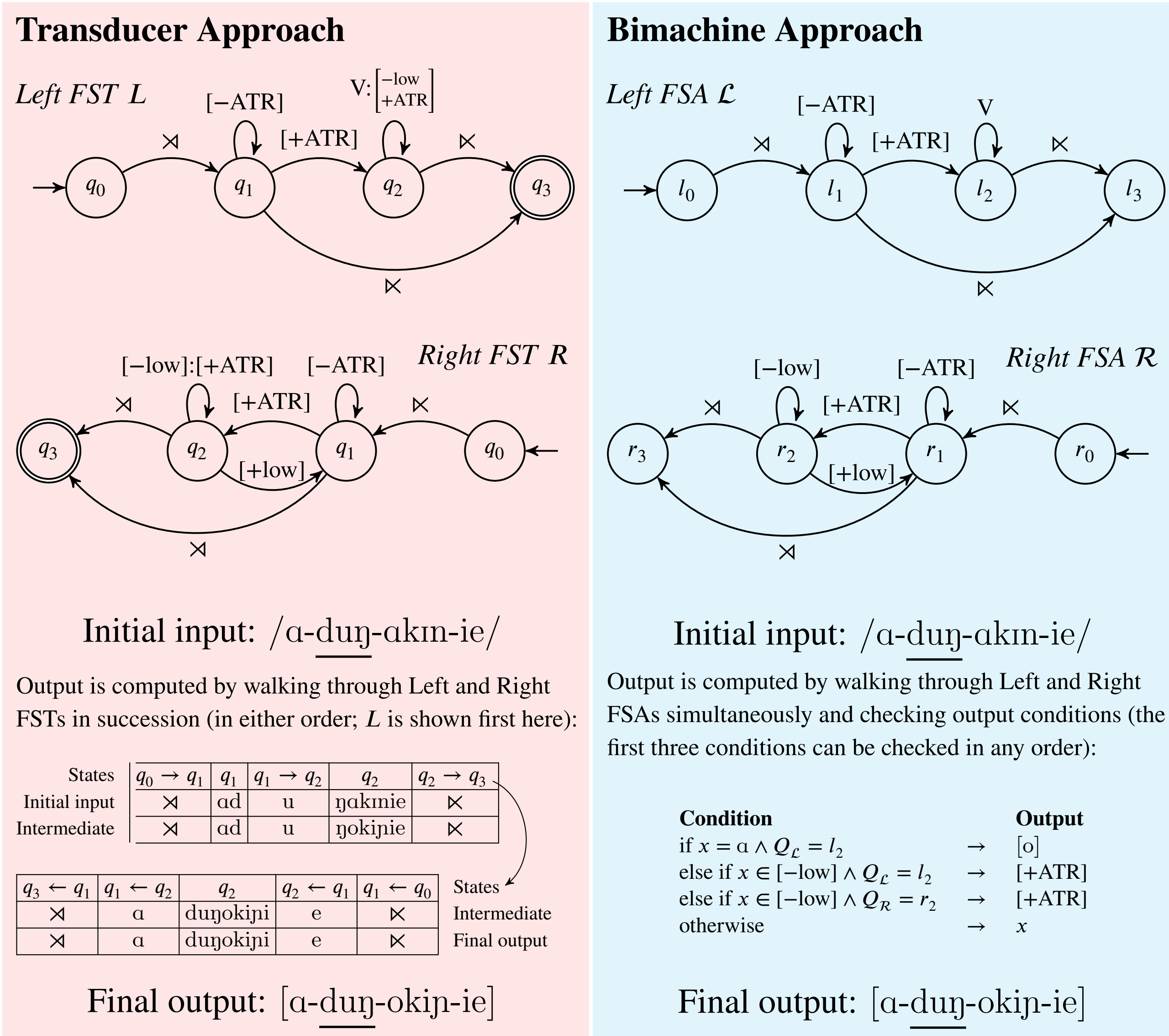

Figure 6 A comparison of transducer and bimachine approaches to Maasai harmony. (Note that for simplicity, transitions to account for the palatalization of /n/ ↦ [ɲ] have been omitted.)

hold: (i) there is an exceptionally dominant [−ATR] suffix vowel somewhere in the input string (possibly, but not necessarily, the final vowel itself), or (ii) all vowels in the input string are recessive [−ATR] vowels. In this case, the right-to-left pass maps all vowels to [−ATR], in subcase (i) by spreading [−ATR] leftward, and in subcase (ii) by simply mapping all the recessive vowels faithfully. Note that in subcase (i), the right-to-left pass will undo any [+ATR] feature changes introduced by the left-to-right pass into the intermediate string — and crucially, what began as /ɑ/ and was re-paired to |o| ultimately becomes [ɔ].

Note that identifying interaction in an unbounded circumambient map (like that of Turkana) presented as an EM composition is not in general easy to spot from inspecting definitions of the two composands. In contrast, the bimachine representation of the Turkana harmony pattern makes clear the origins of the information necessary to determine any given output symbol. The two-sided context that determines whether an input low vowel /ɑ/ will ultimately be output as [ɔ] can be read directly from the first line of the bimachine output function; else-wise, the second and last lines determine whether input /ɑ/ will be output as [o] or

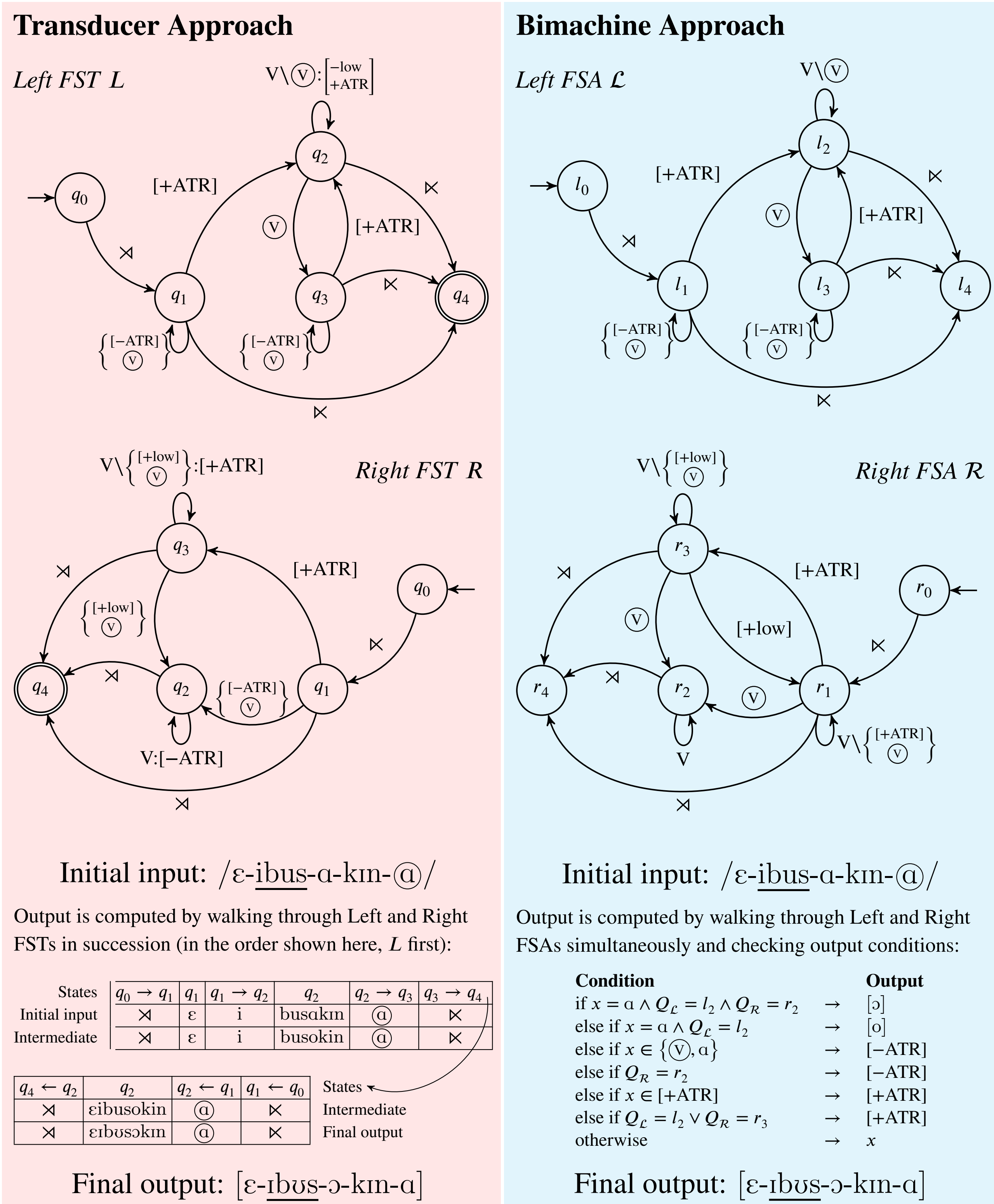

Figure 7 A comparison of transducer and bimachine approaches to Turkana harmony.

faithfully as [ɑ], respectively, as shown in (11).[20]

(11) Output function of a bimachine for Turkana harmony

$$\omega\big(Q_{\mathcal{L}}, x, Q_{\mathcal{R}}\big) = \begin{cases} [\text{ɔ}] & \text{if } x = /\text{ɑ}/ \wedge Q_{\mathcal{L}} = l_2 \wedge Q_{\mathcal{R}} = r_2 & \text{(a)} \\ [\text{o}] & \text{if } x = /\text{ɑ}/ \wedge Q_{\mathcal{L}} = l_2 & \text{(b)} \\ [-\text{ATR}] & \text{if } x \in \big\{/\text{V}/, /\text{ɑ}/\big\} & \text{(c)} \\ [-\text{ATR}] & \text{if } Q_{\mathcal{R}} = r_2 & \text{(d)} \\ [+\text{ATR}] & \text{if } x \in \ [+\text{ATR}] & \text{(e)} \\ [+\text{ATR}] & \text{if } Q_{\mathcal{L}} = l_2 \vee Q_{\mathcal{R}} = r_3 & \text{(f)} \\ x & \text{otherwise} & \text{(g)} \end{cases}$$

If the left automaton $\mathcal{L}$ is in state l_2, the right automaton $\mathcal{R}$ is in state r_2, and the current input symbol is /ɑ/, then it will be mapped to [ɔ] (11a). All other lines in the output function require information about the state of only one of the bimachines's two automata. In contrast, for the transducer approach to modeling ND functions, precise identification of which information unboundedly far away in the suffix (prefix) will ultimately disambiguate an incrementally ambiguous prefix (suffix) is in general implicit and effortful to extract from the structure of an arbitrary left-to-right (right-to-left) non-deterministic transducer. This information is difficult to extract even from the transducers shown in Fig. 7, which were constructed with this exact kind of ease of readability in mind.

Crucially, we can see from the form of the output function for Turkana in (11) that the realization of the low vowel depends on information about both the left and right (i.e. it is unbounded circumambient). Together with the knowledge that the states of the left and right automata encode information from unboundedly far to each side, we can see that this is different from the output function for Maasai in (10). In this analysis, we have focused on the unbounded circumambience required to map /ɑ/ to [ɔ]. However, it is *also* the case that all other input-output ATR mappings in Turkana participate in an unbounded circumambient pattern. Whereas the unbounded circumambience of the low vowel's realization is easy to see from line (11a) of the Turkana bimachine's output function, the unbounded circumambience of the other vowel mappings is obscured by the implicit ordering of the output function's top-to-bottom evaluation as a piecewise function. That is, the conditions that license [+ATR] spreading, lines (11e–f), depend on unbounded circumambient information because they include not only the determination that there is a dominant [+ATR] vowel to the left of the input index, but also that there is *not* an exceptionally dominant [−ATR] vowel to the right of the input index — because otherwise, the prior-ordered lines (11c–d) would have already ensured leftward [−ATR] spreading. In this way, both the conjunctive structure of output condition (11a) and the non-trivial ordering of conditions (11c–d) and (11e–f) belie the unbounded circumambience of the Turkana pattern.

4.4 Weakly deterministic functions and bimachines

To crystallize the intuition that WD ought to be the class that captures unbounded semiambient patterns, we can define the WD functions as the subclass of ND functions definable with a bimachine whose output function meets a specific condition:

Definition 6. A function f is weakly deterministic iff there exists a bimachine implementing f (per Def. 4) such that its output function ω exhibits the following property for all $q^{\mathcal{L}}$, x, y, and $q^{\mathcal{R}}$:

$$\omega(q^{\mathcal{L}}, x, q^{\mathcal{R}}) = y \iff \big(\forall r \in Q_{\mathcal{R}}, \omega(q^{\mathcal{L}}, x, r) = y\big) \vee \big(\forall l \in Q_{\mathcal{L}}, \omega(l, x, q^{\mathcal{R}}) = y\big)$$

[20] Note, we have split up the conditions in the output function for [−ATR] and [+ATR] spreading into a base case on one line and a recursive spreading case on another because of typesetting, clarity, and consistency; there is no other significance to this choice.

In words, this says that if ω is the output function of a WD bimachine and it maps x to y when its left-to-right automaton is in state $q^{\mathcal{L}}$ and its right automaton is in state $q^{\mathcal{R}}$, then either the input-output mapping $\omega(q^{\mathcal{L}}, x, q^{\mathcal{R}}) = y$ is explainable by $q^{\mathcal{L}}$ alone or it is explainable by $q^{\mathcal{R}}$ alone: either ω mapping x to y is licensed by the left state being $q^{\mathcal{L}}$ or it is licensed by the right state being $q^{\mathcal{R}}$. Note that for a given bimachine implementing a WD function and a given input word, the direction to 'look' that determines the output for a given input symbol may be leftwards for some symbols and rightwards for others, and it is not in general a priori predictable *which* direction the determining information will come from.

To be clear, per the semantics of inclusive-or ($\vee$), Def. 6 does not preclude that mapping x to y may be *independently* licensed by both the current left state and current right state; it simply forbids bimachines where the output of some symbol x depends on simultaneous knowledge of *both* the left state and the right state having particular values. The definition also does not require *a priori* knowledge or computation of *which* one side (if any) has information that licenses mapping input symbol x_i to output y_i. Operationally, in a bimachine known to be weakly deterministic, one may look at the current input symbol x and left-to-right state $q^{\mathcal{L}}$ and check if that state alone licenses mapping x to some y, and if $q^{\mathcal{L}}$ does not, then one may proceed to check if the current input symbol and the current right-to-left state $q^{\mathcal{R}}$ alone licenses mapping x to some y. Moreover, again given the semantics of inclusive-or ($\vee$), it will never matter which order one looks at the left vs. right states, or that sometimes one ends up looking at both to check if at least one licenses mapping x to y. Instead, what matters is the directionality of information that licenses mapping input symbol x to output y.

In solidifying the boundary between weak determinism and non-determinism, we have focused exclusively on patterns that require the use of information at an in principle unbounded distance away from the target of a change, and accordingly, we have centered the directionality of information in our discussions of that boundary. This may leave one with questions about how subsequential patterns with bounded bidirectional contexts are handled by our definition. Consider $A \longrightarrow B/C\_\_D$. While the most intuitive origin semantics for this map does require information from both sides and hence the map may appear superficially non-deterministic, the information required is not unboundedly far in either direction and there are alternative origin semantics that are clearly input strictly local. The relevant left-to-right origin semantics maps every input substring CA to the empty string and then either observes a D and maps it to CBD or observes a non-D symbol X and maps it to CAX; the relevant right-to-left origin semantics is symmetrically analogous, mapping AD to the empty string and then either observing a C and mapping it to CBD or observing a non-C symbol X and mapping it to XAD. In both origin semantics, we know that a function only requires lookahead corresponding to at most one symbol beyond the current 'focal' position. Because both origin semantics can be implemented by a bimachine,[21] the existence of both of these origin semantics independently shows that the string function defined by $A \longrightarrow B/C\_\_D$ is clearly subsequential, and in fact input strictly local. Output mappings for all input words can be performed with information from only one of the bimachine's automata.

If the selection of an output symbol only ever depends on information at an unbounded distance from at most a single and consistent direction that is always known a priori, then it is a subsequential function. Alternatively, if the selection of an output symbol only ever depends on information at an unbounded distance from at most a single direction that is not necessarily known a priori and need not in general be uniform from symbol to symbol within the same input word, then it is a weakly deterministic function as defined in Def. 6 and as exemplified by our analysis of Maasai. Alternatively, if the selection of an output symbol sometimes depends on information at an unbounded distance in both directions simultaneously, then it is a non-deterministic regular function, as exemplified by our analysis of Turkana. In this sense, our definition of weak determinism in Def. 6 aligns precisely with the definition of interaction introduced in Def. 1 and crystalized in Def. 5: ND functions involve interaction, while WD functions do not.

[21] That is, a bimachine with one automaton that by construction does not play any role in the operation of the bimachine (e.g. by only having one state) and hence does not provide unbounded lookahead.

5 Discussion

This work presents a novel formal definition of weak determinism, centering interaction as the core property that distinguishes weakly deterministic maps from non-deterministic maps within the computational complexity hierarchy of regular functions illustrated in Fig. 1. Centering interaction as the arbiter of this boundary, rather than markup (Heinz and Lai 2013, McCollum et al. 2018, O'Hara and Smith 2019, Smith and O'Hara 2019, Lamont et al. 2019) or lookahead (Jardine 2016, McCollum et al. 2020a), strengthens the status of weakly deterministic maps as a computational class of linguistic interest because it is a computationally stronger and linguistically more natural concept than those centered in previous formal definitions.

5.1 Interaction is linguistically natural

Interaction meshes well with existing concepts in linguistics. Most obviously, it has a great deal in common with the type of interaction that motivates serial rule ordering in the *SPE* tradition of Chomsky and Halle (1968) to distinguish e.g. 'feeding' from 'counterfeeding' and 'bleeding' from 'counterbleeding' (Chafe 1968, Kiparsky 1968, Wang 1969, Newton 1971, Koutsoudas et al. 1974, Kenstowicz and Kisseberth 1977, 1979, among many, many others).[22] After all, a pair of interacting *SPE* rewrite rules is essentially a specific ordered composition of regular functions $g \circ f$ (Johnson 1972, Kaplan and Kay 1994). Zooming out somewhat, there is also much in common between our concept of interaction and the type of interaction that motivates constraint ranking and weighting in the traditions of OT (Prince and Smolensky 2004) and HG (Legendre et al. 1990). Any insights gained from the study of any of these forms of interaction are thus highly likely to inform the study of the others — as they indeed already have; see for example Norton (2003), Baković (2007, 2011, 2013), and references therein for relevant formal comparisons of the descriptive and explanatory similarities and differences between these serial rule-based and parallel constraint-based theoretical frameworks.[23]

Bringing our focus back to the more immediate concerns of this article, interaction unites all cases of unbounded circumambience and is not limited to those cases that satisfy arbitrary criteria resulting from particular decisions about what counts as markup or about what it means for a function to 'look ahead' in the string. The Turkana harmony pattern is an instructive case in point. There is clear linguistic evidence for bidirectional spreading from [+ATR] roots and dominant [+ATR] suffixes, for low vowels undergoing rightward but not leftward spreading of [+ATR], and for leftward spreading from dominant [−ATR] suffixes. Most significantly, rightward spreading of [+ATR] interacts with leftward spreading of [−ATR], the latter in some sense has the 'final say' on the realization of vowels in a string, and each direction of spreading corresponds to one of the contradirectional FSTs of an EM composition or FSAs of a bimachine. The fact that the surface realization of an underlying low vowel depends on unbounded circumambient information is thus effectively disguised by the independent motivation for each of the interacting spreading processes.

This *feels* unlike other non-myopic patterns such as unbounded tonal plateauing (Jardine 2016; see §5.2 below) or Tutrugbu ATR harmony (McCollum et al. 2020a), where the linguistically motivated spreading process only corresponds to one of the two contradirectional FSTs of the EM composition or FSAs of the bimachine, albeit roughly. The other FST/FSA must be more indirectly inferred from the information in the causal future of the linguistically motivated one that makes the overall pattern non-deterministic. This is what makes some obvious form of markup necessary in those cases. The total mapping in Turkana is unbounded circumambient and involves lookahead, but precisely because non-myopia is not analytically salient, interaction remains a linguistically useful tool for identifying that the subregular complexity of Turkana is the same as other more obviously non-myopic unbounded circumambient maps.

[22] For a review of relevant concepts and terms, see Baković (2011); for a more thorough formal exploration of the typology of rule ordering interactions, see Baković and Blumenfeld (2018, 2019, 2020).

[23] The focus in the text on the extremes of this continuum is not meant to ignore the wealth of work aiming for some kind of balance between serialism and parallelism, ranging from Stratal OT (Kiparsky 2015; Bermúdez-Otero, in prep.), to Harmonic Serialism (McCarthy 2010), to Serial HG (Pater 2012).

5.2 Interaction subsumes markup

Heinz and Lai's (2013) vision of weak determinism as a restriction on regular functions implemented through constraints on function composition remains relevant to the reformulation of WD functions presented in §4. WD functions remain a class of functions that "can be decomposed into [an inner] subsequential and [an outer, contradirectional] subsequential function *without the [inner] function marking up its output in any special way*"; merely the piecewise nature of initial attempts to formally define "any special way" have been exchanged for the unifying notion of interaction described in this article. And while we argue that bimachines make the identification of interaction far easier than EM compositions do, analysts may still choose to represent WD processes using non-interacting EM compositions. For that reason, we discuss here how even for EM composition representations of the WD class, the use of markup-based definitions of the WD class should be abandoned because interaction subsumes markup.

Heinz and Lai's (2013) definition of the WD functions identified two possible markup strategies and accordingly stipulated that WD functions be alphabet- and length-preserving. In identifying and prohibiting strategies case by case, the focus on markup in this definition of weak determinism inspired further work identifying novel variations on these two markup strategies. In these works, an even more clever, third markup strategy has been identified, a form we call PHONOTACTIC CODING.[24] In this strategy, substrings of the input alphabet which would otherwise be phonotactically illicit in the language in question are used to smuggle information from an inner function to an outer function via the intermediate form. More specifically, given the input alphabet $\mathcal{A}$ and a set $\mathcal{S}$ of non-occurring, phonotactically illicit substrings over $\mathcal{A}$, the inner function may change some substring of the input to a substring of the same length drawn from $\mathcal{S}$, conditioned on some motivating property of the input string. If this change is made at a location in the string that the outer function would encounter before it would encounter the motivating property of the input string, then the output of the outer function at that point may be conditioned on information possibly an unbounded distance from that decision point, rendering the composition as a whole non-deterministic. For this to occur, it is only required that each otherwise illicit substring introduced by the inner function be unambiguously interpretable by the outer function (i.e. from the opposite direction) as having been introduced because the inner function encountered particular conditions somewhere in the (unbounded) causal future of the outer function. The result is alphabet-preserving because the set $\mathcal{S}$ is defined over the input alphabet $\mathcal{A}$, and it is length-preserving because substrings from $\mathcal{S}$ only replace substrings of the same length.

Fig. 8 illustrates phonotactic coding with a schematic example of unbounded tonal plateauing (UTP), with the inner function marking up its output with the phonotactic code suggested by Lamont et al. (2019, p. 197). In cases of UTP, all low (L) tones between two high (H) tones surface as H; other Ls surface as L.[25] These are quintessential cases of unbounded circumambience: the output of a given L is dependent on information that is unboundedly far to the left and information that is unboundedly far to the right.

The phonotactic code in this case is the intermediate HLH substring, written in one fell swoop by the inner function f upon encountering an H preceded by two or more Ls which are themselves preceded by another H. More precisely, after reading and faithfully writing an initial string of some number l of Ls (arrow 1) followed by the first H (arrow 2) in the input string x, the inner function f reads an L. At this point, nothing is written — hence the dotted arrow 3 and the corresponding empty symbol λ in the intermediate string y. If the next

[24] The works cited call this "using phonotactically-illicit intermediate symbol sequences as codes for extra symbols," (McCollum et al. 2018), "(intermediate) substring markup" (O'Hara and Smith 2019), "using predictable substrings as markup" (Lamont et al. 2019), and "information [that] is smuggled into an intermediate representation using predictable substrings of the symbols already in a language's alphabet" (Smith and O'Hara 2019). Graf (2016) makes a related proposal, strategically identifying auxiliary symbols from the input alphabet ("green flags") that can be used as markup.

[25] Jardine (2016, pp. 251ff) cites Luganda, Digo, Xhosa, Zulu, Yaka, Saramaccan, Papago (Tohono O'odham), and South Kyungsang Korean as examples of languages with some variant of UTP; see that work for discussion and references to sources. In some cases the underlying tonal distinction is analyzed as H vs. toneless rather than H vs. L. This distinction is irrelevant here; all that matters is that there is a two-way opposition, regardless of how it is represented.

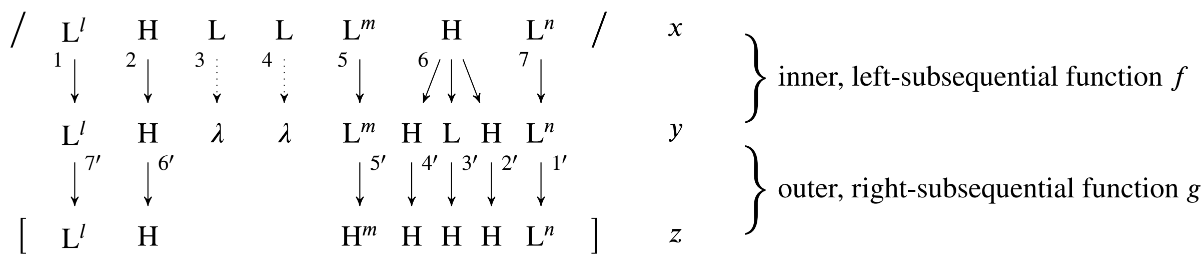

Figure 8 Schematic example of unbounded tonal plateauing, with phonotactic coding

symbol to be read were an H, the 2-symbol sequence HH would be emitted: the first H taking the place of the previously unwritten L and the second being the faithful realization of the H just read. Lamont et al. (2019, p. 197) note that this HLH ↦ HHH mapping "model[s] UTP in a local context". This idiosyncratic treatment of plateaux of length 3 is crucial for this markup strategy to succeed.

The inner function f instead reads another L and once more writes nothing (arrow 4). A subsequent string of some number m of Ls is then written faithfully (arrow 5), and upon reading another H, f writes the sequence HLH (the trio of arrows labeled 6). This compensates for the two previous unwritten Ls, and it does so with the substring HLH which, when present as such in an input string, would have been written by the very same inner function f as HHH (= "UTP in a local context"). HLH thus functions as a phonotactic code, informing the outer function g that the conditions for non-local UTP were found in the input string x and that the L of this HLH coded substring in y, and all subsequent Ls until another H is read, should surface as H in the ultimate output string z.

To be clear, the works cited above acknowledge that phonotactic coding is a form of markup (see fn. 24), and indeed that "all unbounded circumambient patterns require some sort of markup to be used in the application of a pair of subsequential mappings" (O'Hara and Smith 2019, p. 10).[26] Nevertheless, phonotactic coding has been claimed in these works to critically distinguish unbounded circumambient patterns that can be described by WD functions from those requiring more expressive classes of functions because the contradirectional breakdowns of these latter functions crucially require increasing either the alphabet or the length of the string, the only forms of markup explicitly precluded by Heinz and Lai's (2013) formal definition of WD functions. Perhaps most explicitly, O'Hara and Smith (2019, p. 6) claim that alphabet- or length-increasing markup is "more powerful" than phonotactic coding of the kind illustrated in Fig. 8 above. As evidence for this claim, these authors argue that the unattested set of "true sour grapes" patterns, defined in (12), can be modeled with alphabet- or length-increasing markup but not with phonotactic codes.[27] These patterns are intended to be distinguished from "false sour grapes" patterns, a set which includes attested UTP.

(12) An unbounded spreading pattern is an example of *true sour grapes* if, given $\Sigma^* = \{\text{T, B, U}\}$ (T = triggers, B = blockers, U = undergoers), and for all $m, n \in \mathbb{N}$,

	if left-to-right…	*if right-to-left…*
a.	$\text{T}^m\text{U}^n\text{B} \mapsto \text{T}^m\text{U}^n\text{B}$, and	$\text{B}\text{U}^n\text{T}^m \mapsto \text{B}\text{U}^n\text{T}^m$, and
b.	$\text{T}^m\text{U}^n \mapsto \text{T}^{m+n}$.	$\text{U}^n\text{T}^m \mapsto \text{T}^{n+m}$.

O'Hara and Smith (2019) and Smith and O'Hara (2019) show that no length- or alphabet-preserving composition can be devised to correctly describe a true sour grapes pattern, whereas such compositions are

[26] Lamont et al. (2019, p. 197) even go so far as to comment negatively on the result of phonotactic coding from the perspective of substantive and methodological assumptions within phonological theory: "Encoding instructions in intermediate representations is strikingly unphonological and is not intended as a plausible interpretation of the process."

[27] On the (un)attestedness of different types of sour grapes ('non-myopic') spreading patterns, see Wilson (2003, 2006), Walker (2010), Kimper (2012), Mascaró (2019), and McCollum et al. (2020b). See also the discussion to come in §5.3 below.

possible for false sour grapes patterns (e.g. for UTP in Fig. 8). The key difference between UTP (*qua* false sour grapes) and true sour grapes is that UTP defines unbounded circumambient conditions for making a change to an input string (viz. spreading H), whereas true sour grapes defines unbounded circumambient conditions for *not* making a change to an input string. This key difference guarantees the existence of a phonotactic code for UTP, as follows. Upon encountering an H in a UTP pattern, the inner function will not spread it unless there is another H further down the line, in principle an unbounded distance away. The bounded HLH $\mapsto$ HHH mapping is an instance of the more general spreading condition ("UTP in the local context"); such bounded maps can be performed by the inner function, guaranteeing that HLH can function as a phonotactic code.

There is no such guarantee of a phonotactic code for true sour grapes. Upon encountering a trigger T, the inner function *will* spread it unless there is a blocker B further down the line, in principle an unbounded distance away. There are thus only bounded instances of the more general *non*-spreading condition, TUB, which map unchanged to TUB. Thus, no substring can be freed up to function as a phonotactic code. But it is important to keep in mind that all possible substrings of Σ^* are assumed to be equally likely in the toy language of (12): T is just as likely as B or U at every point in the string, leaving no substring of any length capable of both encoding the presence of a blocker elsewhere in the string while preserving the original identity of the substring at the location of the markup. While this toy example neatly demonstrates that phonotactic coding is a non-viable markup strategy for completely unrestricted input languages (that is, input languages where literally *any* member of Σ^* is a viable word of the language), we argue that this is not a linguistically relevant result at the very least because all languages, human or otherwise, are generally not unrestricted in this way. In other words, while a bounded instance of the more general non-spreading condition in cases of true sour grapes is not guaranteed, it is nevertheless highly probable if not fully guaranteed that *some* bounded, phonotactically-illicit substring can be recruited to do the necessary markup work (Graf 2016).

This brings us back to O'Hara and Smith's (2019) claim that alphabet- and length-increasing markup is "more powerful" than phonotactic coding. If "more powerful" is understood to mean "more *expressive*", this is in general false, as expressivity is usually defined and considered. In formal language theory, a function class picks out an extensionally-definable equivalence class of functions.[28] O'Hara and Smith (2019) argue that 'true' sour grapes can only be modeled with alphabet- and length-increasing codes while 'false' sour grapes can be modeled with *either* alphabet-/length-increasing codes *or* phonotactic codes. This is the central logic of their claim that phonotactic codes are not as expressive as alphabet- and length-increasing codes. However, the difference between the mappings of so-called 'true' and 'false' sour grapes (and thus their difference in "power" as claimed by O'Hara and Smith 2019) lies in a restriction on the input languages of the inner and outer functions that O'Hara and Smith (2019) use to describe these patterns. No complexity class that we are aware of relies so centrally on comparable restrictions of the domain of the input language.

For example, consider a majority-rules function $f(a^n b^m) = c^{max(n,m)}$ which maps strings of n a's and m b's to a string of c's whose length is either n or m, whichever value is greater. This function — this mapping between input strings and output strings — is not regular. Yet, if we (arbitrarily) restrict the input domain f to the language a^*, i.e. where no b's will ever occur, our function is suddenly regular: mapping a^n to c^n does not require supra-regular computational power. Considering the behavior of f under this domain restriction may tell us something about the relationship between the structure of the input language and the behavior of f, but the usual approach to classifying the complexity of f itself would still indicate the function is not regular.

That the entropy of some languages is too high to support phonotactic coding strategies is really a statement about those languages, not a description of the expressivity of the machinery required to describe a function on those languages. The exchangability of novel intermediate symbols, length-increasing codes for such symbols, and phonotactic codes for such symbols is apparent from the fact that they all serve the same purpose: creating information about one end of the input string at the other end of the intermediate string. It is the ability of

[28] At the very least, such a function class is expected to have multiple convergent characterizations, including an extensional, machine-independent one, even if we don't currently know what those are.

each of these three strategies to accomplish precisely this goal that permits compositions of contradirectional subsequential functions to define ND functions using any of them.

Beyond the temptation to ascribe computational significance to cosmetic differences between markup strategies, viewing weak determinism as a class defined by its resistance to description by a collection of disparate strategies is less parsimonious than the unified account provided by interaction — and it obscures the sense of 'intermediateness' that is characteristic of the class. Heinz and Lai (2013, p. 54) justify the framing of WD functions as 'weakly' or intermediately deterministic by writing: "they are not necessarily deterministic, but they are 'more' deterministic than regular functions where Elgot and Mezei decomposition *requires* the intermediate markup" (emphasis in the original). A more substantive characterization is in (13).

(13) Informal description of deterministic, weakly deterministic, and non-deterministic regular functions

a. **Deterministic (subsequential)**: there exists a side of the string, for every position in all strings, for which the identity of the output element is determined by elements from that side.

b. **Weakly deterministic**: for every position in all strings, there exists a side of the string for which the identity of the output element is determined by elements from that side.

c. **Non-deterministic**: for every position in all strings, both sides of the string may be used to determine the identity of the output element.

Thus, a WD function is intermediate in expressivity because different output positions may require information from different sides of the form (unlike deterministic functions), but no single output position requires information from *both* sides of the form (unlike non-deterministic functions). In this sense, WD functions are more deterministic than non-deterministic functions but less so than deterministic functions.

This intuition is preserved in thinking of determinism in terms of interaction. Deterministic functions are regular functions that can be defined as non-interacting compositions of equi-directional (as opposed to contradirectional) subsequential functions. Weakly deterministic functions are regular functions that can be defined as a non-interacting compositions of contradirectional subsequential functions. And finally, non-deterministic functions are regular functions that can be defined as a interacting compositions of contradirectional subsequential functions. In this way, interaction captures the essense of what makes weak determinism not quite deterministic; defining WD functions as the class of EM compositions for which the inner and outer functions do not interact is computationally more sound than attempts to exhaustively list all of the ways one can imagine marking up an intermediate string.

5.3 Machine-independence

In §1 we alluded to the value that good definitions can provide in crystalizing linguistic intuitions and in offering clarity and confidence about what linguistic formalisms can actually do or how they compare. In §2.1 and §4.1, we elaborated more specifically on how machine-independent or extensional characterizations of a language or function class can be used as relatively formalism-agnostic tools for phonological metatheory: each machine-independent characterization of a given class offers a means to completely describe and understand that class and any language or function within that class, as well as a solid foundation to understand and prove results about that class — including e.g. determining whether it is distinct from some other class.

In this article we have extended earlier work expressing the intuitive definition, linguistic motivation, and expected boundary of the WD class (Heinz and Lai 2013) by presenting two distinct classes of phenomena — exemplified by Maasai and Turkana — and then offering a definition of 'interaction' between regular string functions and a corrected definition of the WD class that separates those two classes of phenomena. We have chosen a machine-independent characterization of the regular functions transparently related to the unbounded circumambience vs. unbounded semiambience distinction and as a direct result have a definition of the WD class that is simple to understand and whose boundary with respect to ND is clear. The machine-independent characterization we have chosen is also crucial to offering a definition of interaction in regular string functions that can be used to mechanically identify any and all contexts where two string functions interact. The definition of interaction and the revised definition of WD functions also present a unified picture of how any EM composition that defines an ND function does so, regardless of whether it uses novel symbols, length-increasing codes for novel symbols, phonotactic codes for novel symbols, some as-yet unidentified new strategy for encoding novel symbols — or even no codes for novel symbols at all, as in Turkana. In short, we have provided a definition of the WD class of functions that solves the problems of Heinz and Lai's (2013) original, first-pass definition, matching the spirit of that definition by fixing the letter.

Setting aside the issue of the definition of the WD class, the type of canonical bimachines (and the associated machine-independent characterization) presented here offer an alternative to EM compositions for future work on long-distance processes in phonology. First, when presented with a single independently motivated bidirectional phenomenon, an EM composition-based analysis requires the linguist to engineer two tightly coupled subsequential functions so that they interact in just the right way to define the overall map of interest. As we have shown, there are many strategies to accomplish this, there is not in general anything linguistically interesting or insightful about any particular strategy for doing this, it is often nevertheless difficult to do correctly for analysts, and the resulting analyses are generally harder to read and understand as a consequence. Circumstances where there are two independently motivated contradirectional processes still may justify use and exploration of EM compositions, but identification of interaction or classification of a composition's complexity will likely remain easier to analyze using representations like what we have presented here: canonical bimachines offer a simpler path to identifying interaction that is otherwise difficult to spot, and to classifying an EM composition's position on one side of the WD–ND boundary vs. the other.

6 Conclusion

Phonologists have long concerned themselves with the ways that more than one grammatical statement (rules, constraints, etc.) may interact with one another. This article provides a formal definition of function interaction that differentiates properly WD functions from the strictly more expressive class of ND functions. Formal comparison of the finite-state analyses of the ATR harmony patterns of Maasai and Turkana demonstrates the key difference between WD and ND functions: the contradirectional subsequential functions comprising a WD function do not interact, while those comprising an ND function do. By framing the distinction between weak determinism and non-determinism in this way, our proposal subsumes the various strategies employed in previous work to delimit the subregular class of WD functions, thereby offering a single, unified way to characterize the expressivity difference between unbounded circumambient and other, unbounded semiambient bidirectional patterns.

References

Aksënova, A., Rawski, J., Graf, T., and Heinz, J. (2020). The Computational Power of Harmony. To appear in H. van der Hulst (ed.), *Oxford Handbook of Vowel Harmony*, Oxford University Press.

Albert, C. (1995). Derived Environments in Optimality Theory: A Case Study in Turkana. Unpublished ms., UC Santa Cruz.

Archangeli, D. B. and Pulleyblank, D. G. (1994). *Grounded Phonology*, volume 25. MIT Press.

Baković, E. (2000). *Harmony, Dominance and Control*. PhD thesis, Rutgers, The State University of New Jersey.

Baković, E. (2002). Vowel harmony and cyclicity in Eastern Nilotic. In Chang, C., Houser, M. J., Kim, Y., Mortensen, D., Park-Doob, M., and Toosarvandani, M., editors, *Proceedings of the 27th Annual Meeting of the Berkeley Linguistics Society*, pages 1–12, Berkeley, CA. Berkeley Linguistics Society.

Baković, E. (2007). A revised typology of opaque generalisations. *Phonology*, 24:217–259.

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.

Baković, E. (2013). *Blocking and Complementarity in Phonological Theory*. Equinox, London.

Baković, E. and Blumenfeld, L. (2018). Overapplication conversion. In Bennett, R., Angeles, A., Brasoveanu, A., Buckley, D., Kalivoda, N., Kawahara, S., McGuire, G., and Padgett, J., editors, *Hana-bana: A Festschrift for Junko Itô and Armin Mester*. Santa Cruz, CA.

Baković, E. and Blumenfeld, L. (2019). Rule interaction conversion operations. *Loquens*, 6(2):e062.

Baković, E. and Blumenfeld, L. (2020). A typology of map interactions. Manuscript in preparation, UC San Diego and Carleton University.

Bennett, W. G. and DelBusso, N. (2018). The typological effects of ABC constraint definitions. *Phonology*, 35(1):1–37.

Bermúdez-Otero, R. (n.d.). *Stratal Optimality Theory*. Oxford University Press, Oxford. In preparation. Pending publication, see the collection of writings available at http://www.bermudez-otero.com/Stratal_Optimality_Theory.htm.

Bojánczyk, M. (2018). Transducers with origin information. arXiv preprint arXiv:1309.6124.

Bojańczyk, M. and Czerwiński, W. (2018). *An Automata Toolbox*. Course lecture notes, available at https://www.mimuw.edu.pl/~bojan/upload/reduced-may-25.pdf.

Chafe, W. L. (1968). The ordering of phonological rules. *International Journal of American Linguistics*, 34:115–136.

Chandlee, J. (2014). *Strictly local phonological processes*. PhD thesis, University of Delaware.

Chandlee, J., Eyraud, R., and Heinz, J. (2015). Output strictly local functions. In *Proceedings of the 14th Meeting on the Mathematics of Language (MoL 2015)*, pages 112–125.

Chandlee, J. and Heinz, J. (2018). Strict locality and phonological maps. *Linguistic Inquiry*, 49(1):23–60.

Choffrut, C. (2003). Minimizing subsequential transducers: a survey. *Theoretical Computer Science*, 292(1):131–144.

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

Dimmendaal, G. J. (1983). *The Turkana Language*. Foris, Dordrecht.

Dolatian, H. and Heinz, J. (2020). Computing and classifying reduplication with 2-way finite-state transducers. *Journal of Language Modelling*, 8(1):179–250.

Eilenberg, S. (1974). *Automata, languages and machines, Volume A*. Academic 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.

Filiot, E. (2015). Logic-automata connections for transformations. In *Indian Conference on Logic and Its Applications*, pages 30–57. Springer.

Gauwin, O. (2020). *Transductions: resources and characterizations*. Habilitation à diriger des recherches, Université de Bordeaux.

Graf, T. (2016). Weak determinism is not too weak: A reply to Jardine (2016). Unpublished ms., Stony Brook University.

Guion, S. G., Post, M. W., and Payne, D. L. (2004). Phonetic correlates of tongue root vowel contrasts in maa. *Journal of Phonetics*, 32(4):517–542.

Hall, B. L., Hall, R. M. R., and Pam, M. D. (1973). African vowel harmony systems from the vantage point of Kalenjin. *Afrika und Übersee: Sprachen, Kulturen*, 57(4):241–267.

Hao, Y. and Andersson, S. (2019). Unbounded Stress in Subregular Phonology. In *SIGMORPHON 16*, pages 135–143. ACL.

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*, 23:126–195.

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.

Jardine, A. (2016). Computationally, tone is different. *Phonology*, 33(2):247–283.

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

Johnson, C. D. (1972). *Formal aspects of phonological description*. Mouton, The Hague.

Kaplan, R. M. and Kay, M. (1994). Regular models of phonological rule systems. *Computational linguistics*, 20(3):331–378.

Kenstowicz, M. and Kisseberth, C. (1977). *Topics in phonological theory*. Academic Press.

Kenstowicz, M. and Kisseberth, C. (1979). *Generative phonology: Description and theory*. Academic Press.

Kimper, W. A. (2012). Harmony is Myopic: Reply to Walker (2010). *Linguistic Inquiry*, 43(2):301–309.

Kiparsky, P. (1968). Linguistic universals and linguistic theory. In Bach, E. and Harms, R., editors, *Universals in linguistic theory*. Holt, New York.

Kiparsky, P. (2015). Stratal OT: A Synopsis and FAQs. In Hsiao, Y. E. and Wee, L.-H., editors, *Capturing Phonological Shades Within and Across Languages*, pages 2–44. Cambridge Scholars Library, Newcastle upon Tyne.

Koser, N. and Jardine, A. (2020). The computational nature of stress assignment. In *Proceedings of the 2019 Annual Meeting on Phonology*.

Koutsoudas, A., Sanders, G., and Noll, C. (1974). On the application of phonological rules. *Language*, 50:1–28.

Lamont, A., O'Hara, C., and Smith, C. (2019). Weakly deterministic transformations are subregular. In *Proceedings of the 16th Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 196–205.

Legendre, G., Miyata, Y., and Smolensky, P. (1990). Harmonic Grammar – A formal multi-level connectionist theory of linguistic well-formedness: Theoretical foundations. In *Proceedings of the Twelfth Annual Conference of the Cognitive Science Society*, pages 388–395. Lawrence Erlbaum Associates, Inc., Mahwah, NJ.

Lhote, N. (2018). Towards an algebraic characterization of rational word functions. *CoRR*, abs/1506.06497.

Lombardi, L. (1999). Positional faithfulness and voicing assimilation in Optimality Theory. *Natural Language & Linguistic Theory*, 17(2):267–302.

Mascaró, J. (2019). On the lack of evidence for nonmyopic harmony. *Linguistic Inquiry*, 50:862–872.

McCarthy, J. J. (2010). An introduction to harmonic serialism. *Language and Linguistics Compass*, 4(10):1001–1018.

McCollum, A. G., Baković, E., Mai, A., and Meinhardt, E. (2018). The expressivity of segmental phonology and the definition of weak determinism. *Unpublished ms., University of California San Diego*.

McCollum, A. G., Baković, E., Mai, A., and Meinhardt, E. (2020a). Unbounded circumambient patterns in segmental phonology. *Phonology*, 37:215–255.

McCollum, A. G., Baković, E., Mai, A., and Meinhardt, E. (2020b). On the existence of non-myopic harmony. Unpublished manuscript.

McCrary, K. (2001). Low Harmony, ATR Harmony and Verbal Morphology in Kisongo Maasai. *Papers in Phonology*, 5:179–206. UCLA Working Papers in Linguistics, no. 7.

Mihov, S. and Schulz, K. U. (2019). *Finite-State Techniques: Automata, Transducers and Bimachines*. Cambridge University Press.

Newton, B. E. (1971). Ordering paradoxes in phonology. *Journal of Linguistics*, 7:31–53.

Norton, R. J. (2003). *Derivational phonology and optimality phonology: formal comparison and synthesis*. PhD thesis, Rutgers, The State University of New Jersey.

Noske, M. (1996). [ATR] harmony in Turkana. *Studies in African Linguistics*, 25(1):61–99.

Noske, M. (2000). [ATR] Harmony in Turkana: A Case of Faith Suffix » Faith Root. *Natural Language & Linguistic Theory*, 18(4):771–812.

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

O'Hara, C. and Smith, C. (2019). Computational complexity and sour-grapes-like patterns. In *Proceedings of the 2018 Annual Meeting on Phonology*, volume 7.

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. (2012). Serial Harmonic Grammar and Berber syllabification. In Borowsky, T., Kawahara, S., Shinya, T., and Sugahara, M., editors, *Prosody Matters: Essays in Honor of Elisabeth O. Selkirk*, pages 43–72. Equinox, London.

Peikov, I. (2007). Rule Definition Language and a Bimachine Compiler. *Workshop on Finite-State Techniques and Approximate Search*, page 42.

Prince, A. and Smolensky, P. (2004). *Optimality Theory: Constraint interaction in generative grammar*. Wiley-Blackwell.

Quinn-Wriedt, L. T. (2013). *Vowel Harmony in Maasai*. PhD thesis, University of Iowa.

Reutenauer, C. and Schützenberger, M.-P. (1991). Minimization of rational word functions. *SIAM Journal on Computing*, 20:669–685.

Roche, E. and Schabes, Y. (1997). *Finite-State Language Processing*. MIT Press.

Schützenberger, M.-P. (1961). A remark on finite transducers. *Information and Control*, 4:185–196.

Smith, C. and O'Hara, C. (2019). Formal characterizations of true and false sour grapes. In *Proceedings of the Society for Computation in Linguistics*, volume 2, pages 338–341.

Tucker, A. N. and Mpaayei, J. T. O. (1955). *A Maasai grammar, with vocabulary*. Longmans, Green.

Vago, R. M. and Leder, H. (1987). On the Autosegmental Analysis of Vowel Harmony in Turkana. In Odden, D., editor, *Current Approaches to African Linguistics*, volume 4, pages 383–395. Foris, Dordrecht.

van der Hulst, H. (2018). *Asymmetries in Vowel Harmony*. Oxford University Press.

Walker, R. (2010). Nonmyopic harmony and the nature of derivations. *Linguistic Inquiry*, 41(1):169–179.

Wang, W. S.-Y. (1969). Competing changes as a cause of residue. *Language*, 45:9–25.

Wilson, C. (2003). Analyzing unbounded spreading with constraints: marks, targets, and derivations. Unpublished manuscript, UCLA.

Wilson, C. (2006). Unbounded spreading is myopic. Paper presented at the Phonology Fest Workshop on Current Perspectives on Phonology, Indiana University.