The Ticuna nominalizer tone circle and input strict-locality *

Chris Oakden, Rutgers University

July 7, 2020

Abstract

This paper identifies input strict-locality as a computational property of grammatical tone maps, a property shared with many tonal and segmental processes in phonology. Through a computational account of nominalizer tone circles in Cushillococha Ticuna (and an opaque interaction with a dissimilatory tonal operation), the current study shows that grammatical tone necessitates a framework that allows explicit reference to input structure. Crucially, the computational characterization pursued here defines both the tone circle and tonal dissimilation as a single map, meaning that they operate over the same input tonal and morphological structure in tandem. The input-orientedness of these definitions permits a straightforward account of the Ticuna data. It also identifies input strict-locality as a property of circular chain shifts in general; circular shifts—like the Ticuna nominalizer tone circle—are therefore rather simple from a computational perspective, despite the difficulties they pose to rule-based and constraint-based theories of phonology.

1 Introduction

Rolle (2018, 2) defines grammatical tone (GT) as "a tonological operation which is restricted to the context of a specific morpheme or construction, or a natural class of morphemes or constructions." To date, little work has been done to determine the computational properties of GT, or how it relates to non-grammatically-determined tonal processes in phonology. This paper offers an initial exploration into the computational properties of grammatical tone by studying the nature of input-output GT mappings. It identifies *input strict-locality* (Chandlee, 2014; Chandlee and Heinz, 2018) as a property of GT operations, a property shared by many tonal and segmental processes in phonology. Through a computational account of nominalizer tone circles in Cushillococha Ticuna (CT, Anderson, 1962; Skilton, 2017) and an opaque interaction with a dissimilatory tonal operation, the current study shows that GT necessitates a framework that allows explicit reference to input structure.

CT tone circles are formalized as functions that compute outputs using a bounded window of tonal and morphological information in the input. Analysis of the CT data is implemented using *boolean monadic recursive schemes* (Bhaskar et al., 2020; Chandlee and Jardine, 2020); this formalism makes explicit the computational properties (input strict-locality) of the maps. It also elucidates the phonologically-significant generalizations underlying them in a more direct manner than other computational formalisms.

Crucially, the computational characterization pursued here defines both the tone circle and tonal dissimilation as a single map, meaning that they operate over the same input structure in tandem. The input-orientedness of these definitions permits a straightforward account of the CT data. It also identifies input strict-locality as a property of circular chain shifts in general; circular chain shifts in GT—like the CT nominalizer tone circle—are therefore rather simple from a computational perspective, despite the difficulties they pose to rule-based and constraint-based theories of phonology.

This paper is organized as follows. §2 introduces the relevant data from CT, including the morphologically-conditioned tone circle and its interaction with a tonal dissimilation pattern. The main analysis is presented in §3. Important preliminaries are introduced first (§3.1), followed by computational accounts of the tone circle (§3.2) and the opaque interaction (§3.3). §4 evaluates the computational account against previous analyses of CT and of circular chain shifts more generally, and discusses its ramifications. §5 concludes the paper.

*I wish to thank Nic Rolle for introducing me to grammatical tone, and Adam McCollum for introducing me to Ticuna.

2 Nominalizer tone circle

This section introduces the relevant CT facts—the basic tone circle and its interaction with a dissimilatory process—to be analyzed. It is based on fieldwork data reported in (Skilton, 2017).

CT has five contrastive level tones, denoted 1-5 where 5 represents the highest tone in a speaker's register. It also contains four contour tones, all of which are falling: 31, 43, 51, and 41 using Skilton's notation. Not all tones participate in the tone circle and/or dissimilation, so the focus in this paper is on those tones which do participate.

Skilton (2017, 22) claims that only level tones are phonemic, while contour tones are derived, resulting from 'the assignment of two tones to a single TBU'. This paper does not take a stance on this issue, but assumes that level and contour tones serve equally as inputs to the tone circle and dissimilatory processes.

2.1 Basic pattern

The CT nominalizer tone circle is a morphologically-conditioned tonal pattern, triggered when one of four class-specific nominalizer suffixes (NOM) attaches to a verb stem.[1] This alternation targets the final syllable of the verb stem, either a root or other intervening constituent: verbal suffix, enclitic, etc. Example forms are given in (1) as they appear in (Skilton, 2017, 48-52). Tones are indicated with a superscript to the right of the syllable to which they associate, e.g. 'na^4' denotes a syllable [na] bearing tone 4. Additionally, affected syllables are shown in bold.

(1)

a. Tone 1̰

Isolation /ŋa̰^{1}/ 'be raw'	/na^4=ŋa̰^{1}/	→	[na^{4}ŋa̰^{1}]	'it is raw'
With C-IV NOM ʔɨ̃^{4}	/na^4=ŋa̰^{1}-ʔɨ̃^{4}=ka̰^{1}/	→	[na^4**ŋa̰^{5}**ʔɨ̃^{4}ka̰^{1}]	'so that it is raw'

b. Tone 31

Isolation /bu^{31}/ 'be immature'	/na^4=bu^{31}/	→	[na^{4}bu^{31}]	's/he is a child/immature'
With C-IV NOM ʔɨ̃^{4}	/bu^{31}-ʔɨ̃^{4}/	→	[**bu^3**ʔɨ̃^{4}]	'child'

c. Tone 3

Isolation /ã^3/ 'give'	/na^4=na^3=ã^3/	→	[na^4=na^3=ã^3]	's/he gives it'
With C-IV NOM ʔɨ̃^{4}	/wɨ^{43}ʔi^4 i^4 ã^3 ʔɨ̃^{4}/	→	[wɨ^{43}ʔi^{4}i^4**ã^2**ʔɨ̃^{4}]	'one that gives'

In (1), monosyllabic verbs /ŋa̰^{1}/ 'be raw', /bu^{31}/ 'be immature', and /ã^3/ 'give' surface with tones 5, 3, and 2 respectively when the class IV nominalizer ʔɨ̃^{4} attaches to the verb stem.

These alternations are further conditioned by stress, which itself is fixed by morphological structure. Within an inflected verb stem, multiple stress domains can obtain (see Skilton (2017, §5) for more details), but the locus of stress relevant to the tone circle is the verb root domain, where stress is assigned to the leftmost syllable. Stress assignment is relevant because stressed and unstressed syllables follow different—though not entirely disjoint—patterns. For example, the monosyllabic tone 3 root /ã^3/ 'give' maps to tone 2 (1c). This syllable occupies the leftmost position in the verb root domain, and therefore is stressed. A disyllabic root-final tone 3 will alternate in the presence of a nominalizer, but as it occupies a non-stressed position, maps to tone 1 instead, as in (2).

(2)

Isolation /ʧɨ^{4}ta^3/ 'be night'	/na^4 = ʧɨ^{4}ta^3/	→	[na^{4}ʧɨ^{4}ta^3]	'it is night'
With C-IV NOM	/ʧɨ^{4}ta^3- ʔɨ̃^{4}/	→	[ʧɨ^4**ta^1**- ʔɨ̃^{4}]	'darkness'

Example (3) generalizes the alternations for stressed and unstressed syllables. Note that unstressed syllables do not carry contour tones.[2]

[1] CT has four noun classes. Details about the noun class system are not germane to the current study, and therefore are not explored in detail here.

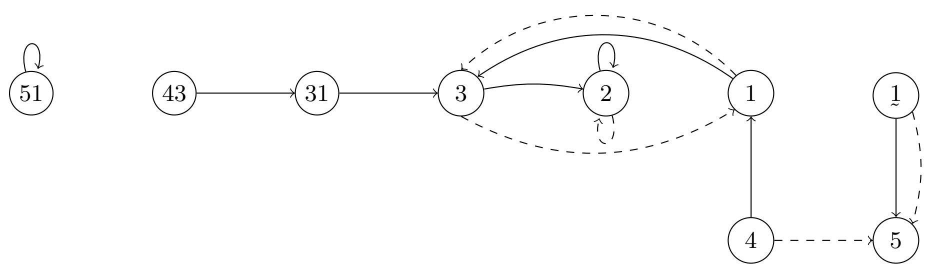
[2] These are simplified from Skilton (2017)'s field notes, specifically the variation in tone 1, which I do not explore further.

(3)

Stressed Syllables		Unstressed Syllables	
Isolation tone	Tone preceding NOM	Isolation tone	Tone preceding NOM
1̰	5	1̰	5
1	3	1	3
2	2	2	2
3	2	3	1
4	1	4	5
31	3		
43	31		
51	51		

A graphical representation of the pattern is given below in (4), where solid lines indicate the path of stressed syllables, and dashed lines indicate the path of unstressed syllables.[3]

(4)



As the graph above illustrates, the CT tone circle is less 'circular' than non-morphologically-conditioned tone circles attested in Southern Min dialects, for example Xiamen (Chen, 1987, 2000). Skilton (2017, 52) views it as "a combination of a chain shift (affecting some contours and some level tones) and two non-chain shifts." Whatever the best characterization of the full set of operations may be, there is a clear circuit between tones 3 and 1 in the unstressed syllables pattern, and as such it constitutes a circular chain shift in the sense of Anderson and Browne (1973) and Moreton (2004).

2.2 Interaction with dissimilation

Among the tonological operations reported in CT, Skilton (2017) also describes a dissimilatory process triggered when a monosyllabic tone 1̰ verb root is followed immediately by an affix/clitic of tone 1 or 1̰. In this environment, the verb root surfaces as tone 5, as in (5).

(5)

/na^4 =	ʧo̰^{1}	-a̰^{1}/	→	[na^4**ʧo^5**a̰^{1}]
3.SBJ=	be.white	-mouth		
'it has a white mouth'				

When the same verb stem (root + affix/clitic) precedes a class nominalizer, the tone on the final (unstressed) syllable of the stem shifts according to the tone circle pattern summarized in (4). This operation does not, however, affect the application of dissimilation. In (6), the affix /- a^1/ '-mouth' maps to tone 5 in normal tone circle fashion, yet dissimilation in the root /ʧo̰^{1}/ 'be white' is unimpeded.

(6)

/na^4 =	ʧo̰^{1}	-a̰^{1}	- ʔɨ̰̃^{4}	=ka̰^{1}/	→	[na^4**ʧo^{5}o^5**ʔɨ̰̃^{4}ka̰^{1}]
3.SBJ=	be.white	-mouth	-NOM	=PURP		
'so that it has a white mouth'						

This interaction is opaque. The tone circle operation counterbleeds dissimilation, appearing to have overapplied in environments where dissimilation also occurs. In a rule-based account, this would require the rules governing the circular shift to be ordered after dissimilation. If the order were reversed, the application of the nominalizer circle on /1̰-1̰-NOM/ → 1̰-5-NOM would bleed the /1̰-1̰/ environment that triggers dissimilation.

[3]Tones 5 and 41 do not participate in the CT nominalizer tone circle pattern for phonotactic reasons. The current study does not address this issue in detail, but see §3.2.

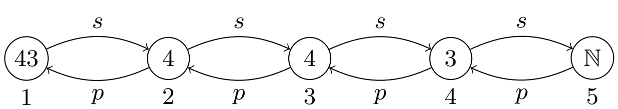
3 Computational account

The computational account pursued here addresses two questions: 1) what is the nature of the function that relates inputs to outputs in a 'circular' fashion, as in CT?; and 2) what is the nature of the function that describes the CT tone circle's interaction with dissimilation? Framing the problem in this way circumvents questions of how such patterns are operationalized in a particular framework, whether rule-based or constraint-based (Chandlee and Heinz, 2018; Heinz, 2018). This also means that any insights drawn from the analysis speak directly to the input-output mapping (i.e. the function) itself, not the set of rewrite rules or ranking of constraints which describe it. The account of CT aligns with the subregular hypothesis for phonology (Heinz and Lai, 2013), whose central claim is that phonological transformations fall within a class of subregular functions, a subset of the regular relations. Before proceeding to the main analyses (§3.2 and §3.3), the following section introduces the adopted computational framework (§3.1.1), the computational notion of input strict-locality (§3.1.2), and the BMRS formalism within which the analysis is couched (§3.1.3).

3.1 Preliminaries: string models and subregularity

Input-output mappings may be formalized using logical *transduction*, or a set of logical statements that define output structure in terms of the input (Courcelle, 1994; Engelfriet and Hoogeboom, 2001). Recent work within the subregular paradigm has leveraged this formalism to make generalizations about the complexity of phonological processes using various representational schema, including autosegmental representations (Chandlee and Jardine, 2019), syllabic structure (Strother-Garcia, 2018), and feature-geometric models of tone (Oakden, 2020). This paper formalizes tonal operations in CT as transductions, specifically over hybrid word models using model theory (Büchi, 1960; Enderton, 2001). These models comprise a set of relations and functions which are used to describe relational structures. The input structures being manipulated are strings containing tonal and morphological information relevant to the tone circle pattern. Each position in the string is labeled—via a set of relations—with a number denoting Skilton (2017)'s tonal notation or a symbol denoting morphological information such as the class nominalizer ('ℕ' below). Predecessor p and successor s functions establish a linear order over these elements by identifying the node that immediately precedes and follows a given structural position. In (7), a hybrid word model representation is given for the input form of (1c) /wɨ⁴³ʔi⁴ i⁴ ã³ ʔɨ̃⁴/ 'one that gives'. Each structural position is labeled with an index.

(7)



By representing processes as input-output mappings in this way, it is possible to characterize (morpho-)phonological transformations in terms of their computational complexity. This perspective has been leveraged to make statements about phonological typology. That is, it can distinguish the set of attested phonological maps from the set of unattested (but logically possible) ones. An early result in this line of inquiry is that phonological generalizations are regular (Johnson, 1972; Kaplan and Kay, 1994), meaning that they can be computed using fixed memory. A further restriction on this generalization has gained traction in recent years, namely that phonological transformations are *sub*-regular, describable by a class of functions known as the subsequential class (Mohri, 1997). This subregular hypothesis (Heinz and Lai, 2013; Heinz, 2018) holds that phonological transformations are those which compute outputs using bounded lookahead. Using this upper bound on complexity, it is possible to distinguish, for example, attested spreading processes (such those found in vowel harmony) from unattested ones like majority rules (Baković, 2000) and sour grapes (Wilson, 2003, 2006).

3.1.1 Input strict-locality

One of the most restrictive class of subregular functions is the *input strictly-local* class (ISL, Chandlee, 2014). Functions of this class, simply put, compute outputs using only local, bounded reference to input structure. To provide an example from segmental phonology, consider a nasal place assimilation process whereby an input /tana/ maps to an output [tana], /tanpa/ to [tampa], and /tanka/ to [taŋka], etc. A digram in (8) shows the mapping /tanpa/ ↦ [tampa].

(8)

input:	t	a	n	p	a
output:	t	a	**m**	p	a

Whether input /n/ maps to [n], [m] or [ŋ] (the gray cell) can be determined solely from a bounded window of information in the input (the pink cells). In this case, the size of the window is two segments: the current segment and one segment to the right. Thus we may say that one property of local nasal place assimilation is input strict-locality. An ISL-2 function—indicating a window of size 2—is sufficient to describe the set of maps /tana/ ↦ [tana], /tanpa/ ↦ [tampa], /tanka/ ↦ [taŋka], etc.

ISL functions are sufficient to model a wide range of segmental and autosegmental phonological processes despite their restrictiveness (Chandlee, 2014; Chandlee and Jardine, 2019). Maps describable by this class encompass those corresponding to individual processes, as well as multiple phenomena applying to the same input structure (Chandlee and Heinz, 2018). Chandlee et al. (2018) show that a number of opaque maps are also ISL, including linear chain shifts such as Danish vowel lowering (Hyman, 1975; Lundskær-Nielsen and Holmes, 2011). A central claim of this paper is that CT nominalizer tone circles—a type of circular chain shift—and the opaque interaction with dissimilation also exhibit the property of input strict-locality.

3.1.2 BMRS

ISL functions may be described using various formalisms, including automata-theoretic (Chandlee, 2014) and logical (Lindell and Chandlee, 2016) characterizations. This paper adopts *boolean monadic recursive schemes* (BMRS; Bhaskar et al., 2020) as a means to represent ISL functions. BMRS have their basis in recursive program schemes, which are used to study the complexity of algorithms (Moschovakis, 2019). In terms of expressivity, schemes of this type have been shown to describe exactly the subsequential class of functions (Bhaskar et al., 2020).[4] As such, they align with the upper bound on complexity put forth by the subregular hypothesis for phonology. BMRS also has the advantage of intensionally expressing phonologically-significant generalizations more directly than previous computational formalisms and implementing phonological substance, all while maintaining the important insights about complexity (Chandlee and Jardine, 2020, see also (Pater, 2018) for relevant criticisms of the computational perspective).

At its core, a BMRS analysis comprises a set of predicates which determine the computation of output structure. A simple 'if...then...else' syntax forms the basis of how these predicates are evaluated, with predicates returning either a 'true' or 'false' value (hence *boolean*). To illustrate, recall the nasal place assimilation mapping in (8). The generalization we aim to capture from this example is: an m in the output maps from an input n when an input p directly follows it. In terms of a BMRS predicate, this computation can be represented as in (9), where x denotes a segment in the input structure, and $\top$ means 'true'.

(9) $m_o(x) \quad = \quad \texttt{if } \underline{n}p(x) \texttt{ then } \top \texttt{ else } m(x)$

This equation describes under what conditions a segment is an output m, where subscripted '$_o$' refers to 'output'. Specifically, if the input segment under evaluation—a single free variable x, hence *monadic*—is an n and is followed immediately followed by an input p, then $m_o(x)$ evaluates to $\top$ 'true', and that segment is output as m. This is represented above as the *expression*[5] $\underline{n}p(x)$ where underlining indicates the current input position being evaluated. If x is not such a structure, however, then evaluation proceeds to the expression after 'else'. In the case of (9), a truth value is returned based on whether x is an input m (represented as $m(x)$).

Two observations about the equation are worth noting. First, the structures $\underline{n}p(x)$ and $m(x)$ are both input-oriented; they refer to a bounded window in the input structure. This means that the determination of how output m is computed can be made solely from a local window of information from the input—i.e. it has the property of input strict-locality. BMRS predicates can be defined *recursively* such that expressions on the right-hand side of the equation contain output predicates such as $m_o(x)$. Such equations can be considered output-oriented; they compute output structure in terms of the output. BMRS describing ISL functions are therefore a special case of BMRS *without* any recursion.

[4] This hinges on a crucial limitation of recursion in only one direction. See (Bhaskar et al., 2020) for more details.

[5] BMRS expressions are any of the following: boolean values $\top$ and $\bot$ (see below); predicates $P(t)$ for any symbol in the input alphabet, and expressions of the form $\texttt{if } E_1 \texttt{ then } E_2 \texttt{ else } E_3$ where E_{1-3} are expressions.

Second, the expression $\underline{np}(x)$ can be understood as a *licensing* structure, because it licenses an output m on some segment x. Any expression of the form '`if` *struc* `then` $\top$', where *struc* refers to some local structure, is a licensing structure in the BMRS formalism.

The equation in (9) only describes half of the assimilation generalization, however. The other crucial portion is that input n *cannot* map to output n when followed by input p. This may be expressed as a BMRS predicate describing conditions on output n segments, i.e. $n_o(x)$. This is given in (10), where $\bot$ indicates 'false'.

(10) $n_o(x) \quad = \quad$ `if` $\underline{np}(x)$ `then` $\bot$ `else` $n(x)$

Thus in the same way $\underline{np}(x)$ licenses output m, it *blocks* output n by causing $n_o(x)$ to evaluate to $\bot$ false, constituting a blocking structure. More generally, any expression of the type '`if` *struc* `then` $\bot$' can be understand as a blocking structure.

Taken together, $m_o(x)$ and $n_o(x)$ form a *system* of BMRS equations. This system describes the mapping in (8).[6] An evaluation table for the n/m alternation in /tanpa/ $\mapsto$ [tampa] (11) illustrates; the third input position (as labeled by the numbers in the top row) evaluates to $\top$ for $m_o(x)$ and $\bot$ for $n_o(x)$.

(11)

Input:	t	a	n	p	a
	1	2	3	4	5
m_o(x)			$\top$		
n_o(x)			$\bot$		
Output:			**m**		

Throughout the paper, such tables will be used to demonstrate evaluations of mappings against BMRS systems.

An alternative formalization of the BMRS system described above might be an SPE rewrite rule $n \rightarrow m$ / __ p. The two are related only to the extent that they describe the same mapping from inputs to outputs: {/tanpa/ $\mapsto$ [tampa], /panpa/ $\mapsto$ [pampa], /nama/ $\mapsto$ [nama], /pa/ $\mapsto$ [pa]...}. In other words, this does not indicate that BMRS is merely a notation for describing rules. As Chandlee and Jardine (2020) show, a single BMRS system can capture the interaction of multiple generalizations, unlike single rules. They also note that local blocking structures like $\underline{np}(x)$ perform a similar function to markedness constraints in OT in that they forbid structures from appearing in the output. Likewise, expressions like $n(x)$ in the definition of $n_o(x)$ function like faithfulness constraints (i.e. it evaluates for direct mapping from the input). Thus `if` $\underline{np}(x)$ `then` $\bot$ `else` $n(x)$ can also be understood as resembling a ranking $*np \gg$ IDENT(n) since $\underline{np}(x)$ is evaluated first in the definition $n_o(x)$. More generally, licensing and blocking structures arrange into *hierarchies* within individual predicates, and in a way that is similar to rankings of constraints in OT grammars. §5 explores this issue in more detail, but I first present a BMRS analysis of the CT tone circle (§3.2) and its interaction with dissimilation (§3.3).

3.2 Tone circle is ISL

The CT nominalizer tone circle exhibits the property of input strict-locality. That is, it describes an ISL function. Grammatically-conditioned tonal shifts can be computed with local reference to input structure, provided that the computation has access to relevant morphological—as well as tonal—information in the input. Such a requirement is expected of a grammatical tone pattern. Tone circles are formalized over hybrid word models containing lexical tones, a symbol denoting the class nominalizer ('$\mathbb{N}$'), and root stress domain boundaries (represented as '$\#_r$'). Enriching the representation beyond tonal information is necessary to capture the grammatical nature of these patterns. Specifically, a symbol for the nominalizer is motivated as it is the main trigger of the pattern itself. Motivation for the boundary symbol stems from the fact that underlying tones map to different surface tones depending on whether they appear in stressed or unstressed syllables (4). Since stress aligns to the left edge of the verbal root domain, having $\mathbb{N}$ as an immediate predecessor reliably distinguishes stressed syllables from unstressed syllables using bound reference to input structure only.

To model the tone circle using BMRS, an output boolean function is defined for each surface tone, keeping in mind that not every lexical CT tone is attested as an output of this process (as in (4)). An

[6] To describe the full transformation including non-alternating segments, predicates $t_o(x)$, $p_o(x)$, and $a_o(x)$ would also be included. Since they map directly from corresponding inputs, I refrain from discussing them here.

auxiliary predicate $\sigma(x)$ is first defined in (12) to isolate syllabic tonal symbols in the set of input labels, i.e. to exclude $\#_r$ and $\mathbb{N}$.

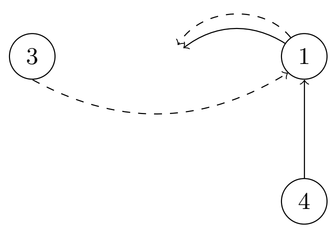
(12) $$\sigma(x) \;=\; \texttt{if } \#_r(x) \texttt{ then } \bot \texttt{ else} \\ \texttt{if } \mathbb{N}(x) \texttt{ then } \bot \texttt{ else } \top$$

Predicate $\sigma(x)$ is used as a shorthand in characterizing relevant input structures. For example, the expression $\sigma\underline{3}(x)$ denotes a structure containing a tone 3 syllable immediately preceded by any other lexical tone, but crucially not a class nominalizer or root boundary. Thus it subsumes $\{1\underline{3}(x)$, $\underset{\sim}{1}\underline{3}(x)$, $2\underline{3}(x)...\}$ under a single definition. It also does so using a bounded window of information from the input structure, so does not increase the complexity beyond ISL of the BMRS systems that refer to it. An equation $1_o(x)$ (for output tone 1) using this definition is given in (13).

(13) $$1_o(x) \;=\; \texttt{if } \#_r\underline{4}\mathbb{N}(x) \texttt{ then } \top \texttt{ else} \\ \texttt{if } \sigma\underline{3}\mathbb{N}(x) \texttt{ then } \top \texttt{ else} \\ \texttt{if } \underline{1}\mathbb{N}(x) \texttt{ then } \bot \texttt{ else } 1(x)$$

This definition states in the first line that a syllable will be output as tone 1 if it is an input 4 tone with a root domain edge to its immediate left (indicating a stressed syllable) and a nominalizer to its immediate right. In other words, the structure $\#_r\underline{4}\mathbb{N}(x)$ licenses tone 1 in the output. Output tone 1 is also licensed when an input 3 tone is flanked by any lexical input tone on the left (indicating an unstressed syllable) and the nominalizer on the right, as in the second line. If an input 1 tone is followed by a nominalizer, represented as $\underline{1}\mathbb{N}(x)$ in the third line, it will not be output as tone 1. This is a blocking structure on output tone 1. Otherwise, syllables specified as 1 in the input map to 1 in the output ($1(x)$). The BMRS definition $1_o(x)$ thus models mappings to (and from) tone 1 in (4), illustrated below in (14).

(14)



Following in this manner, the remaining output lexical tones surfacing in the circular pattern can be defined: 2, 3, 5, 31, 51. All six output tonal definitions comprise the full BMRS system in (15).

(15)

$$1_o(x) \;=\; \texttt{if } \#_r\underline{4}\mathbb{N}(x) \texttt{ then } \top \texttt{ else} \\ \texttt{if } \sigma\underline{3}\mathbb{N}(x) \texttt{ then } \top \texttt{ else} \\ \texttt{if } \underline{1}\mathbb{N}(x) \texttt{ then } \bot \texttt{ else } 1(x)$$

$$2_o(x) \;=\; \texttt{if } \#_r\underline{3}\mathbb{N}(x) \texttt{ then } \top \texttt{ else } 2(x)$$

$$3_o(x) \;=\; \texttt{if } \underline{1}\mathbb{N}(x) \texttt{ then } \top \texttt{ else} \\ \texttt{if } \#_r\underline{31}\mathbb{N}(x) \texttt{ then } \top \texttt{ else} \\ \texttt{if } \underline{3}\mathbb{N}(x) \texttt{ then } \bot \texttt{ else } 3(x)$$

$$5_o(x) \;=\; \texttt{if } \underset{\sim}{\underline{1}}\mathbb{N}(x) \texttt{ then } \top \texttt{ else} \\ \texttt{if } \sigma\underline{4}\mathbb{N}(x) \texttt{ then } \top \texttt{ else } 5(x)$$

$$31_o(x) \;=\; \texttt{if } \#_r\underline{43}\mathbb{N}(x) \texttt{ then } \top \texttt{ else} \\ \texttt{if } \#_r\underline{31}\mathbb{N}(x) \texttt{ then } \bot \texttt{ else } 31(x)$$

$$51_o(x) \;=\; 51(x)$$

A maximum of one input symbol to the left and to the right of the current string position under evaluation is sufficient to compute the circular tone shift as indicated by the definitions above. Thus the CT nominalizer tone circle is describable by an ISL-3 function, underlying its input-oriented nature. Below, evaluations of two mappings modeling the forms /bu^{31}-ʔɨ̃^{4}/ ‘child’ and /ʧɨ^{4}ta^{3}- ʔɨ̃^{4}/ ‘darkness’ illustrate. Inputs are represented in hybrid representation as $\#_r$.31.$\mathbb{N}$ and $\#_r$.4.3.$\mathbb{N}$ respectively.

(16)

Input:	$\#_r$	31	$\mathbb{N}$	$\#_r$	4	3	$\mathbb{N}$
	1	2	3	1	2	3	4
$31'(x)$	$\bot$	$\bot$	$\bot$	$\bot$	$\bot$	$\bot$	$\bot$
$3'(x)$	$\bot$	$\top$	$\bot$	$\bot$	$\bot$	$\bot$	$\bot$
$1'(x)$	$\bot$	$\bot$	$\bot$	$\bot$	$\bot$	$\top$	$\bot$
$\#_r\underline{31}\mathbb{N}(x)$	$\bot$	$\top$	$\bot$	$\bot$	$\bot$	$\bot$	$\bot$
$\underline{3}\mathbb{N}(x)$	$\bot$	$\bot$	$\bot$	$\bot$	$\bot$	$\top$	$\bot$
$\sigma\underline{3}\mathbb{N}(x)$	$\bot$	$\bot$	$\bot$	$\bot$	$\bot$	$\top$	$\bot$
Output:		**3**				**1**	

On the input string $\#_r.31.\mathbb{N}$, the 31 tone conforms to the structure $\#_r\underline{31}\mathbb{N}(x)$. This structure licenses output tone 3 and blocks output tone 31, as per definitions of $3_o(x)$ and $31_o(x)$ respectively. It surfaces as tone 3 in the output, consistent with the attested surface form [**bu**$^{\mathbf{3}}$ʔɨ̃^{4}] ‘child’. Likewise, input tone 3 in the string $\#_r.4.3.\mathbb{N}$ returns a true value for structures $\underline{3}\mathbb{N}(x)$ (blocking output tone 3) and $\sigma\underline{3}\mathbb{N}(x)$ (licensing output tone 1). The predicted output tone 1 is also consistent with the observed [ʧɨ^4**ta**$^{\mathbf{1}}$- ʔɨ̃^{4}] ‘darkness’.

3.3 Interaction with dissimilation is ISL

The tone circle’s interaction with $\utilde{1}$-dissimilation is also ISL. This section first describes the dissimilatory pattern as an ISL-definable BMRS system of equations, then appends the definition in (15) to reflect the opaque interaction.

To first describe $\utilde{1}$-assimilation, another shorthand predicate is introduced in (17). Its purpose is to isolate syllabic tone symbols of either tone 1 or $\utilde{1}$. Like the predicate $\sigma(x)$ defined in (12), it is input-oriented, and is composed of well-formed BMRS expressions.

(17)
$$\begin{array}{lll} \sigma_{\{1,\utilde{1}\}}(x) & = & \texttt{if } \sigma(x) \texttt{ then} \\ & & \quad \texttt{if } 1(x) \texttt{ then } \top \texttt{ else} \\ & & \quad \texttt{if } \utilde{1}(x) \texttt{ then } \top \texttt{ else } \bot \\ & & \texttt{else } \bot \end{array}$$

Using this predicate, the conditioning environment of dissimilation can be characterized within an input structural window of size three: a verbal root domain boundary ($\#_r$) followed by an input $\utilde{1}$-toned syllable—thus describing a monosyllabic root—followed by another syllable of tone 1 or $\utilde{1}$, condensed into $\sigma_{\{1,\utilde{1}\}}(x)$. This structure is distributed across two output tonal definitions $5_o(x)$ and $\utilde{1}_o(x)$, licensing the former in the output and blocking the latter. A partial system for the dissimilatory pattern is given in (18), where the remaining lexical tones are understood to map directly from corresponding inputs. Note that, as with the circular tonal shift, this process is described by an ISL-3 function.

(18)
$$\begin{array}{lll} 5_o(x) & = & \texttt{if } \#_r\underline{\utilde{1}}\sigma_{\{1,\utilde{1}\}}(x) \texttt{ then } \top \texttt{ else } 5(x) \\ \utilde{1}_o(x) & = & \texttt{if } \#_r\underline{\utilde{1}}\sigma_{\{1,\utilde{1}\}}(x) \texttt{ then } \bot \texttt{ else } \utilde{1}(x) \end{array}$$

The function maps tone $\utilde{1}$ verbal roots to tone 5 when followed by a 1- or $\utilde{1}$-toned syllable. An evaluation table in (19) modeling the form /na^4=ʧo̰^{1}-a̰^{1}/ ‘it has a white mouth’ (‘$4.\#_r.\utilde{1}.\utilde{1}$’ in hybrid representation) illustrates. The mapping $/4.\#_r.\utilde{1}.\utilde{1}/ \mapsto [4.\#_r.\mathbf{5}.\utilde{1}]$ evaluates to true via the definitions in (18), and the output accords with the attested surface form [na^4**ʧo**$^{\mathbf{5}}$a̰^{1}] (5).

(19)

Input:	4	$\#_r$	$\utilde{1}$	$\utilde{1}$
	1	2	3	4
$5_o(x)$	$\bot$	$\bot$	$\top$	$\bot$
$\utilde{1}_o(x)$	$\bot$	$\bot$	$\bot$	$\top$
$\#_r\underline{\utilde{1}}\sigma_{\{1,\utilde{1}\}}(x)$	$\bot$	$\bot$	$\top$	$\bot$
$\utilde{1}(x)$	$\bot$	$\bot$	$\top$	$\top$
Output:			**5**	$\utilde{1}$

Specifically, string position 3 evaluates to true for the structure $\#_r\underline{\utilde{1}}\sigma_{\{1,\utilde{1}\}}(x)$, licensing tone 5—and blocking tone $\utilde{1}$—on that position in the output. Position 4, also an input $\utilde{1}$ tone, does not satisfy this structure and maps directly from the input by virtue of evaluating to true for $\utilde{1}(x)$ in the definition $\utilde{1}_o(x)$.

Both processes in isolation can be modeled as BMRS systems describing ISL-3 functions. A third system combines these definitions into a single map representing the interaction of the tone circle with $\utilde{1}$-dissimilation. This function is also ISL-3, indicating the input-oriented nature of the counterbleeding interaction.

Consider first an appended definition of $5_o(x)$ from (15), combining licensing/blocking structures from both the tone circle mapping and the dissimilatory pattern, as in (20).

(20)
$$\begin{array}{lll} 5_o(x) & = & \texttt{if } \underline{\utilde{1}}\mathbb{N}(x) \quad\quad \texttt{ then } \top \texttt{ else} \\ & & \texttt{if } \sigma\underline{4}\mathbb{N}(x) \quad\quad \texttt{ then } \top \texttt{ else} \\ & & \texttt{if } \#_r\underline{\utilde{1}}\sigma_{\{1,\utilde{1}\}}(x) \texttt{ then } \top \texttt{ else } 5(x) \end{array}$$

Crucially, the dissimilation trigger *and* the tone circle trigger, both of which result in output tone 5, refer to the same input string. An interaction occurs when these triggers converge on overlapping input positions—e.g. the hybrid string /4.$\#_r$.$\utilde{1}$.$\utilde{1}$.$\mathbb{N}$.1/ corresponding to the form /na^4=tʃo̰^{1}-a̰^{1}-ʔĩ^4=ka̰^{1}/ 'so that it has a white mouth'. In the evaluation table in (21), two input $\utilde{1}$-tone syllables map to tone 5 in the output, illustrating the counterbleeding interaction.

(21)

Input:	4	$\#_r$	$\utilde{1}$	$\utilde{1}$	$\mathbb{N}$	1
	1	2	3	4	5	6
$5_o(x)$	$\bot$	$\bot$	$\top$	$\top$	$\bot$	$\bot$
$\#_r\utilde{\underline{1}}\sigma_{\{1,\utilde{1}\}}(x)$	$\bot$	$\bot$	$\top$	$\bot$	$\bot$	$\bot$
$\utilde{\underline{1}}\mathbb{N}(x)$	$\bot$	$\bot$	$\bot$	$\top$	$\bot$	$\bot$
Output:			**5**	**5**		

The third string position outputs tone 5 by satisfying the dissimilation structure $\#_r\utilde{\underline{1}}\sigma_{\{1,\utilde{1}\}}(x)$. In this way it is identical to the third position in the string in (19). String position 4 also maps to tone 5, but instead does so by satisfying $\utilde{\underline{1}}\mathbb{N}(x)$, the relevant tone circle structure. The input window of each structure is of size 3, meaning that the triggers of string-adjacent targets of both processes overlap.

Appended $5_o(x)$, along with $\utilde{1}_o(x)$ from (18), joins the tone circle system in (15) to form a single system modeling the interaction map, as in (22).

(22)

$$\begin{array}{rcl}
1_o(x) & = & \texttt{if}\ \#_r\underline{4}\mathbb{N}(x)\ \texttt{then}\ \top\ \texttt{else} \\
 & & \texttt{if}\ \sigma\underline{3}\mathbb{N}(x)\ \texttt{then}\ \top\ \texttt{else} \\
 & & \texttt{if}\ \underline{1}\mathbb{N}(x)\ \texttt{then}\ \bot\ \texttt{else}\ 1(x) \\
2_o(x) & = & \texttt{if}\ \#_r\underline{3}\mathbb{N}(x)\ \texttt{then}\ \top\ \texttt{else}\ 2(x) \\
3_o(x) & = & \texttt{if}\ \underline{1}\mathbb{N}(x)\ \texttt{then}\ \top\ \texttt{else} \\
 & & \texttt{if}\ \#_r\underline{31}\mathbb{N}(x)\ \texttt{then}\ \top\ \texttt{else} \\
 & & \texttt{if}\ \underline{3}\mathbb{N}(x)\ \texttt{then}\ \bot\ \texttt{else}\ 3(x) \\
\utilde{1}_o(x) & = & \texttt{if}\ \#_r\utilde{\underline{1}}\sigma_{\{1,\utilde{1}\}}(x)\ \texttt{then}\ \bot\ \texttt{else}\ \utilde{1}(x) \\
5_o(x) & = & \texttt{if}\ \utilde{\underline{1}}\mathbb{N}(x)\ \texttt{then}\ \top\ \texttt{else} \\
 & & \texttt{if}\ \sigma\underline{4}\mathbb{N}(x)\ \texttt{then}\ \top\ \texttt{else} \\
 & & \texttt{if}\ \#_r\utilde{\underline{1}}\sigma_{\{1,\utilde{1}\}}(x)\ \texttt{then}\ \top\ \texttt{else}\ 5(x) \\
31_o(x) & = & \texttt{if}\ \#_r\underline{43}\mathbb{N}(x)\ \texttt{then}\ \top\ \texttt{else} \\
 & & \texttt{if}\ \#_r\underline{31}\mathbb{N}(x)\ \texttt{then}\ \bot\ \texttt{else}\ 31(x) \\
51_o(x) & = & 51(x)
\end{array}$$

All expressions in the definition refer to input structures with a maximum length of 3. The interaction map thus describes an ISL-3 function. Importantly, dissimilation and the circular shift interact in what is described as a counterbleeding order. As the system above illustrates, this effect can be attributed to input-orientedness; as the tone circle and dissimilation are sensitive to the same input structure, they are 'blind' to the modifications made to the input string by the other process.

4 Discussion

As the preceding sections demonstrate, grammatical tone necessitates a framework that allows reference to input structure. The CT tone circle and its interaction with dissimilation align with a broad set of attested phonological processes in that they exhibit the computational property of input strict-locality. This section compares the BMRS analysis pursued in this paper with alternative analyses of the CT data and tone circles more broadly, including rule- and constraint-based approaches. It also comments on ramifications for our understanding of the BMRS framework, as well as the relationship between GT and non-grammatically-conditioned tonal and segmental processes.

Skilton (2017) proposes a 'perturbation' account of the CT tone circle, borrowing from literature on Oto-Manguean tone systems (Mak, 1953). Under this analysis, each level and contour CT tone has two allotones: an isolation allotone and a perturbation allotone. The latter surfaces in environments causing perturbation, such as when it precedes a class nominalizer or in the environment for dissimilation. Skilton draws a parallel between the isolation/perturbation form relationship and segments undergoing consonant mutation in Modern Irish (Ní Chiosáin, 1991; Green, 2006); neither generalization can be expressed as a single phonological rule, and the relationship between one pair of variants is uninformative regarding how other pairs relate. Mortensen (2006, 90-93, 106-108), for example, has criticized similar arguments about the non-grammatically-conditioned Xiamen tone circle (Ballard, 1988; Tsay and Myers, 1996) which aim to push the computation outside of phonology proper, presenting a phonological 'null hypothesis' in the sense of Kenstowicz and Kisseberth (1979). One important objection to allomorph selection hypotheses is that they are insufficiently restrictive in the types of alternations they predict.

The BMRS analysis pursued here, by contrast, identifies a restrictive computational property—input strict-locality—of the CT tone circle maps, placing its complexity well within established bounds for phonology. It provides a phonological explanation of the data, as ISL-ness is an established property of phonological maps. In doing so, it is important to note that the intent is not a demarcation between phonology and morphology, or to claim that the phonological explanation of CT tone circles precludes a morphological explanation. In fact, input strict-locality has been shown to be a property of morphological processes as well (Chandlee, 2017; Dolatian and Rawski, 2020). What the analysis instead shows is that the computational power necessary to capture a wide range of phonological patterns is also sufficient for the CT data. More generally, the ISL bound on complexity makes predictions about possible and impossible patterns which can later be falsified as more data are introduced. It therefore provides a rigorous point of departure for further inquiry into the computational nature of grammatical tone.

Within the BMRS formalism, input strict-locality is reflected in the set of input-oriented structures which license some outputs and block others. Access to this type of reference (beyond simply ensuring faithful input-output mappings) is an advantage of this approach, and is unavailable to output-based models like OT, where markedness constraints target surface structures only.

Access to input structures also permits a straightforward account of circular chain shifts in general, not just CT. Mappings of this nature pose fundamental difficulties to both rule- and optimization-based formalisms. Consider the closed circuit between tones 1 and 3 for unstressed syllables in the CT tone circle; tone 1 maps to tone 3, and tone 3 maps to tone 1. In an SPE-style analysis, linear chain shifts reflect counterfeeding orders (Kisseberth and Kenstowicz, 1977), but a circular chain shift cannot be derived by any order over a set of individual rewrite rules, because an unintended feeding relationship is inevitable at some point in the derivation. This gives rise to an ordering paradox. To illustrate with a simplified example, derivations of unstressed syllables (indicated by 'ˌ') are given in (23) using both orders of rewrite rules.[7] Each produces an unwanted Duke of York effect.

(23) a.

	/ˌ1ℕ/	/ˌ3ℕ/
1 → 3 / __]ˌℕ	ˌ3ℕ	—
3 → 1 / __]ˌℕ	ˌ1ℕ	ˌ1ℕ
	*[ˌ1ℕ]	[ˌ1ℕ]

b.

	/ˌ1ℕ/	/ˌ3ℕ/
3 → 1 / __]ˌℕ	—	ˌ1ℕ
1 → 3 / __]ˌℕ	ˌ3ℕ	ˌ3ℕ
	[ˌ3ℕ]	*[ˌ3ℕ]

Circular chain shifts are equally problematic for parallel OT. Like linear chain shifts, these maps are non-output-driven (Tesar, 2014), and are non-idempotent (Magri, 2018), that is, they do not constitute maps for which phonotactically-licit forms map faithfully to themselves. Moreton (2004) proves that circular mappings are uncomputable by conservative OT grammars—i.e. grammars consisting of only markedness and faithfulness constraints. This makes them more daunting than linear chain shifts, which Kirchner (1996) demonstrates can be captured using conservative OT grammars. To illustrate with the same example as above, the mapping /1/ → [3] implies that surface form [3] is less marked than the fully-faithful candidate [1]. But the other mapping /3/ → [1] also implies that surface [1] is less marked then fully-faithful [3]. This leads to a contradiction whereby [3] must be less marked than itself.[8] No ranking of constraints can produce this type of effect.

The current analysis instead identifies input strict-locality as a basic quality of grammatically-conditioned circular chain shifts. In principle, this can be extended to non-GT circular chain shifts, as well. The general intuition is as follows. Information available to the computation is confined to a single input string, and crucially not intermediate representations generated by pairwise rule ordering, thus circumventing the Duke of York effects. Additionally, input-oriented licensing and blocking structures determine well-formed outputs based on 1) which inputs they map from, and 2) the local environments of those inputs. This is a key component missing from OT constraints that only target marked surface structures.

Relatedly, while no hierarchy of classic OT constraints produces the circular shift in CT, (almost) any hierarchy of licensing and blocking structures within the BMRS definitions in (22) captures the pattern.

[7] While I assume that stress is assigned elsewhere in the computation, the issue of stress assignment is not central to the ordering paradox.

[8] A possible rejoinder is that CT is conditioned by non-homogeneous representational factors (as Moreton claims for Xiamen in his original argument), and therefore is not a proper test case. I do not explore this issue further, as the main focus is on the computation of circular chain shifts and not their representation.

For example, consider an alternate definition of $3_o(x)$ in (24); the order of licensing and blocking structures has been rearranged.

(24) $$3_o(x) = \begin{array}{lll} \texttt{if } \underline{3}\mathbb{N}(x) & \texttt{then } \bot \texttt{ else} & \\ \texttt{if } \#_r\underline{31}\mathbb{N}(x) & \texttt{then } \top \texttt{ else} & \\ \texttt{if } \underline{1}\mathbb{N}(x) & \texttt{then } \top \texttt{ else} & 3(x) \end{array}$$

Whereas in (22) the licensing structures rank above the blocking structure in the hierarchy, in (24) the order is reversed. This does not, however, change the input-output mapping being described. Definition (24) accepts CT tone circle mappings in an identical manner as the original definition. Indeed, any permutation of these structures accepts the mappings because, when integrated into the larger system, it describes the same function. This result also sheds light on the analogy between hierarchies of licensing/blocking structures in the BMRS formalism and constraint hierarchies in OT (Chandlee and Jardine, 2020). The three structures in the definition of $3_o(x)$ being manipulated are completely independent of one another—no one structure is a subset of the other. In a sense they are akin to OT constraints within a hierarchy for which no critical ranking is motivated. Future work can explore the formal nature of this analogy in more detail, but I do not pursue it further here. What matters is that these structures, unranked with respect to one another, all outrank the relevant 'faithfulness' constraint: $3(x)$. Ranking this at the top of the hierarchy would, like an analogous ranking of constraints in an OT grammar, predict a faithful mapping.

This generalization extends to the portion of the system governing the opaque interaction, as well. Permuting the order of licensing structures in the definition of $5_o(x)$ as in (25) does not alter the predictions of the system, recalling that a tone surfaces as tone 5 either as a result of its participation in the tone circle or as a result of tonal dissimilation.

(25) $$5_o(x) = \begin{array}{lll} \texttt{if } \#_r\underline{\underset{\sim}{1}}\sigma_{\{1,\underset{\sim}{1}\}}(x) & \texttt{then } \top \texttt{ else} & \\ \texttt{if } \sigma\underline{4}\mathbb{N}(x) & \texttt{then } \top \texttt{ else} & \\ \texttt{if } \underline{\underset{\sim}{1}}\mathbb{N}(x) & \texttt{then } \top \texttt{ else} & 5(x) \end{array}$$

The definition above differs from (22) in that the licensing structure for $\underline{\underset{\sim}{1}}$-dissimilation is situated at the top of the hierarchy instead of the bottom. As before, there is no overlap in the input-oriented licensing structures and therefore their respective order in the hierarchy is inconsequential. More importantly, though, this type of permutation does not change the complexity of the function; the interaction is ISL. Thus the opaque interaction of grammatically-conditioned tonal operations in CT aligns with Chandlee et al. (2018)'s result from segmental phonology that some opaque maps are ISL. Included among these is a counterbleeding interaction of vowel lowering and shortening in Yowlumne (McCarthy, 1999) not unlike the CT counterbleeding interaction described here. Thus ISL-ness is a property of both grammatically-conditioned and non-grammatically-conditioned opacity. Subsequent studies can seek out opaque GT interactions attested in other languages, compare them to the typology of opaque interactions in (non-GT) segmental/tonal phonology (e.g. Baković, 2007), and determine whether they exhibit the property of input strict-locality, as well.

5 Conclusion

This paper has offered an initial exploration into the computational properties of grammatical tone. In particular, it identifies input strict-locality as a characteristic of GT maps, a property also shared by non-grammatically-determined tonal and segmental processes in phonology. An analysis of a nominalizer tone circle in Ticuna and opaque interaction with dissimilation demonstrate that GT necessitates a framework that allows explicit reference to input structure. The computational analysis presented in this paper, couched within the BMRS formalism, highlights the input-orientedness of the CT tonal operations. Bounded reference to tonal and morphological information in the input is sufficient to capture the CT paradigm. This is an improvement on analyses using rule-based and constraint-based theories of phonology, which struggle to account for circular chain shifts.

References

Anderson, D. (1962). *Conversational Ticuna*. SIL, Yarinacocha, Peru.

Anderson, S. R. and Browne, W. (1973). On keeping exchange rules in Czech. *Research on Language & Social Interaction*, 6:445–482.

Baković, E. (2000). *Harmony, dominance and control*. PhD thesis, Rutgers University.

Baković, E. (2007). A revised typology of opaque generalisations. *Phonology*, 24(2):217–259.

Ballard, W. L. (1988). *The History and Development of Tonal Systems and Tonal Alternations in South China*. Institute for the Study of Languages and Cultures of Asia and Africa.

Bhaskar, S., Chandlee, J., Jardine, A., and Oakden, C. (2020). Boolean monadic recursive schemes as a logical characterization of the subsequential functions. In Leporati, A., Martin-Vide, C., Shapira, D., and Zandron, C., editors, *Language and Automata Theory and Applications: 14th International Conference LATA 2020*, pages 157–169.

Büchi, J. R. (1960). Weak second-order arithmetic and finite automata. *Zeitschrift für Mathematische Logik und Grundlagen der Mathmatik*, 6:66—92.

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

Chandlee, J. (2017). Computational locality in morphological maps. *Morphology*, 27(4):599–641.

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

Chandlee, J., Heinz, J., and Jardine, A. (2018). Input strictly local opaque maps. *Phonology*, 35(2):171–205.

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

Chandlee, J. and Jardine, A. (2020). Recursive schemes for phonological analysis. Unpublished ms, Haverford College and Rutgers University.

Chen, M. Y. (1987). The syntax of Xiamen tone sandhi. *Phonology Yearbook*, 4:109–150.

Chen, M. Y. (2000). *Tone sandhi: Patterns across Chinese dialects*, volume 92. Cambridge University Press.

Courcelle, B. (1994). Monadic second-order definable graph transductions: a survey. *Theoretical Computer Science*, 126:53–75.

Dolatian, H. and Rawski, J. (2020). Multi-input strictly local functions for templatic morphology. *Proceedings of the Society for Computation in Linguistics*, 3(1):282–296.

Enderton, H. (2001). *A mathematical introduction to logic*. Elsevier.

Engelfriet, J. and Hoogeboom, H. J. (2001). Mso definable string transductions and two-way finite transducers. *ACM Transactions on Computational Logic*, 2:216–254.

Green, A. D. (2006). The independence of phonology and morphology: The Celtic mutations. *Lingua*, 116(11):1946–1985.

Heinz, J. (2018). The computational nature of phonological generalizations. In Hyman, L. and Plank, F., editors, *Phonological Typology*, Phonetics and Phonology, chapter 5, pages 126–195. De Gruyter Mouton.

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

Hyman, L. M. (1975). *Phonology: Theory and Analysis*. New York: Holt, Rinehart and Winston.

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

Kaplan, R. and Kay, M. (1994). Regular models of phonological rule systems. *Computational Linguistics*, 20:331–78.

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

Kirchner, R. (1996). Synchronic chain shifts in Optimality Theory. *Linguistic Inquiry*, 27(2):341–350.

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

Lindell, S. and Chandlee, J. (2016). A logical characterization of strictly local functions. Paper presented at the Fourth Workshop on Natural Language and Computer Science, New York, NY. Available at `http://chandlee.sites.haverford.edu/wp-content/uploads/2020/06/Quantifier-free-is-local.pdf`.

Lundskær-Nielsen, T. and Holmes, P. (2011). *Danish: An essential grammar*. Routledge.

Magri, G. (2018). Idempotency in optimality theory. *Journal of Linguistics*, 54(1):139–187.

Mak, C. (1953). A comparison of two Mixtec tonemic systems. *International Journal of American Linguistics*, 19(2):85–100.

McCarthy, J. J. (1999). Sympathy and phonological opacity. *Phonology*, 16:331–399.

Mohri, M. (1997). Finite-state transducers in language and speech processing. *Computational Linguistics*, 23(2):269–311.

Moreton, E. (2004). Non-computable functions in Optimality Theory. In McCarthy, J. J., editor, *Optimality theory in phonology: A reader*, pages 141–164. Blackwell Publishing.

Mortensen, D. (2006). *Logical and substantive scales in phonology*. PhD thesis, University of California Berkeley.

Moschovakis, Y. N. (2019). *Abstract recursion and intrinsic complexity*, volume 48. Cambridge University Press.

Ní Chiosáin, M. (1991). *Topics in the phonology of Irish*. PhD thesis, University of Massachusetts, Amherst.

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

Pater, J. (2018). Substance matters: a reply to Jardine (2016). *Phonology*, 35(1):151–156.

Rolle, N. R. (2018). *Grammatical tone: typology and theory*. PhD thesis, University of California, Berkeley.

Skilton, A. (2017). Phonology and nominal morphology of Cushillococha Ticuna. Unpublished PhD prospectus. University of California, Berkeley.

Strother-Garcia, K. (2018). Imdlawn Tashlhiyt Berber syllabification is quantifier-free. *Proceedings of the Society for Computation in Linguistics (SCiL) 2018*, pages 145–153.

Tesar, B. (2014). *Output-Driven Phonology: Theory and Learning*. Cambridge Studies in Linguistics.

Tsay, J. and Myers, J. (1996). Taiwanese tone sandhi as allomorph selection. In *Annual Meeting of the Berkeley Linguistics Society*, volume 22, pages 395–405.

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

Wilson, C. (2006). Unbounded spreading is myopic. Paper presented at the workshop Current Perspectives in Phonology.