Recursive Schemes for Phonological Analysis[*]

Jane Chandlee[†] and Adam Jardine[‡]

March 31, 2020

1 Introduction

A central goal of generative linguistics is to determine abstract universals that govern the shape and structure of language (Chomsky, 1957; Baker, 2009). An abstract *computational* universal of natural language phonology is that it is *regular* (Johnson, 1972; Kaplan and Kay, 1981, 1994). Briefly, a pattern is regular if and only if it can be computed with constant memory; that is, there is some fixed amount of memory for which the pattern can be computed for any input string, regardless of its length. Recent work has pursued stronger computational universals for phonology by positing further constraints on the nature of this memory (Heinz, 2009; Heinz and Lai, 2013; Chandlee and Heinz, 2018, a.o.). This work argues that these universals allow for restrictive, yet robust typological generalizations about the shape and structure of phonology that are unavailable to previous derivational or Optimality Theory (OT)-based theories (Chandlee et al., 2018; Heinz, 2018; Jardine, 2019).

However, a criticism of a perspective that focuses on computational universals is that it does not allow for a theory that intensionally represents phonological generalizations in the way phonologists recognize them. Writes Pater (2018, p. 156): "it is a mistake to conceive of [such computational work] as an alternative to OT (and other theories with similar goals)" because "it provides no obvious way of stating the kinds of substantive restrictions on phonological systems that are needed to delimit phonological typology."

In this article, we refute this criticism by presenting a computational formalism that retains the important discoveries about the computational restrictiveness of phonology but in a way that better aligns with commonly-held assumptions about phonological representations and grammars. Specifically, we introduce *Boolean Monadic Recursive Schemes (BMRSs)* and demonstrate their utility for phonological analysis. As we will show and discuss, this formalism has a well-understood complexity bound that corresponds to previous results in the study of computational phonology, but it also provides a way to implement phonological substance. As such it addresses several prior criticisms of computational approaches to phonology. Furthermore, we argue that it solves problems of both rule-based, derivational

[*]We sincerely thank Siddharth Bhaskar for teaching us about recursive schemes, as well as Steve Lindell, Jeff Heinz, Eric Bakovic, Adam McCollum, Anna Mai, Eric Meinhardt, and the Rutgers Math Ling research group for their input and suggestions.

[†]Haverford College, jchandlee@haverford.edu

[‡]Rutgers University, adam.jardine@rutgers.edu

frameworks and OT grammars. Thus, the BMRS formalism is a theory of phonology that captures both intensional *and* computational generalizations about phonology in a way that is unavailable to previous frameworks.

BMRSs are based on the well-studied concept of recursive program schemes, which are an abstract way of studying the complexity of algorithms (Moschovakis, 2019). Algorithms are crucial to linguistic theory, no less so in either rule- or constraint-based phonological frameworks. Algorithms are commonly posited in derivational phonology (e.g., syllabification, autosegmental association paradigms, stress-building paradigms, etc.). And a single algorithm that does all of these things is central to theories like Optimality Theory, which implements them all via optimization over an infinite set of candidates.

BMRSs implement a simple 'if...then...else' structure over properties of elements in phonological representations. Here we give a brief, simple example of what a BMRS analysis looks like, with a more thorough explanation of the formalism to follow in §3. The heart of a BMRS analysis is a system of logical predicates that establish the conditions under which a segment holds a particular feature value in the output. For example, consider a process in which a high (H) tone, marked below as an acute accent, spreads from an underlying syllable rightward, up to the penult.

(1) $/\sigma\acute{\sigma}\sigma\sigma\sigma/ \rightarrow [\sigma\acute{\sigma}\acute{\sigma}\acute{\sigma}\sigma]$

The crucial question here in computing the output is determining whether or not an output syllable has the property of carrying a high tone. (For now we abstract away from non-string representations like autosegmental representations, though we discuss how to extend BMRSs to non-string representations at the end of the article.) The BMRS predicate below describes how to compute this. The variable x refers to any given input syllable, $\top$ means true, and $\bot$ means false.

(2)
$$\begin{array}{rcl} \acute{\Box}_{\text{o}}(x) & = & \texttt{if } \text{final}(x) \texttt{ then } \bot \texttt{ else} \\ & & \texttt{if } \acute{\Box}_{\text{o}}(p(x)) \texttt{ then } \top \texttt{ else} \\ & & \acute{\Box}(x) \end{array}$$

The equation in (2) defines when $\acute{\Box}_{\text{o}}(x)$; that is, when x is H-toned in the output. The first line on the right-hand side of the equation states that if x is final, then $\acute{\Box}_{\text{o}}(x)$ evaluates to $\bot$ (false). This implements a non-finality condition on H-tone spreading. If x is *not* final, then evaluation goes to the second line, which states that if the syllable immediately preceding x (denoted $p(x)$) is H-toned in the output ($\acute{\Box}_{\text{o}}(p(x))$), then $\acute{\Box}_{\text{o}}(x)$ evaluates to $\top$; that is, x is H-toned in the output. This implements the iterative nature of spreading. The final line simply states that (non-final) H-tones are copied over faithfully from the input to the output.

BMRSs maintain one of the main strengths of OT in implementing a ranking of constraint-like predicates that identify particular structures in either the input or output. These structures may alternately license or block particular feature values in the output depending on how they are fit into the overall BMRS template. For example, in (2), final(x) is a *blocking structure* for an output H tone, because it blocks it from surfacing on x. In contrast, $\acute{\Box}_{\text{o}}(p(x))$ is a *licensing structure* for an output H tone, because it causes it to surface on x. Furthermore, the BMRS syntax arranges these structures into a *hierarchy* of local structures that functions not unlike a constraint ranking in OT. For example, in (2), the blocking structure final(x) supercedes the licensing structure $\acute{\square}_o(p(x))$, and thus any x that satisfies both will *not* surface with a H tone.

In contrast to OT, however, the evaluation of a BMRS hierarchy is necessarily *local* in nature, and therefore avoids the computational over-generation that has been attributed to OT's global evaluation strategy (Frank and Satta, 1998; Gerdemann and Hulden, 2012; Lamont, 2018). In fact, BMRSs are guaranteed to describe a strict subclass of the regular functions (Bhaskar et al., 2020).

Thus, through BMRSs, we have a computational characterization of a phonological grammar that offers the following advantages. One, it captures both input and output-based mappings, because the output predicates can refer to either the input structure or the output structure, or both. Two, it intensionally expresses phonologically significant generalizations (as opposed to, e.g., automata). Three, it directly captures "do X unless Y"-type behavior using the if...then...else syntax and interpretation. Four, it captures typological conspiracies by implementing markedness constraints that can variably serve as licensing and blocking structures. And five, it is connected to previous formal results on the computational complexity and learnability of phonology.

The remainder of the paper is structured as follows. In §2 we situate BMRS analyses in the context of a theory of phonology that captures the computational nature of phonology. In §3 we explain in detail how BMRSs are defined and how they represent a phonological input-output mapping. Then in §4 we apply BMRSs to analyses of three significant phonological case studies that demonstrate the advantages of this formalism. These case studies include the interaction of stress and length in Hixkaryana (§4.1), Elsewhere Condition effects (§4.2), and the typology of *NC̥ (§4.3). §5 then discusses a few potential questions and points of interest raised by the analyses, before §6 concludes.

2 Motivation

We first take some time to detail and motivate the computational characterizations of phonology that BMRS analyses are meant to capture. Readers already familiar with this work can safely skip to §3.

From the perspective of typology, one goal of a generative theory of phonology is to characterize possible phonological generalizations as opposed to impossible ones. The observation that phonological generalizations are regular (Johnson, 1972; Kaplan and Kay, 1994; Heinz, 2018) delineates phonological generalizations from many logically possible ones, as most computations are not regular (see, e.g., Immerman 1980). For example, consider a hypothetical unbounded tone spreading pattern which instead spreads to the center-most syllable of the word.

(3) $/\acute{\sigma}\sigma\sigma\sigma\sigma\sigma\sigma/ \rightarrow [\acute{\sigma}\acute{\sigma}\acute{\sigma}\acute{\sigma}\sigma\sigma\sigma]$

Any procedure that finds the center of a word requires an amount of memory proportional to the length of the word, because it must keep track of the number of syllables in the word. Because regular computations can only have a fixed memory, the statement that phonology is regular thus explains why (3) is unattested. Incorporating the observation that phonology is regular into a generative theory of phonology thus provides for a restrictive theory of phonological typology that characterizes precisely what kinds of processes should be possible versus what kinds of processes should be impossible (Heinz, 2018).

Even more restrictive statements can be made. Heinz and Lai (2013) posit the hypothesis that phonological processes are *subsequential* (Mohri, 1997). Subsequential processes are those that are both regular *and* can be computed deterministically. That is, at any point in reading the input, there is exactly one choice that can be made for the output. For example, a deterministic *finite-state transducer (FST)* in Fig. 1 can compute the attested unbounded high tone spreading pattern given in (1).

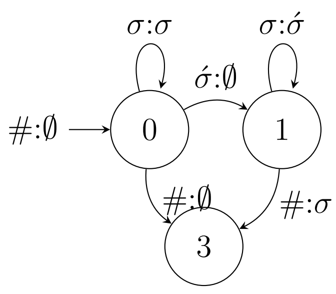

Figure 1: A FST computing unbounded H-tone spread to the penult. States are represented as circles, and transitions are represented as arrows with labels of the form "input:output."

The FST in Fig. 1 operates on an underlying string of syllables as follows. It begins in *state* 0 (states are depicted as circles), reading the beginning word boundary # and outputting nothing ($\emptyset$). From state 0, it reads in the next symbol in the input. If the next symbol is a toneless syllable, it takes the *transition* labeled "σ:σ", meaning that upon reading a toneless syllable in the input (marked on the left of the colon), it outputs a toneless syllable (marked on the right of the colon). Note that this transition loops on state 0, so this process repeats for any number of toneless syllables. However, if an input H-toned syllable is read from state 0, the transition labeled "$\acute{\sigma}$:$\emptyset$" is taken to state 1. This means that the syllable is first output as nothing ("σ:$\emptyset$"), because the machine does not yet know if that syllable is final (and thus should not receive a high tone). From state 1, any subsequent syllable produces an output H-toned syllable. This represents assigning H to the *preceding* syllable, which has now been determined to be nonfinal (and thus susceptible to spreading). Upon seeing the word boundary, the machine outputs a single non-high syllable ('#:σ'), which realizes the final syllable as not H-toned.[1]

Importantly, the FST in Fig. 1 is deterministic: at any point in the input, there is only one transition that can be taken. Deterministic FSTs are strictly less expressive than nondeterminsitic ones (Mohri, 1997), essentially because they have a 'bounded lookahead'. That is, the determinism forces any change to a target in the input be made some fixed number of steps after it has been seen.

Heinz and Lai (2013) point out that spreading in phonology operates in this way, and that the subsequential hypothesis correctly distinguishes attested spreading processes from

[1]Note this FST would delete the H-tone from a monosyllabic form, under the assumption that final syllables can't bear a tone. However, that assumption isn't necessary for the map to be deterministic.

possible, yet unattested processes. Two examples they give are 'sour grapes' (Wilson, 2003, 2006) and 'majority rules' (Baković, 2000), both of which are generable in OT but which require unbounded lookahead. This aligns with the independent characterization of spreading as myopic (Wilson, 2003; Kimper, 2012). Empirically, this hypothesis has also been borne out with studies of long-distance consonant harmony (Luo, 2017) and dissimilation (Payne, 2017). It is also connected to learnability, as subsequential functions are learnable from positive data (Oncina et al., 1993; Jardine et al., 2014), but it is likely that the full regular class is not (which follows from the fact, demonstrated by Gold (1967), that the regular languages are not learnable). Exceptions to this hypothesis in tone (Jardine, 2016) and vowel harmony (McCollum et al., 2017) have been noted, but the fact remains that a significant amount of phonology is subsequential.

However, aside from typological concerns, another goal of a generative phonological theory is to capture linguistically significant generalizations (Chomsky and Halle, 1968). A theory that both incorporates computational characterizations and intensionally captures linguistically significant generalizations has so far proved elusive. For example, the FST in Fig. 1 *extensionally* captures unbounded spreading to the penult, and it makes explicit that the pattern is regular (because the FST has a fixed memory, represented by the states) and subsequential (by the determinism of the FST). However, it does not *intensionally* capture the motivations for the pattern in the usual vocabulary of phonological analysis. For example, the constraint on non-finality (Prince and Smolensky, 1993; Walker, 1996; Yip, 2002) is not explicit anywhere in the formalism.

Furthermore, FSTs that assume alphabets of feature values when defining their transitions (i.e., [+voice], [−son]) have not been widely pursued, in part because their most obvious implementation would lose the property of determinism. As noted above, determinism requires that there be exactly one possible transition from a given state for an input symbol, and that restriction is lost when processing feature bundles (i.e., a /b/ could follow either the [+voice] or the [−son] transition).

Logical descriptions, instead, can capture the same generalizations about the complexity of phonological patterns (Rogers et al., 2013), and can do so with more realistic phonological representations, such as features, syllable structure, or autosegmental representations (Jardine, 2017a; Strother-Garcia et al., 2016; Strother-Garcia, 2017). However, previous work applying logical descriptions to phonological processes have been defined entirely on the input (Heinz, forthcoming), thereby missing generalizations about output-based constraints motivating processes. Furthermore, there has not been a logical characterization of subsequential functions.

However, Bhaskar et al. (2020) present BMRSs as a logical description of the subsequential functions. We apply BMRSs to phonological analyses, and show that they can intensionally capture phonological generalizations—using features, reference to the output, and the interaction of constraints. At the same time, the advantages of automata—in particular that they make the computational properties of the processes they model clear and that they come with proofs of formal learnability—are preserved with BMRSs.

3 Boolean monadic recursive schemes

3.1 Basic structure

Recursive schemes are definitions of the output value of a segment based on a fixed set of (unary) predicates that refer to the input and output structures local to that segment. We illustrate with representations of words as strings of feature bundles, but this easily extends to other structures—for instance, we will momentarily describe a tonal process with syllables and tone values. (We leave detailed investigation of the consequences of varying structure to future work.)

As BMRSs are rather intuitive, readers who are satisfied by the example given in the introduction may skip ahead to Section 4 to see how BMRSs can be applied to more complicated phonological analyses. The purpose of this section is to define BMRSs from the ground up. It serves both as a reference and to show that BMRSs are a precise, well-defined system.

3.1.1 Input feature predicates

The primitives of BMRSs are the boolean values $\top$ (true) and $\bot$ (false) and a finite set of *monadic predicates* $P(t)$ that take a single argument t and return $\top$ or $\bot$. We can apply this to strings of feature bundles as follows.

For a set $\mathbb{F} = \{[(\pm)\text{F}], [(\pm)\text{G}], ..., [(\pm)\text{Z}]\}$ of valued features and a boundary symbol #, we use a set $\mathcal{I} = \{[\text{F}](t), [\text{G}](t), ..., [\text{Z}](t), \#(t)\}$ of *input feature predicates* as well as a special set of *output feature predicates* $\mathcal{O} = \{[\text{F}]_{\text{o}}(t), [\text{G}]_{\text{o}}(t), ..., [\text{Z}]_{\text{o}}(t), \#_{\text{o}}(t)\}$; the latter marked with subscript o.

The argument t stands for some *term* which ranges over the elements in a word—in our case, the segments and boundaries #. Terms are defined inductively as follows: the variable x is a term, and if t is a term, then $p(t)$ is a term referring to the *predecessor* of t, and $s(t)$ is a term referring to the *successor* of t.

We first focus on explaining the input feature predicates, using the word model in Fig. 2 as an example. Fig. 2 represents the word [tɛd] with each feature bundle indicated with its usual IPA value and the predecessor and successor of each segment explicitly marked with arrows. The distinct elements in the representation—the three segments and the word boundaries—are numbered with indices.

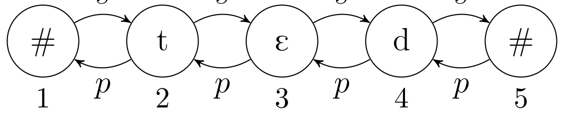
$$\begin{tikzcd}
\# \arrow[r, bend left, "s"] & \text{t} \arrow[l, bend left, "p"] \arrow[r, bend left, "s"] & \text{ɛ} \arrow[l, bend left, "p"] \arrow[r, bend left, "s"] & \text{d} \arrow[l, bend left, "p"] \arrow[r, bend left, "s"] & \# \arrow[l, bend left, "p"] \\
1 & 2 & 3 & 4 & 5
\end{tikzcd}$$

Figure 2: An example model of the word /tɛd/

Table 1 gives some example input predicates, given a standard set of features, taking various terms as arguments. They are listed with their truth values for each element in Fig. 2.

	#	t	ɛ	d	#
	1	2	3	4	5
[son](x)	$\bot$	$\bot$	$\top$	$\bot$	$\bot$
[voi](x)	$\bot$	$\bot$	$\top$	$\top$	$\bot$
$\#(x)$	$\top$	$\bot$	$\bot$	$\bot$	$\top$
$\#(s(x))$	$\bot$	$\bot$	$\bot$	$\top$	$\bot$
[cor]$(p(p(x)))$	$\bot$	$\bot$	$\bot$	$\top$	$\bot$

Table 1: The values of some input predicates given the segments in Fig. 2

For example, [son](x) is true when x is interpreted the [+son] element 3 in Fig. 2; i.e. the sole vowel [ɛ]. The predicate evaluates to false for all other segments. Similarly, [voi](x) is true only for 3 and 4, and $\#(x)$ is true only for 1 and 5.

Note that the feature predicates are interpreted as follows: [F](t) is true for a position t when t is [+F] in the input, and false when t is [−F] in the input. Clearly, this equates the plus and minus values of a feature to the boolean values of true and false. We do this because in a binary feature system, it is redundant to have separate predicates [+F](t) and [−F](t) (as the former is true if and only if the latter is false). This cannot, of course, deal with unspecified feature values. To keep the exposition focused on the logic of BMRSs, we maintain this simplification throughout the paper, but revisit an implementation of three (or more)-valued feature systems in the discussion section.

The last two rows show predicates whose arguments refer to successors and predecessors. We can read $\#(s(x))$ as "the successor of x is #"; thus, this is true only when x is evaluated to 4. Likewise, [cor]$(p(p(x)))$ can be read "the predecessor of the predecessor of x is coronal"—in other words, the segment two elements preceding x is coronal. In Fig. 2, this is only true for 4.[2]

3.1.2 Logical control and defining local structures

The logical backbone of BMRSs are *expressions* built out of these basic predicates and an `if`...`then`...`else` structure as defined inductively below.

(4)
a. $\top$ and $\bot$ are expressions;
b. any predicate $P(t)$ is an expression;
c. if E_1, E_2, and E_3, are expressions, then

$$\texttt{if } E_1 \texttt{ then } E_2 \texttt{ else } E_3$$

is an expression.

An expression E of the form "`if` E_1 `then` E_2 `else` E_3" returns a value as follows. First E_1 is evaluated; if it is true, then E returns the result of evaluating E_2. If E_1 is false,

[2]Note that for the initial element $p(x)$ is undefined (e.g., there is no predecessor of 1 in Fig. 2) and likewise for the final element $s(x)$ is undefined (e.g. there is no successor of 5 in Fig. 2). We adopt the convention that $P(t)$ evaluates to $\bot$ (false) whenever t is undefined.

then E returns the result of evaluating E_3. The left-hand side of Fig. 3 gives a schematic representation of this evaluation. Note that BMRS conditionals evaluate like "if...then" statements in programming languages, *not* like logical implication. That is, E_1 does not directly contribute to the output of E. This is in contrast to a logical implication such as "if E_1 then E_2," in which the value of the entire statement is understood to be true if E_1 is false.

$$\begin{tikzcd}
& {} \arrow[d] & \\
& E_1 \arrow[dl, "\top"'] \arrow[dr, "\bot"] & \\
E_2 \arrow[d] & & E_3 \arrow[d] \\
{} & & {}
\end{tikzcd}
\qquad
\begin{tikzcd}
& {} \arrow[d] & \\
& {[\text{son}](x)} \arrow[dl, "\top"'] \arrow[dr, "\bot"] & \\
\bot \arrow[d] & & {[\text{voi}](x)} \arrow[d] \\
{} & & {}
\end{tikzcd}$$

Figure 3: Evaluation of `if` ... `then` ... `else` statements (left), with (5) an example (right).

To illustrate, the below expression defines what it means to be a voiced obstruent.

(5) $$\begin{bmatrix} -\text{son} \\ +\text{voi} \end{bmatrix}(x) = \texttt{if } [\text{son}](x) \texttt{ then } \bot \texttt{ else } [\text{voi}](x)$$

As shown in the right-hand side of Fig. 3, $\begin{bmatrix} -\text{son} \\ +\text{voi} \end{bmatrix}(x)$ first checks if x is [son] (= E_1). If [son] returns $\top$, then $\bot$ (= E_2) is returned—this implements the logic that if a segment is a sonorant, then it cannot be a voiced obstruent. If [son] returns $\bot$, then the value of [voi](x) (= E_3) is returned. Thus if [voi](x) is true, then we know that x is both voiced and an obstruent. Thus, $\begin{bmatrix} -\text{son} \\ +\text{voi} \end{bmatrix}(x)$ returns $\top$ if and only if x is a voiced obstruent. The reader can confirm via Table 2 that this is true for the segments in the example word from Fig. 2.

		#	t	ɛ	d	#
		1	2	3	4	5
E_1	[son](x)	$\bot$	$\bot$	$\top$	$\bot$	$\bot$
E_2	$\bot$			$\bot$		
E_3	[voi](x)	$\bot$	$\bot$		$\top$	$\bot$
$\begin{bmatrix} -\text{son} \\ +\text{voi} \end{bmatrix}(x)$		$\bot$	$\bot$	$\bot$	$\top$	$\bot$

Table 2: Evaluation of (5) given the segments in Fig. 2. A cell in a column for a segment is filled only if the predicate on the right is evaluated for that segment.

Before moving on, it is important to note that (5) names an expression with the shorthand predicate $\begin{bmatrix} -\text{son} \\ +\text{voi} \end{bmatrix}(x)$. In this way, we can derive new predicates that are not in the original set of primitive predicates that we started with. However, because the right-hand side of the equation is a BMRS expression, $\begin{bmatrix}-\text{son}\\+\text{voi}\end{bmatrix}(x)$ is also a BMRS expression. In other words, we have defined nothing new. This is important because we know that we have not extended the computational power of BMRSs—we have simply defined a 'shorthand' that allows us to easily refer to natural classes.

We will now show how we can derive *local structures* around x—that is, we can derive new predicates that refer not only to the properties of x but also to the properties of elements within a fixed distance of x. This allows us to define predicates that perform a function similar to structural descriptions in SPE rules or markedness constraints in OT.

For example, the following expression $\underline{\begin{bmatrix}-\text{son}\\+\text{voi}\end{bmatrix}}\#(x)$ identifies word-final voiced obstruents.[3]

(6) $$\underline{\begin{bmatrix}-\text{son}\\+\text{voi}\end{bmatrix}}\#(x) = \texttt{if } \begin{bmatrix}-\text{son}\\+\text{voi}\end{bmatrix}(x) \texttt{ then } \#(s(x)) \texttt{ else } \bot$$

This expression is evaluated similar to the definition for $\begin{bmatrix}-\text{son}\\+\text{voi}\end{bmatrix}(x)$ in (5). Here, $\begin{bmatrix}-\text{son}\\+\text{voi}\end{bmatrix}(x)$ itself is used as the initial conditional E_1. If it is false, then $\bot$ is returned. If it is true, then the value for $\#(s(x))$ is returned. Recall that $\#(s(x))$ returns $\top$ if and only if the successor of x is the word boundary; in other words, if x is word final. Thus, $\underline{\begin{bmatrix}-\text{son}\\+\text{voi}\end{bmatrix}}\#(x)$ returns $\top$ if and only if x is both a voiced obstruent and word-final. Examples of how this expression is evaluated are given in Table 3.

		#	b	æ	d	#
		1	2	3	4	5
E_1	$\begin{bmatrix}-\text{son}\\+\text{voi}\end{bmatrix}(x)$	$\bot$	$\top$	$\bot$	$\top$	$\bot$
E_2	$\#(s(x))$		$\bot$		$\top$	
E_3	$\bot$	$\bot$		$\bot$		$\bot$
	$\underline{\begin{bmatrix}-\text{son}\\+\text{voi}\end{bmatrix}}\#(x)$	$\bot$	$\bot$	$\bot$	$\top$	$\bot$

Table 3: Evaluation of (6) given the segments in the word /bæd/. A cell in a column for a segment is filled only if the predicate on the right is evaluated for that segment.

Thus, the expression $\underline{\begin{bmatrix}-\text{son}\\+\text{voi}\end{bmatrix}}\#(x)$ defined in (6) identifies a local structure—that of a word-final voiced obstruent—and has been so named. Note that in the name of the expression $\begin{bmatrix}-\text{son}\\+\text{voi}\end{bmatrix}$ has been underlined to indicate x's position in the structure. This notation will be

[3]Equivalently, we could use the standard logical conjunction (i.e.,[-son,+voice](x) $\wedge$ #(s(x)). However, for the sake of consistency, we will use the `if` ... `then` ... `else` ... syntax for all definitions in the paper.

used throughout the paper. The following section shows how BMRSs can be used to define a map from underlying forms to surface forms by defining output feature predicates that refer to the input feature predicates defined in this section.

3.1.3 Defining output feature predicates

Predicate logics can define maps by defining the output in terms of the input (Courcelle, 1994; Engelfriet and Hoogeboom, 2001).[4] This means that we can define a phonological map using featural representations through a BMRS definition that specifies the featural content of each output element. These definitions describe exactly when an input segment will have that feature in the output.

For example, consider the following word-final obstruent devoicing rule.

(7) $[-\text{son}] \rightarrow [-\text{voice}]\ /\ \_\#$

Clearly, the crux of describing this rule is determining when a segment surfaces as [+voice] and when a segment surfaces as [−voice]. We can do this with the following BMRS definition of the output feature predicate $[\text{voi}]_\text{o}(x)$ (8).

(8) $[\text{voi}]_\text{o}(x) = \texttt{if}\ \underline{\begin{bmatrix} -\text{son} \\ +\text{voi} \end{bmatrix}}\#(x)\ \texttt{then}\ \bot\ \texttt{else}\ [\text{voi}](x)$

This states: the value of $[\text{voi}]_\text{o}(x)$ for x is false—that is, x is [−voi] in the output—if it is a word-final voiced obstruent. Otherwise, $[\text{voi}]_\text{o}(x)$ takes the value x has for $[\text{voi}](x)$ in the input—that is, it is output faithfully. In other words, as long as it is not a word-final voiced obstruent, x surfaces as [+voi] if it was [+voi] in the input and [−voi] if it was [−voi] in the input. This is illustrated in Table 4.

The output values for other features, such as [±sonorant], [coronal], [labial], etc., do not change—and so we define the output feature predicates for these features to be faithful to their inputs. Thus, we have something like in (9).

(9)
$$\begin{array}{lcl} [\text{son}]_\text{o}(x) & = & [\text{son}](x) \\ [\text{cor}]_\text{o}(x) & = & [\text{cor}](x) \\ [\text{lab}]_\text{o}(x) & = & [\text{lab}](x) \end{array}$$

Assuming, as a toy example, that this is an exhaustive set of features for our representations ([±voi],[±son],[cor],[lab]), then we have defined all we need to determine the output for any input segment.

More generally, a full BMRS definition of a map for featural representations is as follows:

(10) Given an input set of valued features $\mathbb{F} = \{[(\pm)\text{F}], [(\pm)\text{G}], ..., [(\pm)\text{Z}]\}$, a BMRS

[4] For those familiar with the concept this is an extension of the well-studied notion of *logical interpretations* (as defined in, e.g., Enderton, 1972).

	#	b	æ	d	#
	1	2	3	4	5
$\underline{\begin{bmatrix}-\text{son}\\+\text{voi}\end{bmatrix}\#(x)}$	$\bot$	$\bot$	$\bot$	$\top$	$\bot$
$\bot$				$\top$	
$[\text{voi}](x)$	$\bot$	$\top$	$\top$		$\bot$
$[\text{voi}]_{\text{o}}(x)$	$\bot$	$\top$	$\top$	$\bot$	$\bot$

	#	b	æ	d	#
	$1'$	$2'$	$3'$	$4'$	$5'$
[voi]	$-$	$+$	$+$	$-$	$-$

Table 4: Evaluation of the output predicate definitions in (8) for devoicing given the input /bæd/. The bottom table shows the value for [±voi] for each output segment i' corresponding to input segment i.

definition of a map is a list of definitions

$$\begin{array}{lcl}
\texttt{out}(x) & = & E_0 \\
[\text{F}]_{\text{o}}(x) & = & E_1 \\
[\text{G}]_{\text{o}}(x) & = & E_2 \\
 & \dots & \\
[\text{Z}]_{\text{o}}(x) & = & E_n
\end{array}$$

where each E_i is a BMRS expression and $\texttt{out}(x)$ is a special output predicate that determines whether x has a corresponding output segment (i.e., whether or not it is deleted).

Thus, (11) is a BMRS definition of word-final obstruent devoicing. We assume for now that word boundaries are not included in the output. Fig. 4 shows how these predicates evaluate for each segment in the example input /bæd/, producing the correct output [bæt].

(11)
$$\begin{array}{lcl}
\texttt{out}(x) & = & \texttt{if } \#(x) \texttt{ then } \bot \texttt{ else } \top \\
[\text{voi}]_{\text{o}}(x) & = & \texttt{if } \underline{\begin{bmatrix}-\text{son}\\+\text{voi}\end{bmatrix}\#(x)} \texttt{ then } \bot \texttt{ else } [\text{voi}](x) \\
[\text{son}]_{\text{o}}(x) & = & [\text{son}](x) \\
[\text{cor}]_{\text{o}}(x) & = & [\text{cor}](x) \\
[\text{lab}]_{\text{o}}(x) & = & [\text{lab}](x)
\end{array}$$

As none of the analyses to follow in Sec. 4 include epenthesis, we abstract away from how to implement this. However, there are established ways of doing this in logical transductions, by specifying copies of input elements. We refer interested readers to, e.g., Strother-Garcia (2017) for examples.

	#	b	æ	d	#
	1	2	3	4	5
$\texttt{out}(x)$	$\bot$	$\top$	$\top$	$\top$	$\bot$
$[\text{voi}]_\text{o}(x)$	$\bot$	$\top$	$\top$	$\top$	$\bot$
$[\text{son}]_\text{o}(x)$	$\bot$	$\bot$	$\top$	$\bot$	$\bot$
$[\text{cor}]_\text{o}(x)$	$\bot$	$\bot$	$\bot$	$\top$	$\bot$
$[\text{lab}]_\text{o}(x)$	$\bot$	$\top$	$\bot$	$\bot$	$\bot$
	$1'$	$2'$	$3'$	$4'$	$5'$
		b	æ	t	

Figure 4: Evaluation of the BMRS definition for word-final obstruent devoicing in (11) for the example input /bæd/.

3.2 Licensing and blocking

In the definition for $[\text{voi}]_\text{o}(x)$ in (7), the predicate $\underline{\begin{bmatrix} -\text{son} \\ +\text{voi} \end{bmatrix}}\#(x)$ acts as a markedness constraint: if x is in this position in this structure, then $[\text{voi}]_\text{o}(x)$ evaluates to false. Likewise, $[\text{voi}](x)$ is a faithfulness constraint—it states that x's input value for [±voi] should be reproduced faithfully in the output. Like a faithfulness constraint, it can be violated—exactly in case when x satisfies $\underline{\begin{bmatrix} -\text{son} \\ +\text{voi} \end{bmatrix}}\#(x)$, which takes precedence over [voi] in the conditional structure of the definition.

This illustrates how the general structure of BMRSs defines the output in terms of structure-based conditions that may be violated. In general, the output property $A'(x)$ is defined as a BMRS expression with the structure in (12).

(12)
```
A'(x) = if STRUCT_1(x) then {⊤, ⊥} else
        if STRUCT_2(x) then {⊤, ⊥} else
         ...
        if STRUCT_n(x) then {⊤, ⊥} else
        A(x)
```

Where STRUCT$_i$ for any line of the form `if` STRUCT$_i(x)$ `then` $\top$ is a *licensing structure*, because if it evaluates to $\top$ it causes $A'(x)$ to *true* in the output. Likewise, for any line of the form `if` STRUCT$_i(x)$ `then` $\bot$, the structure STRUCT$_i$ is a *blocking structure*, because if it evaluates to $\top$ it causes $A'(x)$ to be *false* in the output.

Importantly, the entire definition of $A'(x)$ is an *ordering* of licensing structures and blocking structures. This is depicted graphically in Fig. 5. If STRUCT$_i$ appears before STRUCT$_j$ in the definition, then STRUCT$_i$ *takes priority over* STRUCT$_j$. In other words, the licensing and/or banning condition expressed by STRUCT$_j(x)$ can be violated if and only if some STRUCT$_i$ earlier in the order has been satisfied. The analyses that follow in this paper give concrete examples of this logic of computation of the output.

$$\begin{tikzcd}
 & \mathrm{STRUC}_1(x) \arrow[dl, "\top"'] \arrow[dr, "\bot"] & & & & \\
\{\top, \bot\} & & \mathrm{STRUC}_2(x) \arrow[dl, "\top"'] \arrow[dr, "\bot"] & & & \\
 & \{\top, \bot\} & & \cdots \arrow[dr, "\bot"] & & \\
 & & & & \mathrm{STRUC}_n(x) \arrow[dl, "\top"'] \arrow[dr, "\bot"] & \\
 & & & \{\top, \bot\} & & A(x)
\end{tikzcd}$$

Figure 5: Ordering of structures in a BMRS definition of an output property $A'(x)$.

3.3 Recursion and output structures

From the discussion so far the *boolean* part of BMRSs should be clear.[5] But the true power of BMRSs lies in their potential for *recursive* predicate definitions. This means that when defining output predicates we can refer to other output predicates, including the one we are currently defining. In the context of a phonological analysis, this means that BMRS definitions of maps can refer to local *output* structures as well.

To illustrate, consider unbounded spreading, as in high (H) tone spreading in Shambaa (Odden, 1982). In Shambaa, an underlying H tone spreads to the right, until reaching the penult.

(13) Shambaa (Odden, 1982)

a.	/ku-hand-a/	[ku-hand-a]	‘to plant’
b.	/ku-fúmbatiʃ-a/	[ku-fúmbátíʃ-a]	‘to tie securely’
c.	/ku-fúmbatiʃ-ij-an-a/	[ku-fúmbátíʃ-íj-án-a]	‘to tie securely for each other’

We can schematize this spreading using the following rule. For expository purposes we shall describe this map in terms of strings of syllables (which we arbitrarily choose as the tone-bearing unit) instead of autosegmental representations.

(14) a. $\sigma \rightarrow \acute{\sigma} / \ \acute{\sigma} \_\_ \ \sigma$(iterative)

b. $/\sigma\acute{\sigma}\sigma\sigma\sigma/ \rightarrow \sigma\acute{\sigma}\acute{\sigma}\sigma\sigma \rightarrow \sigma\acute{\sigma}\acute{\sigma}\acute{\sigma}\sigma \rightarrow [\sigma\acute{\sigma}\acute{\sigma}\acute{\sigma}\sigma]$

The rule in (14a) states that a nonfinal syllable takes an H tone when following another H-toned syllable, and that this process applies iteratively. Thus, the rule applies repeatedly until it no longer can, as illustrated in (14b).

We can capture this map using recursion in a BMRS definition as follows. We use the

[5]The *monadic* part comes from the limitation to a single variable in each term.

input predicates $\acute{\Box}$ to represent the privative property of having a high tone and σ to indicate being a syllable.[6] The truth values of each of these predicates for the input $/\sigma\acute{\sigma}\sigma\sigma\sigma/$ is given in Fig. 6.

	#	σ	$\acute{\sigma}$	σ	σ	σ	#
	1	2	3	4	5	6	7
$\#(x)$	$\top$	$\bot$	$\bot$	$\bot$	$\bot$	$\bot$	$\bot$
$\sigma(x)$	$\bot$	$\top$	$\top$	$\top$	$\top$	$\top$	$\bot$
$\acute{\Box}(x)$	$\bot$	$\top$	$\bot$	$\bot$	$\bot$	$\bot$	$\bot$

Figure 6: Values for the input predicates $\#(x)$, $\sigma(x)$, and $\acute{\Box}(x)$ for the input $/\sigma\acute{\sigma}\sigma\sigma\sigma/$.

As finality is clearly important to this generalization, we first define a predicate final(x) that identifies a word final syllable.

(15) $\quad \text{final}(x) = \texttt{if } \#(s(x)) \texttt{ then } \top \texttt{ else } \bot$

With our input predicates established, we can define unbounded tone spread with the BMRS definition below that defines the output versions of $\sigma(x)$ and $\acute{\Box}(x)$.

(16)

$$\begin{array}{lrcl} \text{a.} & \sigma_{\text{o}}(x) & = & \sigma(x) \\ \text{b.} & \acute{\Box}_{\text{o}}(x) & = & \texttt{if } \text{final}(x) \texttt{ then } \bot \texttt{ else} \\ & & & \texttt{if } \acute{\Box}_{\text{o}}(p(x)) \texttt{ then } \top \texttt{ else} \\ & & & \acute{\Box}(x) \end{array}$$

The definition of $\sigma_{\text{o}}(x)$ states that syllables are all mapped faithfully to the output, and the definition of $\acute{\Box}_{\text{o}}(x)$ states three conditions that govern whether or not a syllable is high-toned in the output. The first is that, if a syllable is final, then it cannot be high ($\texttt{if}$ final(x) $\texttt{then}$ $\bot$). The second states that if the preceding syllable is high *in the output*, then x is high ($\texttt{if}$ $\acute{\Box}_{\text{o}}(p(x))$ $\texttt{then}$ $\bot$). The final line states that any underlying high-toned syllable is mapped faithfully with a high tone.

In the second condition of (16) we see the output predicate $\acute{\Box}_{\text{o}}(x)$ is used in its own definition, to state that an x is high in the output if its predecessor is. This is evaluated as shown in Tab. 5.

The top three lines of Tab. 5 give the truth values for these three conditions. For example, element 2, the first syllable, satisfies none of these conditions: it is not final (final(x) evaluates to $\bot$); its predecessor is not high in the output ($\acute{\Box}_{\text{o}}(p(x))$ evaluates to $\bot$—see Fn. 2); and it itself is not high in the input ($\acute{\Box}(x)$ evaluates to false). Thus $\acute{\Box}_{\text{o}}(x)$ evaluates to $\bot$ for 2. When evaluating element 3, then we know that $\acute{\Box}_{\text{o}}(p(x))$ evaluates to $\bot$, because $\acute{\Box}_{\text{o}}(x)$ is $\bot$ for element 2.[7] However, because 3 is itself high in the input, and thus satisfies $\acute{\Box}(x)$, it does satisfy $\acute{\Box}_{\text{o}}(x)$.

[6] We use $\acute{\Box}$ instead of $\acute{\sigma}$ as the latter implies a binary contrast with non-high σ.

[7] This exposition implies the string is evaluated left-to-right, but it is also possible to evaluate the string right-to-left, by recursively computing the value of $\acute{\Box}_{\text{o}}(s(x))$ each time it is encountered.

	$\#$	σ	$\acute{\sigma}$	σ	σ	σ	$\#$
	1	2	3	4	5	6	7
$\text{final}(x)$	$\bot$	$\bot$	$\bot$	$\bot$	$\bot$	$\top$	$\bot$
$\acute{\square}_{\text{o}}(p(x))$	$\bot$	$\bot$	$\bot$	$\top$	$\top$		$\bot$
$\acute{\square}(x)$	$\bot$	$\bot$	$\top$				$\bot$
$\acute{\square}_{\text{o}}(x)$	$\bot$	$\bot$	$\top$	$\top$	$\top$	$\bot$	$\bot$
$\sigma_{\text{o}}(x)$	$\bot$	$\top$	$\top$	$\top$	$\top$	$\top$	$\bot$
Output		σ	$\acute{\sigma}$	$\acute{\sigma}$	$\acute{\sigma}$	σ	

Table 5: Evaluation of (16) given an input /$\sigma\acute{\sigma}\sigma\sigma\sigma$/. A cell in a column for a segment is filled only if the predicate on the right is evaluated for that segment.

Because $\acute{\square}_{\text{o}}(p(x))$ is a licensing structure for $\acute{\square}_{\text{o}}$ then $\acute{\square}_{\text{o}}(x)$ is also true for 4. Likewise, because $\acute{\square}_{\text{o}}(p(x))$ is true for 5, then $\acute{\square}_{\text{o}}(x)$ is true for 5 as well. This implements the iterativity of the rule: the *output* value of the preceding segment affects the output of a subsequent segment.

Note that $\acute{\square}_{\text{o}}(p(x))$ is also true for element 6. However, crucially, $\text{final}(x)$ is *also* true for 6, and $\text{final}(x)$ is a blocking structure for $\acute{\square}_{\text{o}}$. Because this blocking structure *outranks* (i.e., takes priority over) the licensing structure $\acute{\square}_{\text{o}}(p(x))$, $\acute{\square}_{\text{o}}(x)$ evaluates to $\bot$ for element 6. This captures the generalization that non-finality is a crucial condition for the high tone spreading.

Because we allow output predicates in their own definitions, it is possible to write BMRS definitions that are circular; i.e. non-terminating. An example is given in §4.3 below. It is possible to define exactly the conditions under which this circularity can occur, but that is beyond the scope of this paper (see discussion in Bhaskar et al. 2020). For present purposes, we consider such circular grammars to be non-interpretable and therefore excluded from the typology of possible grammars.

This brief demonstration has shown how BMRSs use recursion to refer to output structures. It has also illustrated how BMRSs capture linguistically significant generalizations through the ranking of licensing and blocking structures. Both of these aspects of BMRSs will be more thoroughly illustrated in the analyses to follow in §4.

3.4 Expressive power

While recursion is a potentially powerful tool, BMRSs are also restricted in their expressivity, as has been noted. Recursive schemes are a useful way of studying algorithms abstractly because their behavior is well understood (Moschovakis, 2019). In particular, Bhaskar et al. (2020) prove that, as long as the recursion goes in one direction—that is, the definitions embed multiple p's or s's but never combine them —BMRSs describe exactly the subsequential functions. Thus, for example, a structure we *cannot* define is a predicate like $\text{center}(x)$ identifying the center syllable in a word, because this would require looking in both directions to identify the distance from the word edges. Thus, BMRSs cannot describe unattested patterns like the "spread to center" pattern mentioned in (3).

We briefly explain why this is the case. Roughly, the recursively-defined monadic predicates correspond to states in a FST. The determinism comes from the bounded lookahead: if we only recurse over, for example, p, the transduction can only "look" backwards an unbounded amount. This corresponds to the notion of bounded lookahead which was described in §2 as crucial to being described by a deterministic FST. The analyses in the following sections all have this property.

4 Analyses

In this section we provide BMRS analyses of three phonological phenomena. These three case studies were chosen to showcase the ability of BMRSs to capture the nature of phonology in three particular areas. The analysis of Hixkaryana stress in §4.1 shows how the hierarchy of licensing and blocking structures in a BMRS analysis captures the conflicting pressures that give rise to phonological patterns. In §4.2, an analysis of English stress shows how Elsewhere Condition effects are derived from the manner in which BMRS are evaluated. Finally, the typology of NC̥ effects in §4.3 shows how varying whether a marked sequence appears as a blocking or licensing structure captures the different repairs languages choose to avoid that marked sequence.

4.1 Stress and length in Hixkaryana

Stress in Hixkaryana (Derbyshire, 1985) is predictable, and Kager (1999)'s analysis of the interaction between stress and weight in Hixkaryana is a well-known argument for OT's ability to capture the interaction of conflicting generalizations in a language. Here we give a BMRS analysis that not only captures these same conflicts, but also addresses some of Halle and Idsardi (2000)'s criticisms of Kager (1999)'s analysis. The following are taken from Kager (1999)'s discussion of the data.

In strings of open syllables, Hixkaryana stress has an iterative, iambic pattern, with long vowels in the stressed syllables. As in Kager (1999), we put aside the distinction between primary and secondary stress.

(17)	a.	[to.róː.no]	'small bird'
	b.	[ne.móː.ko.tóː.no]	'it fell'
	c.	[a.tʃóː.wo.wo]	'wind'
	d.	[k^{w}áː.ja]	'red and green macaw'

Furthermore, as illustrated in (17c) [a.tʃóː.wo.wo] 'wind', Hixkaryana has a nonfinality condition that prevents this iteration from reaching the end of the word (c.f. *[a.tʃóː.wo.wóː]). In bisyllabic words such as (17d) [k^{w}áː.ja] 'red and green macaw', this forces the initial syllable to be stressed (c.f. *[k^{w}a.jáː]).

Because stress and length in open syllables are entirely predictable, neither need be present in the underlying forms. Using a schematic representation of L for light syllables and H for heavy syllables, we can represent this map as in (18). Like Kager, this schematic abstracts away from the portion of the grammar that implements syllabification, and takes syllables as the input.[8]

(18)	a.	LLL	$\mapsto$	[LH́L]	(=(17a))
	b.	LLLLL	$\mapsto$	[LH́LH́L]	(=(17b))
	c.	LLLL	$\mapsto$	[LH́LL]	(=(17c))
	d.	LL	$\mapsto$	[H́L]	(=(17d))

Additionally, Hixkaryana allows closed syllables, which are also treated as heavy and receive stress.

(19)	a.	[ák.ma.táː.ri]	'branch'
	b.	[tóh.ku.rʲéː.ho.na]	'to Tohkurye'
	c.	[nák.ɲóh.játʃ.ke.náː.no]	'they were burning it'
	d.	[kʰa.náː.níh.no]	'I taught you'
	e.	[mi.háː.na.níh.no]	'you taught him'

As (19a) [ák.ma.táː.ri] 'branch' shows, for example, iterative stress over light syllables resets following a heavy syllable. In terms of our schematic notation, we can represent these examples as below.

(20)	a.	HLLL	$\mapsto$	[H́LH́L]	(=(19a))
	b.	HLLLL	$\mapsto$	[H́LH́LL]	(=(19b))
	c.	HHHLLL	$\mapsto$	[H́H́H́LH́L]	(=(19c))
	d.	LLHL	$\mapsto$	[LH́H́L]	(=(19d))
	e.	LLLHL	$\mapsto$	[LH́LH́L]	(=(19e))

A BMRS analysis of the Hixkaryana facts defines when a syllable is heavy in the output ($\mathrm{H}_{\mathrm{o}}(x)$), when it is light in the output ($\mathrm{L}_{\mathrm{o}}(x)$), and when it is stressed in the output ($\acute{\Box}_{\mathrm{o}}(x)$), given the input predicates $\mathrm{L}(x)$ and $\mathrm{H}(x)$. The following will capture this with iteration governed by the notions of *clash* and *lapse* (Prince, 1983; Selkirk, 1984; Gordon, 2002).

The following define what a syllable is (essentially, any H or L)[9] and when a syllable x will be, respectively, in a clash and lapse situation.

(21)

$$\begin{aligned}
&\text{a.} && \sigma(x) &&= \texttt{if } \mathrm{L}(x) \texttt{ then } \top \texttt{ else } \mathrm{H}(x)\\
&\text{b.} && \mathrm{clash}(x) &&= \texttt{if } \sigma(x) \texttt{ then } \acute{\Box}_{\mathrm{o}}(p(x)) \texttt{ else } \bot\\
&\text{c.} && \mathrm{lapse}(x) &&= \texttt{if } \acute{\Box}_{\mathrm{o}}(p(x)) \texttt{ then } \bot \texttt{ else}\\
&&&&& \quad\ \texttt{if } \sigma(p(x)) \texttt{ then } \sigma(x) \texttt{ else } \bot
\end{aligned}$$

The predicate $\mathrm{clash}(x)$ defines a structure in which x is preceded by a stressed syllable in the output. Note that this is defined recursively, using the output predicate $\acute{\Box}_{\mathrm{o}}(x)$. The predicate $\mathrm{lapse}(x)$ is defined essentially as the opposite situation—x is in a lapse structure exactly when it is a syllable preceded by another syllable but not in a clash structure.

[8]This is a safe move because we can assume that the functions $\mathrm{L}(x)$ and $\mathrm{H}(x)$ identifying light and heavy syllables have been defined elsewhere in the grammar; for more on composing BMRSs into a complete grammar see §5.

[9]This is needed to distinguish syllables from word boundaries.

We also make use of a predicate only(x), which identifies the first syllable of mono- and di-syllabic words, as these syllables are always stressed:

```
(22)    only($x$) =  if final($x$) then initial($x$) else
                     if final($s(x)$) then initial($x$) else
                     $\bot$
```

The full definition is given in (23). The basic iterative pattern of stress is given by the final three lines, which show the critical role played by the clash and lapse structures.

```
(23)    $\acute{\square}_{\text{o}}(x)$ =  if only($x$) then $\top$ else
                  if final($x$) then $\bot$ else
                  if H($x$) then $\top$ else
                  if initial($x$) then $\bot$ else
                  if clash($x$) then $\bot$ else
                  lapse($x$)
```

Setting clash(x) as a blocking structure, and lapse as a licensing structure, captures exactly left-to-right binary stress. Setting initial(x) as a blocking structure makes the iteration iambic. This is illustrated with the derivation in Tab. 6 for an input LLLLL (=(17b) [ne.móː.ko.tóː.no] 'it fell'). As with the derivations in the previous section, in 6 and in the derivations throughout this section, a cell in a column for an element is filled only if the predicate on the right is evaluated for that segment.

	#	L	L	L	L	L	#
	1	2	3	4	5	6	7
initial(x)	$\bot$	$\top$	$\bot$	$\bot$	$\bot$	$\bot$	$\bot$
clash(x)	$\bot$		$\bot$	$\top$	$\bot$	$\top$	$\bot$
lapse(x)	$\bot$		$\top$		$\top$		$\bot$
$\acute{\square}_{\text{o}}(x)$	$\bot$	$\bot$	$\top$	$\bot$	$\top$	$\bot$	$\bot$

Table 6: Evaluation of (23) given the input /LLLLL/

As shown in Tab. 6, element 2 in the representation—that is, the first L in LLLLL—satisfies initial(x). This is a blocking structure in the definition of $\acute{\square}_{\text{o}}(x)$, and so 2 evaluates to $\bot$ for the entire expression. An initial L thus will not receive stress in the output.

Next, we can evaluate 3, the second L syllable. Element 3 is not initial, so the evaluation moves on to the next line. This evaluates clash(x), which again is true if and only if $\acute{\square}_{\text{o}}(p(x))$ is true. As we know already that element 2 evaluates to $\bot$, so clash(x) evaluates to $\bot$ for element 3. Conversely, because element 2 evaluates to $\bot$ for $\acute{\square}_{\text{o}}(x)$, lapse($x$) evaluates to $\top$ for 3. As lapse(x) is a licensing structure for $\acute{\square}_{\text{o}}(x)$, $\acute{\square}_{\text{o}}(x)$ evaluates to $\top$ for 3. Thus, element 3 will be stressed in the output. In turn, this means that the blocking structure clash(x) evaluates to $\top$ for 4, so $\acute{\square}_{\text{o}}(x)$ evaluates to false for 4. This in turn means that the licensing structure lapse(x) evaluates to $\top$ for 5, and so it evaluates to $\top$ for $\acute{\square}_{\text{o}}(x)$, and so on.

We now turn to the preceding two lines in the definition in (23) of $\acute{\Box}_{\mathrm{o}}(x)$, which are repeated below in (24).

(24)
```
if $\mathrm{final}(x)$ then $\bot$ else
if $\mathrm{H}(x)$ then $\top$ else
```

As with the unbounded spreading example in §3.3, non-finality is implemented by setting final(x) as a blocking structure. Likewise, setting H(x) as a licensing structure implements the weight-to-stress principle (Kager, 1999, 2007).

Crucially, both of these statements take precedence over the structures responsible for iterative stress. This means that both of them will interrupt the normal iterative placement of stress. This is illustrated in Tab. 7 for an input LLLL (=(17c) [a.tʃóː.wo.wo] 'wind').

	#	L	L	L	L	#
	1	2	3	4	5	6
final(x)	$\bot$	$\bot$	$\bot$	$\bot$	$\top$	$\bot$
H(x)	$\bot$	$\bot$	$\bot$	$\bot$		$\bot$
initial(x)	$\bot$	$\top$	$\bot$	$\bot$		$\bot$
clash(x)	$\bot$		$\bot$	$\top$		$\bot$
lapse(x)	$\bot$		$\top$			$\bot$
$\acute{\Box}_{\mathrm{o}}(x)$	$\bot$	$\bot$	$\top$	$\bot$	$\bot$	$\bot$

Table 7: Evaluation of (23) for the input /LLLL/

First, the output of LLLL is [LH́LL], not *[LH́LH́] because stress avoids the final syllable. This is accomplished by ranking final(x) as a blocking structure above the licensing structures H(x) and lapse(x). Thus, the final L in LLLL (element 5), would satisfy the licensing structure lapse(x) (note that its predecessor evaluates to $\bot$), but it first satisfies final(x). This means that $\acute{\Box}_{\mathrm{o}}(x)$ immediately evaluates to $\bot$, leaving lapse(x) not evaluated. Thus, any final L will not be assigned stress.

Conversely, any H syllable must receive stress. By ordering H(x) as a licensing structure above the blocking structures initial(x) and clash(x), this assigns stress to Hs that are in either of these positions. This is illustrated below in (25a) for an input HLLLL (=(19b) [tóh.ku.r^{j}éː.ho.na] 'to Tohkurye') and in (25b) for HHHLL (=(19c) [nák.ɲóh.játʃ.ke.náː.no] 'they were burning it').

(25) a.

	#	H	L	L	L	L	#
	1	2	3	4	5	6	7
final(x)	$\bot$	$\bot$	$\bot$	$\bot$	$\bot$	$\top$	$\bot$
H(x)	$\bot$	$\top$	$\bot$	$\bot$	$\bot$		$\bot$
initial(x)	$\bot$		$\bot$	$\bot$	$\bot$		$\bot$
clash(x)	$\bot$		$\top$	$\bot$	$\top$		$\bot$
lapse(x)	$\bot$			$\top$			$\bot$
$\acute{\Box}_{\text{o}}(x)$	$\bot$	$\top$	$\bot$	$\top$	$\bot$	$\bot$	$\bot$

b.

	#	H	H	H	L	L	L	#
	1	2	3	4	5	6	7	8
final(x)	$\bot$	$\bot$	$\bot$	$\bot$	$\bot$	$\bot$	$\top$	$\bot$
H(x)	$\bot$	$\top$	$\top$	$\top$	$\bot$	$\bot$		$\bot$
initial(x)	$\bot$				$\bot$	$\bot$		$\bot$
clash(x)	$\bot$				$\top$	$\bot$		$\bot$
lapse(x)	$\bot$					$\top$		$\bot$
$\acute{\Box}_{\text{o}}(x)$	$\bot$	$\top$	$\top$	$\top$	$\bot$	$\top$	$\bot$	$\bot$

In (25a), the initial H (element 2) satisfies the licensing structure H(x) and thus $\acute{\Box}_{\text{o}}(x)$ evaluates to $\top$. This is despite the fact that H is initial and would otherwise satisfy the blocking structure initial(x). Thus, ranking H(x) over intial(x) allows initial Hs to receive stress. Likewise, ranking H(x) over clash(x) captures the generalization that assigning stress to heavy syllables outweighs the drive for placing stress on alternate syllables. Thus, the first three Hs in (25b) evaluate to $\top$ for $\acute{\Box}_{\text{o}}(x)$, even though they would satisfy the blocking structure clash(x).

The analysis in (23) captures stress assignment for all forms of three syllables or more. But as exemplified in (17d) [kʷáː.ja] 'red and green macaw,' shorter words do not follow the above generalizations. Instead, in mono- and di-syllables the first syllable is stressed.

As shown in Tab. 8, for an input LL, setting only(x) as a licensing structure and ordering it before the blocking structures final(x) and initial(x) correctly assigns stress to the initial syllable. (The remaining parts of the definition of $\acute{\Box}_{\text{o}}(x)$ are omitted.)

	#	L	L	#
	1	2	3	4
only(x)	$\bot$	$\top$	$\bot$	$\bot$
final(x)	$\bot$		$\top$	$\bot$
H(x)	$\bot$			$\bot$
initial(x)	$\bot$			$\bot$
$\acute{\Box}_{\text{o}}(x)$	$\bot$	$\top$	$\bot$	$\bot$

Table 8: Evaluation of relevant statements in (23) for input /LL/

With stress captured, the definitions for the output weight of syllables is simple.

(26) a. $\text{H}_{\text{o}}(x) \;=\;$ `if` $\acute{\Box}_{\text{o}}(x)$ `then` $\top$ `else`
H(x)

b. $\text{L}_{\text{o}}(x) \;=\;$ `if` $\acute{\Box}_{\text{o}}(x)$ `then` $\bot$ `else`
L(x)

Definition (26a) for $\text{H}_{\text{o}}(x)$ implements the stress-to-weight principle: any syllable that receives stress becomes heavy. This captures the (near) equivalence of weight and stress. As a default any input heavy syllable also becomes heavy in the output (this only applies to final H syllables that did not receive stress). Definition (26b) is the converse: any syllable that does not receive stress in the input, and was L in the input, surfaces as L.

How (23) and (26) work together to produce the correct outputs for the inputs LLHL (=(19d) [kʰa.náː.níh.no] 'I taught you') and HHHLLL (=(19c) [nák.ɲóh.játʃ.ke.náː.no] 'they were burning it') are shown in the full derivations in Tab. 9. The individual lines of the definition of $\acute{\square}_{\mathrm{o}}(x)$ are given for clarity.

	#	L	L	H	L	#
	1	2	3	4	5	6
only(x)	⊥	⊥	⊥	⊥	⊥	⊥
final(x)	⊥	⊥	⊥	⊥	⊤	⊥
H(x)	⊥	⊥	⊥	⊤		⊥
initial(x)	⊥	⊤	⊥			⊥
clash(x)	⊥		⊥			⊥
lapse(x)	⊥		⊤			⊥
$\acute{\square}_{\mathrm{o}}(x)$	⊥	⊥	⊤	⊤	⊥	⊥
$\mathrm{H}_{\mathrm{o}}(x)$	⊥	⊥	⊤	⊤	⊥	⊥
$\mathrm{L}_{\mathrm{o}}(x)$	⊥	⊤	⊥	⊥	⊤	⊥
Output		L	H́	H́	L	

	#	H	H	H	L	L	L	#
	1	2	3	4	5	6	7	8
only(x)	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥
final(x)	⊥	⊥	⊥	⊥	⊥	⊥	⊤	⊥
H(x)	⊥	⊤	⊤	⊤	⊥	⊥		⊥
initial(x)	⊥				⊥	⊥		⊥
clash(x)	⊥				⊤	⊥		⊥
lapse(x)	⊥					⊤		⊥
$\acute{\square}_{\mathrm{o}}(x)$	⊥	⊤	⊤	⊤	⊥	⊤	⊥	⊥
$\mathrm{H}_{\mathrm{o}}(x)$	⊥	⊤	⊤	⊤	⊥	⊤	⊥	⊥
$\mathrm{L}_{\mathrm{o}}(x)$	⊥	⊥	⊥	⊥	⊤	⊥	⊤	⊥
Output		H́	H́	H́	L	H́	L	

Table 9: Evaluation of (23) and (26) for inputs /LLHL/ and /HHHLLL/

This concludes our BMRS analysis of stress and length in Hixkaryana, which clearly illustrates the ability of BMRSs to state conflicting pressures in the language: overall there is an iterative left-to-right stress-building process, but this is overridden by existing heavy syllables. This is captured in the BMRS analysis by ordering the licensing and blocking structures responsible for iterative stress below the licensing structure responsible for assigning stress to heavy syllables. It also captures the relationship between stress and weight by literally equating output heavy syllables with accented syllables.

It is worth making a few remarks comparing this analysis to previous analyses. It is similar to Hayes (1995)'s analysis, in that it builds iterative stress and then assigns weight based on this stress. However, it is distinct from a derivational procedure in that conflicting licensing and blocking structures are primitives of the system that defines stress. For example, instead of marking final syllables as extrametrical, the BMRS analysis ranks a blocking structure final(x) above the licensing structures (except for only(x)) responsible for the usual assignment of stress.

This latter aspect of the BMRS analysis is somewhat like an OT analysis, in that it implements stress assignment based on a series of ranked structures reminiscent of constraints (e.g., non-finality, clash, lapse). However, the evaluation of these structures given a particular input is fundamentally *local*, as opposed to OT's global evaluation. This is perhaps clearest in the example of an input LLHL, which, as Halle and Idsardi (2000) point out, is incorrectly output under Kager (1999)'s analysis as *[LLH́L] (as opposed to the attested [LH́H́L], e.g.

(19d) [k^{h}a.náː.níh.no] 'I taught you'). This is due to the fact that, in OT, *[LLH́L] is directly compared to [LH́H́L], and the former turns out to be more optimal than the latter given Kager (1999)'s high ranking of Uneven-Iamb, which penalizes iambs of the form (LĹ) and (H́). The BMRS analysis obtains the correct output [LH́H́L] by assigning stress exactly as Halle and Idsardi (2000) describe the generalization in Hixkaryana: iterating from left-to-right "instead of 'looking ahead' to skip a single light syllable in order to form a single canonical iamb" (p. 202). The way the statements in a BMRS analysis are evaluated crucially limits its ability to look ahead.

This leads to another, related point: one cannot verify the accuracy of an OT analysis until one is absolutely sure they have considered all possible output candidates (Bane and Riggle, 2010). In a BMRS analysis, it is easier to check this because each part of the grammar is defined explicitly, and so there is a step-by-step procedure for determining the output for any input. This also relates to a theme of Halle and Idsardi (2000)'s critique of Kager (1999): given a ranking of OT constraints with multiple strata, it is hard to understand what part of the grammar carries which part of the explanatory power. With a BMRS analysis, it is clear exactly how each licensing and blocking structure contributes to the overall map.

Another point of divergence between the analysis presented here and those of Hayes (1995) and Kager (1999) is that, like Gordon (2002), this analysis does not explicitly refer to feet. This is not to refute feet as a psychologically real part of the grammar, but rather was an expository choice to focus on how BMRSs function, instead of how they can be extended to build structures. Logical characterizations can also capture structure building; see, for example the work of Strother-Garcia (2019) and Jardine (2017b). However, the simplicity of the foot-free analysis is striking. It is worth noting that feet are a crucial part of Kager (1999)'s analysis, which claims to explain lengthening of open syllables through a universal principle of (LH́) as the ideal iambic foot. However, this claim does not hold up under Halle and Idsardi (2000)'s detailed analysis of the behavior of Kager's constraints. They show that a preference for (LH́) iambs cannot be encapsulated in a single OT constraint, and instead can only be captured with a conspiracy of various constraints interacting in a fixed ranking. Thus, like Gordon (2002), our aim is to not to thoroughly evaluate the evidence for feet, but instead to give an analysis that captures the pattern.

Finally, a potential weakness of this analysis is the need for the licensing structure only(x) to identify the initial syllable in mono- and di-syllabic forms. While the use of this structure is somewhat stipulative, treating mono- and di-syllabic forms differently is unavoidable. As noted by Halle and Idsardi (2000), both the OT analysis in Kager (1999) and the rule-based analysis of Hayes (1995) require additional machinery that refers specifically to mono- and di-syllabic forms.[10] In Kager (1999)'s analysis, the ranking Ft-Bin $\gg$ Dep-IO motivates lengthening in bisyllabic words, whereas the separate ranking Uneven-Iamb $\gg$ Dep-IO motivates lengthening in longer words (p. 160). In Hayes's rule-based analysis, an additional rule is required, as pointed out by Kager (1999): "When the entire metrical domain is a single light syllable, assign (L) to it" (p. 149, (17), Step 4). Thus, in its treatment of short forms, this BMRS analysis is not significantly different from other analyses.

[10]It is possible to divide only(x) into the slightly more universal licensing structures $1\sigma(x)$ identifying monosyllables and 2σ-first(x) identifying only the initial syllable in di-syllabic words. However, to simplify the notation, we collapse them here.

4.2 Elsewhere Condition effects

The hierarchy of structures in a BMRS definition admits a characterization not unlike Pāṇini's Theorem on Constraint-ranking in OT (Prince and Smolensky, 2004; Baković, 2006) from which Elsewhere Condition (EC; Kiparsky, 1973) effects can be derived.[11] We state this characterization as the Strict Substructure Ordering Principle, or SSOP, given in (27).

(27) The Strict Substructure Ordering Principle (SSOP)
For any ranking of structures in a BMRS definition, whenever $\text{STRUCT}_j(x)$ implies $\text{STRUCT}_i(x)$ but the converse is not true (i.e., STRUCT_i is a strict substructure of STRUCT_j), it must be the case that $i < j$ in the order, otherwise j will never be evaluated.

This follows directly from the logic of the `if` ... `then` ... `else` syntax. The structures only(x) and initial(x) from the Hixkaryana analysis in §4.1 are illustrative. Recall the definition of only(x), repeated in (28).

(28)
$$\begin{array}{lcl} \text{only}(x) & = & \texttt{if}\ \text{final}(x)\ \texttt{then}\ \text{initial}(x)\ \texttt{else} \\ & & \texttt{if}\ \text{final}(s(x))\ \texttt{then}\ \text{initial}(x)\ \texttt{else} \\ & & \bot \end{array}$$

Intuitively, this definition implies initial(x)—the initial syllable in a mono- or di-syllable must, of course, be initial. Formally, the reader can confirm that only(x) is only true when initial(x) is true.

The SSOP then fixes the relative ranking of only(x) and initial(x) in the BMRS definition for Hixkaryana stress. Recall that in (23) only(x) takes priority over initial(x), as repeated in (29a) (with irrelevant lines omitted).

(29) a.
$$\begin{array}{lcl} \acute{\Box}_{\text{o}}(x) & = & \texttt{if}\ \text{only}(x)\ \texttt{then}\ \top\ \texttt{else} \\ & & ... \\ & & \texttt{if}\ \text{initial}(x)\ \texttt{then}\ \bot\ \texttt{else}\ ... \end{array}$$

b.

	# σ #
only(x)	⊤
...	
initial(x)	⊤
$\acute{\Box}_{\text{o}}(x)$	⊤

	# σ σ σ #
only(x)	⊥
...	
initial(x)	⊤
$\acute{\Box}_{\text{o}}(x)$	⊥

If only(x) evaluates to true, then $\acute{\Box}_{\text{o}}(x)$ evaluates to ⊤. If not, then evaluation passes down the line, and if it reaches initial(x) and it evaluates to true, then $\acute{\Box}_{\text{o}}(x)$ evaluates to ⊥. This is illustrated in (29b), which gives the truth values for $\acute{\Box}_{\text{o}}(x)$ for the initial syllable in a monosyllabic form and a trisyllabic form, respectively.

If the priority of the two were reversed, then only(x) would never be evaluated, because for any x for which only(x) is true, initial(x) will be true, and if the latter takes priority it

[11]We thank Chris Oakden for discussion on the points in this section.

will always be evaluated instead. This is illustrated below in (30).

(30) a. $\acute{\Box}_{\text{o}}(x) \; = \; \texttt{if}\ \text{initial}(x)\ \texttt{then}\ \bot\ \texttt{else}\ ...$
$\qquad ...$
$\qquad \texttt{if}\ \text{only}(x)\ \texttt{then}\ \top\ \texttt{else}$

b.

	# σ #
initial(x)	⊤
...	
only(x)	⊤
$\acute{\Box}_{\text{o}}(x)$	⊥

	# σ σ σ #
initial(x)	⊤
...	
only(x)	⊥
$\acute{\Box}_{\text{o}}(x)$	⊥

Note that now the truth value for $\acute{\Box}_{\text{o}}(x)$ in both of the examples in (30b) is ⊥, because even though the initial syllable in the disyllable satisfies only(x), initial(x) preempts it. In other words, only(x) might as well not be present in the definition at all. This illustrates how the SSOP fixes the order in (29) as the only one in which only(x) can play a role in the grammar.

To see how the SSOP predicts EC effects, we turn to stress and length in English. As discussed by Baković (2006) (following Chomsky and Halle, 1968; Myers, 1987; Halle, 1995), long and short vowels in English are in complementary distribution in stressed syllables in certain metrical configurations. In general, the stressed vowel in a binary foot is short. Following the above authors, we likewise assume the final suffixed syllables in these forms are extrametrical.

(31) a.	(n$\underline{\breve{\text{a}}}$tu)ral	(cf. n$\underline{\bar{\text{a}}}$ture)
b.	di(v$\underline{\breve{\text{ı}}}$ni)ty	(cf. div$\underline{\bar{\text{ı}}}$ne)
c.	(r$\underline{\breve{\text{a}}}$di)cal	

However, when stressed [−high] vowels in this configuration are also followed by an [i] vowel in hiatus, they are long instead of short.

(32) a.	re(m$\underline{\bar{\text{e}}}$di)al	*re(m$\underline{\breve{\text{e}}}$di)al
b.	co(l$\underline{\bar{\text{o}}}$ni)al	*co(l$\underline{\breve{\text{o}}}$ni)al
c.	(r$\underline{\bar{\text{a}}}$di)al	*(r$\underline{\breve{\text{a}}}$di)al

Thus, stressed vowels in binary feet are long when followed by an [iV] sequence, and short elsewhere. Because this generalization has a "do X unless the more specific situation in Y holds" flavor, rule-based analyses have argued for the disjunctive ordering of rules according to the EC. The EC is discussed more below, but first we will give a BMRS analysis of these facts.

We assume a predicate brhd(x) that is true exactly when x is the head of a binary branching foot. To identify the environment for lengthening we define the following additional predicate brhd&$\underline{[-\text{hi}]}$CiV(x).

(33) $\text{brhd\&}\underline{[-\text{hi}]}\text{CiV}(x) = \texttt{if}\ \text{brhd}(x)\ \texttt{then}\ \underline{[-\text{hi}]}\text{CiV}(x)\ \texttt{else}\ \bot$

The BMRS definition for the feature [±long] in English would thus be as follows.

(34)

```
$[\text{long}]_\text{o}(x)$ = if $\text{brhd\&}\underline{[-\text{hi}]}\text{CiV}(x)$ then $\top$ else
                      if $\text{brhd}(x)$ then $\bot$ else
                      $[\text{long}](x)$
```

This definition involves one licensing structure, brhd&$\underline{[-\text{hi}]}$CiV(x) (the lengthening environment), ordered before a blocking structure brhd(x) (the shortening environment). Note that this order is fixed by the SSOP: brhd&$\underline{[-\text{hi}]}$CiV(x) implies bhrd(x), so the former must take priority over the latter. (34) is evaluated as shown in (35), with the examples *radical* and *radial*.

(35) a.

	(r$\underline{\text{a}}$di)cal
brhd&$\underline{[-\text{hi}]}$CiV(x)	$\bot$
brhd(x)	$\top$
$[\text{long}]_\text{o}(x)$	$\bot$
Output:	r$\underline{\breve{\text{a}}}$dical

b.

	(r$\underline{\text{a}}$di)al
brhd&$\underline{[-\text{hi}]}$CiV(x)	$\top$
brhd(x)	
$[\text{long}]_\text{o}(x)$	$\top$
Output:	r$\underline{\bar{\text{a}}}$dial

As illustrated in (35a), the stressed vowel in *radical* fails brhd&$\underline{[-\text{hi}]}$CiV(x), so evaluation passes along to bhrd(x). This evaluates to $\top$, and since bhrd(x) is a blocking structure for $[\text{long}]_\text{o}(x)$, the entire predicate evaluates to true. This produces the correct output *rădical*. In contrast, the stressed vowel in *radial* satisfies brhd&$\underline{[-\text{hi}]}$CiV(x). As this is a licensing structure for $[\text{long}]_\text{o}(x)$, the predicate immediately evaluates to $\top$, skipping the evaluation of brhd(x). This produces the correct output *rādial*.

Note that if the order of the two structures were reversed, the licensing structure brhd&$\underline{[-\text{hi}]}$CiV(x) would never have an effect. This is illustrated below in (36).

(36) a.

	(r$\underline{\text{a}}$di)cal
brhd(x)	$\top$
brhd&$\underline{[-\text{hi}]}$CiV(x)	
$[\text{long}]_\text{o}(x)$	$\bot$
Output:	r$\underline{\breve{\text{a}}}$dical

b.

	(r$\underline{\text{a}}$di)al
brhd(x)	$\top$
brhd&$\underline{[-\text{hi}]}$CiV(x)	
$[\text{long}]_\text{o}(x)$	$\bot$
Output:	$*$ r$\underline{\breve{\text{a}}}$dial

Exactly as in the example with only(x) and initial(x), bhrd(x) is a strict substructure of brhd&$\underline{[-\text{hi}]}$CiV(x); logically, the latter implies the former. This means that for any element for which brhd&$\underline{[-\text{hi}]}$CiV(x) is true, so is bhrd(x). Thus, if we were to order bhrd(x) before brhd&$\underline{[-\text{hi}]}$CiV(x), the latter would never be evaluated. In effect, as predicted by the SSOP, brhd&$\underline{[-\text{hi}]}$CiV(x) is extraneous in the grammar.

Note under the BMRS analysis the lengthening and shortening processes behave disjunctively: each vowel is either lengthened or shortened, depending on which structure it satisfies. This is not due to any independent principle, but is simply a result of how the grammar operates.

To contrast this with a rule-based analysis, we briefly discuss how derivational analyses invoke the EC to affect this disjunctive behavior. The below formulations of rules for lengthening and shortening are essentially those of Kenstowicz (1994, p. 218, (36)).

(37) English lengthening and shortening (Kenstowicz, 1994)

a. Shortening

$$\begin{array}{cccccc} V & \rightarrow & \breve{V} & / & \underline{\qquad} & \\ & & & & | & \\ & & & & (\ \acute{\sigma} & \sigma) \end{array}$$

b. Lengthening

$$\begin{array}{cccccccc} V & \rightarrow & \bar{V} & / & \underline{\qquad} & C & i & V \\ & & & & | & & | & \\ & & & & (\ \acute{\sigma} & & \sigma) & \end{array}$$

As Kenstowicz (1994) argues, (see also Halle 1995; Halle and Idsardi 1998), ordering these rules serially is problematic. Ordering lengthening before shortening produces the wrong forms (shortening would undo any changes made by lengthening; see (38a) below), and ordering shortening before lengthening produces a "Duke of York" effect in which shortening makes a change that is then undone (see (38b) below; and see Halle and Idsardi (1998) for discussion).

(38) Non-disjunctive ordering of shortening and lengthening

a.

	/radical/	/radial/
Length.	—	(rādi)al
Short.	(rădi)cal	(rădi)al
Output	rădical	*rădial

b.

	/radical/	/radial/
Short.	(rădi)cal	(rădi)al
Length.	—	(rādi)al
Output	rădical	rādial

Instead, Kenstowicz (1994) notes, the structural description of the lengthening rule in (37b) is strictly more specific than that of (37a). This is exactly the situation in which the EC states that shortening and lengthening should be disjunctively ordered, with lengthening (37b) applying in the specific situations in which its environment is met, and shortening (37a) elsewhere. The derivations in (39) illustrate.

(39)

	/radical/	/radial/
Shortening	(rădi)cal	Blocked by EC
Lengthening	Blocked by EC	(rādi)al
Output	rădical	rādial

This produces the correct forms. However, as pointed out by Baković (2006), the EC is an external principle independent of the operation of the SPE rule system. In other words, we must stipulate the additional mechanism of the EC in order to obtain a satisfactory analysis of "do X unless Y" generalizations like shortening and lengthening in English. With BMRSs, this disjunctive behavior instead falls out of how the grammars are evaluated.

4.3 The typology of N$\underset{\circ}{\text{C}}$ effects

Our third case study is the well-known typology of N$\underset{\circ}{\text{C}}$ effects, in which different languages apply different repair strategies to avoid a sequence of a nasal followed by a voiceless consonant. The separation of markedness and faithfulness in OT grammars captures the generalization missed in rule-based grammars, which is that the single markedness constraint in (40) is responsible for the diverse set of repairs.

(40) *N$\underset{\circ}{\text{C}}$
No nasal/voiceless obstruent sequences (Pater, 2004, p. 273, (3))

Pater (2004) demonstrates how the typology of repairs is captured through permutations of ranking various faithfulness constraints with *N$\underset{\circ}{\text{C}}$. In rule-based analyses, each language has its own rule (depending on the chosen repair) and this set of rules just happens to share a structural description.

In this section we demonstrate how the licensing and blocking structures in a BMRS analysis handle the conspiratorial nature of a constraint like *N$\underset{\circ}{\text{C}}$. As in the previous analyses, the placement of these structures is constrained by the inherent logic of BMRSs.

We first establish nasal place assimilation, which plays a role in all of these grammars. These definitions hold for all of the analyses in this subsection.[12]

(41)
$$\begin{aligned}
[\text{lab}]_{\text{o}}(x) &= \texttt{if } \underline{[\text{nas}]}[\text{lab}](x) \texttt{ then } \top \texttt{ else } [\text{lab}](x)\\
[\text{cor}]_{\text{o}}(x) &= \texttt{if } \underline{[\text{nas}]}[\text{cor}](x) \texttt{ then } \top \texttt{ else } [\text{cor}](x)\\
[\text{dor}]_{\text{o}}(x) &= \texttt{if } \underline{[\text{nas}]}[\text{dor}](x) \texttt{ then } \top \texttt{ else } [\text{dor}](x)
\end{aligned}$$

Next we define the following two predicates, which identify the two segments in a N$\underset{\circ}{\text{C}}$ structure. The reason two predicates are needed is because the range of repair strategies vary in whether they target the nasal or the obstruent in these structures.

(42)
a. $\underline{\text{N}}\underset{\circ}{\text{C}}(x) = \texttt{if } [\text{nas}](x) \texttt{ then } \begin{bmatrix} -\text{son} \\ -\text{voi} \end{bmatrix}(s(x)) \texttt{ else } \bot$

b. $\text{N}\underline{\underset{\circ}{\text{C}}}(x) = \texttt{if } \begin{bmatrix} -\text{son} \\ -\text{voi} \end{bmatrix}(x) \texttt{ then } [\text{nas}](p(x)) \texttt{ else } \bot$

We can then examine the various effects of these predicates as licensing and blocking structures for the definitions of the predicates $\texttt{out}(x)$, $[\text{nas}]_{\text{o}}(x)$, and $[\text{voi}]_{\text{o}}(x)$ determining whether x is output, nasal, and voiced, respectively. Note that here, we have defined $\underline{\text{N}}\underset{\circ}{\text{C}}(x)$ and $\text{N}\underline{\underset{\circ}{\text{C}}}(x)$ as input structures, instead of output structures like Pater's *N$\underset{\circ}{\text{C}}$ markedness constraint. It is possible to define them in terms of the output by using recursive definitions, but for simplicity we leave them defined as input structures. Regardless, the definitions in

[12] A more succinct statement would be possible with a three-valued (i.e., non-Boolean) predicate $[\text{place}](x)$ that takes one of the values in $\{\texttt{lab}, \texttt{cor}, \texttt{dor}\}$:

$$[\text{place}](x) = \texttt{if } \underline{[\text{nas}]}\text{C}(x) \texttt{ then } [\text{place}](s(x)) \texttt{ else } [\text{place}](x)$$

This deviates slightly from the overall structure we established in §3.1, in that x takes on the value of $[\text{place}](s(x))$ rather than a boolean value. However, this is still valid BMRS syntax.

(42) capture all of the relevant generalizations.

We will show that, like Pater (2004)'s factorial typology of $*\mathrm{N}\underset{\circ}{\mathrm{C}}$ and various faithfulness constraints, this describes the typology of nasal-voiceless consonant repairs. Furthermore, the following principle, similar to the SSOP in the previous section, governs where these structures can be placed. We will use this principle to explain why certain logically possible repairs do not occur.

(43) Meaningful Structures in BMRSs
We consider BMRS transductions whose licensing and blocking structures are *meaningful.* In a BMRS of the structure

$$\texttt{if } X \texttt{ then } B \texttt{ else } Y$$

X is meaningful iff neither of the following conditions holds. B is $\top$ (i.e., X is a licensing structure) and X implies Y, or B is $\bot$ (i.e., X is a blocking structure) and X implies $\neg Y$.

Here, X implies Y means that whenever X is true, Y is true. For example, the default for $\texttt{out}(x)$ is $\top$. Thus, unless some other blocking structure intervenes in the order, using either $\underline{\mathrm{N}}\underset{\circ}{\mathrm{C}}(x)$ or $\mathrm{N}\underline{\underset{\circ}{\mathrm{C}}}(x)$ as a licensing structure is not meaningful, because $\texttt{out}(x)$ in this case would be evaluated to $\top$ anyway. To see this, in the definition of $\texttt{out}(x)$ in (44a), $\underline{\mathrm{N}}\underset{\circ}{\mathrm{C}}(x)$ is a licensing structure. However, it also implies $\top$—whenever $\underline{\mathrm{N}}\underset{\circ}{\mathrm{C}}(x)$ is true, $\top$ is true (somewhat vacuously, as $\top$ is always true). Therefore by (43) $\underline{\mathrm{N}}\underset{\circ}{\mathrm{C}}(x)$ is not meaningful.

(44) a. $\texttt{out}(x) = \texttt{if } \underline{\mathrm{N}}\underset{\circ}{\mathrm{C}}(x) \texttt{ then } \top \texttt{ else}$
$\quad\top$

b. $\texttt{out}(x) = \texttt{if } \underline{\mathrm{N}}\underset{\circ}{\mathrm{C}}(x) \texttt{ then } \top \texttt{ else}$
$\quad\texttt{if } [\mathrm{nas}](x) \texttt{ then } \bot \texttt{ else}$
$\quad\top$

However, if we add another intervening blocking structure, such as $[\mathrm{nas}](x)$ in (44b), then $\underline{\mathrm{N}}\underset{\circ}{\mathrm{C}}(x)$ is now meaningful, because it does not imply all of $\texttt{if } [\mathrm{nas}](x) \texttt{ then } \bot \texttt{ else } \top$.

Thus, $\underline{\mathrm{N}}\underset{\circ}{\mathrm{C}}(x)$ on its own cannot function as a licensing structure for $\texttt{out}(x)$, and for similar reasons neither can $\mathrm{N}\underline{\underset{\circ}{\mathrm{C}}}(x)$. However, they can act as a blocking structure for $\texttt{out}(x)$, which results in deletion. Deletion of the consonant results from using $\mathrm{N}\underline{\underset{\circ}{\mathrm{C}}}(x)$. This is attested in Indonesian (Onn, 1980; Herbert, 1986; Pater, 2004), in which the prefix /məN/ causes alternations in roots beginning with a voiceless obstruent. As shown below, this initial obstruent deletes and the nasal takes on its place of articulation.[13]

[13] As noted in Pater (2004), this alternation only occurs at morpheme boundaries; for example /əmpat/ 'four' surfaces faithfully as [əmpat], not as *[əmat]. Root faithfulness in this case can be implemented by defining a structure $\texttt{rt-internal}(x)$ and assigning it as a higher-ranked licensing structure for $\texttt{out}(x)$ as below.

i. $\texttt{rt-internal}(x) = \texttt{if } +(p(x)) \texttt{ then } \bot \texttt{ else } \texttt{root}(x)$

ii. $\texttt{out}(x) = \texttt{if } \texttt{rt-internal}(x) \texttt{ then } \top \texttt{ else}$
$\quad\texttt{if } \mathrm{N}\underline{\underset{\circ}{\mathrm{C}}} \texttt{ then } \bot \texttt{ else}$
$\quad\top$

(45) Indonesian nasal substitution (Onn, 1980)
/məNpilih/ ↦ [məmilih], 'to choose, vote'

(46)
$$\begin{aligned}\texttt{out}(x) &= \texttt{if } \mathrm{N}\underline{\underset{\circ}{\mathrm{C}}}(x) \texttt{ then } \bot \texttt{ else } \top\\ [\mathrm{nas}]_{\mathrm{o}}(x) &= [\mathrm{nas}](x)\\ [\mathrm{voi}]_{\mathrm{o}}(x) &= [\mathrm{voi}](x)\end{aligned}$$

Note that the place assimilation is accomplished through the definitions in (41). This applies even in the case of deletion of the consonant because they are defined in terms of the input. Note also that this analysis is similar to the derivational analysis in which both assimilation and deletion occur (as in, e.g., Onn 1980), rather than Pater (2004)'s Linearity-based fusion analysis. This is an artifact of our abstraction away from changes to the order of segments, though see below for arguments against this explanation for Indonesian. Furthermore, this analysis does not occur in "two steps" as Pater points out (p. 272) of analyses like that of Onn—the BMRS is a description of a single mapping from input to output.

Deletion of the nasal instead is achieved by using $\underline{\mathrm{N}}\underset{\circ}{\mathrm{C}}(x)$ as a blocking structure for $\texttt{out}(x)$. This is attested in, for example, Swahili (Choti, 2015).

(47) Nasal deletion in Swahili (Choti, 2015)

/Nbaya/	↦	[mbaya]	'Class 9/10-bad'
/Nkubwa/	↦	[kubwa]	'Class 9/10-big'

(48)
$$\begin{aligned}\texttt{out}(x) &= \texttt{if } \underline{\mathrm{N}}\underset{\circ}{\mathrm{C}}(x) \texttt{ then } \bot \texttt{ else } \top\\ [\mathrm{nas}]_{\mathrm{o}}(x) &= [\mathrm{nas}](x)\\ [\mathrm{voi}]_{\mathrm{o}}(x) &= [\mathrm{voi}](x)\end{aligned}$$

Changes to voicing and nasality then result from using $\mathrm{N}\underline{\underset{\circ}{\mathrm{C}}}(x)$ or $\underline{\mathrm{N}}\underset{\circ}{\mathrm{C}}(x)$ as licensing or blocking structures for $[\mathrm{voi}]_{\mathrm{o}}(x)$ and $[\mathrm{nas}]_{\mathrm{o}}(x)$, respectively. We will go through these various options in turn.

First, when $\mathrm{N}\underline{\underset{\circ}{\mathrm{C}}}(x)$ appears as a licensing structure in $[\mathrm{voi}]_{\mathrm{o}}(x)$, as in (49), the result is postnasal voicing. This occurs in Puyo Pongo Quechua (Orr, 1962), as shown in (50).

(49)
$$\begin{aligned}\texttt{out}(x) &= \top\\ [\mathrm{nas}]_{\mathrm{o}}(x) &= [\mathrm{nas}](x)\\ [\mathrm{voi}]_{\mathrm{o}}(x) &= \texttt{if } \mathrm{N}\underline{\underset{\circ}{\mathrm{C}}}(x) \texttt{ then } \top \texttt{ else}\\ &\quad [\mathrm{voi}](x)\end{aligned}$$

(50) Puyo Pungo Quechua post-nasal voicing (Orr, 1962)
/kampa/ ↦ [kamba]

$\mathrm{N}\underline{\underset{\circ}{\mathrm{C}}}(x)$ cannot, however, serve as a blocking structure for $[\mathrm{voi}]_{\mathrm{o}}(x)$, because it would not be meaningful according to definition (43). This is because $\mathrm{N}\underline{\underset{\circ}{\mathrm{C}}}(x)$ identifies a voiceless segment and therefore it does imply $\neg[\mathrm{voi}](x)$.

We now consider $\underline{\mathrm{N}}\underset{\circ}{\mathrm{C}}(x)$ as a licensing structure for $[\mathrm{voi}]_{\mathrm{o}}(x)$. First we recognize that in most languages, nasality implies voicing. This can be captured by adding $[\mathrm{nas}](x)$ as a licensing structure for the definition of $[\mathrm{voi}]_{\mathrm{o}}(x)$:

Above, $+(p(x))$ indicates that x follows a morpheme boundary, and $\texttt{root}(x)$ indicates that x is in a root.

(51) $$[\text{voi}]_{\text{o}}(x) = \texttt{if } [\text{nas}]_{\text{o}}(x) \texttt{ then } \top \texttt{ else} \\ [\text{voi}](x)$$

Given (51), $\underline{\mathrm{N}}\underset{\circ}{\mathrm{C}}(x)$ is either not meaningful as a licensing structure or it violates the SSOP.

(52) a. $$[\text{voi}]_{\text{o}}(x) = \texttt{if } \underline{\mathrm{N}}\underset{\circ}{\mathrm{C}}(x) \texttt{ then } \top \texttt{ else} \\ \texttt{if } [\text{nas}]_{\text{o}}(x) \texttt{ then } \top \texttt{ else} \\ [\text{voi}](x)$$

b. $$[\text{voi}]_{\text{o}}(x) = \texttt{if } [\text{nas}]_{\text{o}}(x) \texttt{ then } \top \texttt{ else} \\ \texttt{if } \underline{\mathrm{N}}\underset{\circ}{\mathrm{C}}(x) \texttt{ then } \top \texttt{ else} \\ [\text{voi}](x)$$

Consider first (52a). Whenever $\underline{\mathrm{N}}\underset{\circ}{\mathrm{C}}(x)$ is true, $[\text{nas}]_{\text{o}}(x)$ is also true (unless it is blocked in that exact configuration). Therefore, $\underline{\mathrm{N}}\underset{\circ}{\mathrm{C}}(x)$ implies $\texttt{if } [\text{nas}]_{\text{o}}(x) \texttt{ then } \top \texttt{ else } [\text{voi}](x)$, in violation of (43). So (52a) is excluded by being meaningless. Likewise, the reverse ordering in (52b) violates the SSOP, because $[\text{nas}](x)$ is a proper substructure of $\underline{\mathrm{N}}\underset{\circ}{\mathrm{C}}(x)$. Thus, either ordering of $\underline{\mathrm{N}}\underset{\circ}{\mathrm{C}}(x)$ and $[\text{nas}](x)$ as licensing structures is forbidden by these two principles.

However, $\underline{\mathrm{N}}\underset{\circ}{\mathrm{C}}(x)$ can appear as a blocking structure for $[\text{voi}]_{\text{o}}(x)$, causing devoicing of the nasal.

(53) Nasal devoicing
/ampa/ ↦ [am̥pa]

(54) $$\begin{array}{lcl} \texttt{out}(x) & = & \top \\ [\text{nas}]_{\text{o}}(x) & = & [\text{nas}](x) \\ [\text{voi}]_{\text{o}}(x) & = & \texttt{if } \underline{\mathrm{N}}\underset{\circ}{\mathrm{C}}(x) \texttt{ then } \bot \texttt{ else } [\text{voi}](x) \end{array}$$

Whether this repair is attested is controversial, as noted by Pater (p. 285, fn. 1). Herbert (1986) cites Ndonga (Viljoen and Amakali, 1978) and Pokomo (Hinnebusch, 1975) as having nasal devoicing in this context, though Huffman and Hinnebusch (1998) argue that this is a phonetic effect in Pokomo.

Finally, we consider $\underline{\mathrm{N}}\underset{\circ}{\mathrm{C}}(x)$ and $\mathrm{N}\underline{\underset{\circ}{\mathrm{C}}}(x)$ as licensing and blocking structures for $[\text{nas}]_{\text{o}}(x)$. As a licensing structure for $[\text{nas}]_{\text{o}}(x)$, $\mathrm{N}\underline{\underset{\circ}{\mathrm{C}}}(x)$ predicts nasalization of the obstruent. This occurs in Konjo (Friberg and Friberg, 1991, pp. 84–86) as shown in (55) and (56).

(55) Nasalization in Konjo (Friberg and Friberg, 1991)

/aŋpinahaŋ/	↦	[əmminahaŋ]	'to follow'
/aŋkanre/	↦	[əŋŋanre]	'to eat'

(56) $$\begin{array}{lcl} \texttt{out}(x) & = & \top \\ [\text{nas}]_{\text{o}}(x) & = & \texttt{if } \mathrm{N}\underline{\underset{\circ}{\mathrm{C}}}(x) \texttt{ then } \top \texttt{ else} \\ & & [\text{nas}](x) \\ [\text{voi}]_{\text{o}}(x) & = & \texttt{if } [\text{nas}]_{\text{o}}(x) \texttt{ then } \top \texttt{ else} \\ & & [\text{voi}](x) \end{array}$$

Note that the definition of $[\text{voi}]_{\text{o}}(x)$ includes the fact that nasality implies voicing, producing a voiced nasal.

As a blocking structure for $[\text{nas}]_{\text{o}}(x)$, $\mathrm{N}\underline{\underset{\circ}{\mathrm{C}}}(x)$ prevents a voiceless obstruent from nasalizing. On its face, this seems redundant, as obstruents are typically not nasal (though see Durvasula 2010). This implication can be captured by using $[-\text{son}](x)$ as a blocking structure for $[\text{nas}]_\text{o}(x)$.

(57) a. $[-\text{son}]_\text{o}(x) \;=\; \texttt{if}\ [\text{son}]_\text{o}(x)\ \texttt{then}\ \bot\ \texttt{else}\ \top$

b. $[\text{nas}]_\text{o}(x) \;=\; \texttt{if}\ [-\text{son}]_\text{o}(x)\ \texttt{then}\ \bot\ \texttt{else}$
$[\text{nas}](x)$

But, whenever this implication is present for $[\text{nas}]_\text{o}(x)$, as a blocking structure, $\text{N}\underline{\text{C̥}}(x)$ is either meaningless or is blocked by the SSOP, for reasons parallel to those given above with $\underline{\text{N}}\text{C̥}(x)$ when nasality implied voicing.

(58) a. $[\text{nas}]_\text{o}(x) \;=\; \texttt{if}\ \text{N}\underline{\text{C̥}}(x)\ \texttt{then}\ \bot\ \texttt{else}$
$\texttt{if}\ [-\text{son}]_\text{o}(x)\ \texttt{then}\ \bot\ \texttt{else}$
$[\text{nas}](x)$

b. $[\text{nas}]_\text{o}(x) \;=\; \texttt{if}\ [-\text{son}]_\text{o}(x)\ \texttt{then}\ \bot\ \texttt{else}$
$\texttt{if}\ \text{N}\underline{\text{C̥}}(x)\ \texttt{then}\ \top\ \texttt{else}$
$[\text{nas}](x)$

Turning to $\underline{\text{N}}\text{C̥}(x)$, as a blocking structure for $[\text{nas}]_\text{o}(x)$ it causes denasalization, as in Mandar (Mills, 1975).

(59) Mandar gemination (Mills, 1975)
/mantunu/ $\mapsto$ [mattunu], ‘to burn’

(60) $\texttt{out}(x) \;=\; \top$
$[\text{nas}]_\text{o}(x) \;=\; \texttt{if}\ \underline{\text{N}}\text{C̥}(x)\ \texttt{then}\ \bot\ \texttt{else}\ [\text{nas}](x)$
$[\text{voi}]_\text{o}(x) \;=\; \texttt{if}\ [-\text{voi}](p(x))\ \texttt{then}\ \bot\ \texttt{else}$
$[\text{voi}](x)$

As a blocking structure for $[\text{nas}]_\text{o}(x)$ in (60), $\underline{\text{N}}\text{C̥}(x)$ prevents the N in a NC̥ sequence from surfacing as nasal. (While it is unclear from the discussion in Mills (1975) why exactly the nasal is also devoiced, we provisionally analyze this with $[-\text{voi}](p(x))$ as a blocking structure for $[\text{voi}]_\text{o}$.)

Finally, as a licensing structure for $[\text{nas}]_\text{o}$, $\underline{\text{N}}\text{C̥}(x)$ is meaningless. This is because whenever $\underline{\text{N}}\text{C̥}(x)$ is true, then clearly the default predicate $[\text{nas}](x)$ is also true.

Table 10 thus summarizes the processes predicted by varying $\underline{\text{N}}\text{C̥}(x)$ and $\text{N}\underline{\text{C̥}}(x)$ as licensing and blocking structures for $[\text{nas}]_\text{o}(x)$ and $[\text{voi}]_\text{o}(x)$.

The two structures can also appear in different places in the grammar. This is attested in Umbundu, in which roots beginning with a voiceless obstruent surface with an initial, single nasal of the same place of articulation. A systematic exception is with fricatives, which never nasalize.

(61) Umbundu (Schadeberg, 1982)

/N+tuma/	$\mapsto$	[numa]	‘I send’
/N+seva/	$\mapsto$	[seva]	‘I cook’

	$\underline{\mathrm{N}}\underset{\circ}{\mathrm{C}}$		$\mathrm{N}\underline{\underset{\circ}{\mathrm{C}}}$	
	licensing	blocking	licensing	blocking
`out`	(meaningless)	nasal deletion (Swahili)	(meaningless)	consonant deletion (Indonesian)
$[\mathrm{nas}]_{\mathrm{o}}$	(meaningless)	denasalization (Mandar)	nasalization (Konjo)	(meaningless when $[-\mathrm{son}]\rightarrow\neg[\mathrm{nas}]$)
$[\mathrm{voi}]_{\mathrm{o}}$	(meaningless when $[\mathrm{nas}]\rightarrow[\mathrm{voi}]$)	nasal devoicing (Ndonga (?) , Pokomo (?))	voicing (Quechua)	(meaningless)

Table 10: Summary of *$\mathrm{N}\underset{\circ}{\mathrm{C}}$ typology

This can be captured by placing $\underline{\mathrm{N}}\underset{\circ}{\mathrm{C}}(x)$ as a blocking structure for $\mathtt{out}(x)$ *and* $\mathrm{N}\underline{\underset{\circ}{\mathrm{C}}}(x)$ as a licensing structure for [nas], as shown below in (62).

(62)
$$\begin{aligned}
\mathtt{out}(x) &= \mathtt{if}\ \underline{\mathrm{N}}\underset{\circ}{\mathrm{C}}(x)\ \mathtt{then}\ \bot\ \mathtt{else} \\
&\quad \top \\
[\mathrm{nas}]_{\mathrm{o}}(x) &= \mathtt{if}\ [\mathrm{cont}](x)\ \mathtt{then}\ \bot\ \mathtt{else} \\
&\quad \mathtt{if}\ \mathrm{N}\underline{\underset{\circ}{\mathrm{C}}}(x)\ \mathtt{then}\ \top\ \mathtt{else} \\
&\quad [\mathrm{nas}](x) \\
[\mathrm{voi}]_{\mathrm{o}}(x) &= [\mathrm{voi}](x)
\end{aligned}$$

Thus, in /N+tuma/ ‘I send’, the initial /N/ returns $\bot$ for $\mathtt{out}(x)$ and is thus deleted; additionally, the /t/ is also nasalized. This yields the output [numa]. The fact that non-continuent consonants exceptionally do *not* nasalize after a nasal is captured in (62) by placing $\mathrm{N}\underline{\underset{\circ}{\mathrm{C}}}(x)$ as a licensing structure for $[\mathrm{nas}]_{\mathrm{o}}(x)$. The complication that continuants do not nasalize is directly captured by placing the blocking structure $[\mathrm{cont}](x)$ above the licensing structure $\mathrm{N}\underline{\underset{\circ}{\mathrm{C}}}(x)$. Thus, for /N-seva/ ‘I cook’, the nasal deletes, but the continuant does not nasalize, yielding [seva].

Thus, a BMRS theory of phonology can derive typological generalizations about how a marked structure can be repaired by varying this structure as a blocking or licensing structure for different output predicates. Of course, as in OT, there are some “too many solutions”-type problems. As Pater points out, epenthesis is never attested as a repair for $\mathrm{N}\underset{\circ}{\mathrm{C}}$ sequences, though it is predicted by his typology of constraints. While we have not here discussed how to implement epenthesis in BMRSs, it is reasonable to suppose that once we do, BMRSs will similarly predict this unattested repair. However, by not appealing to changing the order of segments, the “too many solutions” problem is in some ways less stark than Pater’s OT typology. For example, one prediction of ranking *$\mathrm{N}\underset{\circ}{\mathrm{C}}$ above Linearity is metathesis of the nasal and consonant: /$\mathrm{N}\underset{\circ}{\mathrm{C}}$/→[$\underset{\circ}{\mathrm{C}}$N]. Other, more dramatic (and non-regular) effects of Linearity in OT are discussed in Wilson (2003) and Lamont (2018). Further work with BMRSs will of course have to deal with phenomena like metathesis, but there are computational analyses of metathesis on which such analyses can be built (Chandlee and Heinz, 2012; Chandlee, 2017). Regardless of how it is implemented, BMRSs are guaranteed to be more constrained than OT Linearity analyses because they are subregular.

5 Discussion

The analyses presented in the previous sections have demonstrated several of the strengths of the BMRS formalism: namely, they can capture the interaction of multiple generalizations in a local way, which maintains the desirable computational restrictiveness of previous implementations of subregularity but with representations that are more intuitive from a phonological perspective. In this section we discuss a few potential extensions of this approach to phonological analysis.

The case studies presented in this paper demonstrate how BMRSs can capture multiple generalizations directly, rather than representing each generalization separately (i.e., as a separate rule or set of constraints) and then combining them (i.e., via ordering or ranking). None of the examples, however, incorporated the entire phonology of the language, raising the question of how that task might be accomplished. In other words, is a combination operation of some type still needed, and if so, what is that operation?

The most apparent option for combining functions is function composition, as was used in the early finite-state models of SPE grammars (Johnson, 1972; Kaplan and Kay, 1981, 1994) and OT grammars (e.g., Karttunen, 1993; Frank and Satta, 1998; Gerdemann and Hulden, 2012; Riggle, 2004, among others). Composition can be studied as a syntactic operation that combines two BMRS transductions such that the function described by this combined transduction is equal to the composition of the two functions. This has yet to be worked out in formal detail, but there is a clear path forward for doing so. Briefly, to compose two BMRS transductions A and B, we combine both systems of equations and replace the *input* feature predicates of B with the *output* feature predicates of A. This implements the notion that the output of A becomes the input of B.

A second question involves long-distance interactions, such as long-distance nasal harmony in Kikongo (Ao, 1991; Odden, 1994; Rose and Walker, 2004). In Kikongo, a suffix liquid becomes nasal following a nasal in the root, no matter how far to the left it is.

(63) Kikongo (Ao, 1991; Odden, 1994)

/sakid-ila/	[sakid-ila]	'congratulate for'
/ku-toot-ila/	[ku-toot-ila]	'to harvest for'
/ku-kin-ila/	[ku-kin-ina]	'to dance for'
/ku-dumuk-ila/	[ku-dumuk-ina]	'to jump for'
/ku-dumuk-is-ila/	[ku-dumuk-is-ina]	'to make jump for'

This can be captured by the following BMRS system of equations. First, we need a recursively defined predicate follows-[nas](x) as follows.

(64)
$$\text{follows-[nas]}(x) = \texttt{if } \#(x) \texttt{ then } \bot \texttt{ else } \texttt{if } [\text{nas}](p(x)) \texttt{ then } \top \texttt{ else } \text{follows-[nas]}(p(x))$$

Note that follows-[nas](x) is explicitly recursive as it refers to itself: follows-[nas](x) is false if x is the word boundary, true if the immediate predecessor of x is nasal, or true if follows-[nas](x) is true of x's predecessor. Thus, it will recurse backwards in a word unless it finds a nasal segment (and then returns true) or finds the initial word boundary (and then returns false).

We can use this to define the following predicate $\text{N...}\underline{\text{L}}(x)$ which we can then use as a licensing structure for nasality.

(65) $\text{N...}\underline{\text{L}}(x) = \texttt{ if } [\text{lat}](x) \texttt{ then } \text{follows-}[\text{nas}](x) \texttt{ else } \bot$

The full BMRS transduction that captures this transformation would then be as follows.

(66)
$$\begin{aligned}
\text{follows-}[\text{nas}](x) &= \texttt{if } \#(x) \texttt{ then } \bot \texttt{ else} \\
&\quad\ \texttt{if } [\text{nas}](p(x)) \texttt{ then } \top \texttt{ else} \\
&\quad\ \text{follows-}[\text{nas}](p(x)) \\
[\text{nas}]_\text{o}(x) &= \texttt{if } \text{N...}\underline{\text{L}}(x) \texttt{ then } \top \texttt{ else} \\
&\quad\ [\text{nas}](x) \\
[\text{lat}]_\text{o}(x) &= \texttt{if } [\text{nas}]_\text{o}(x) \texttt{ then } \bot \texttt{ else} \\
&\quad\ [\text{lat}](x)
\end{aligned}$$

Note that, because it is explicitly recursive, follows-[nas](x) must be part of the system of equations. This is in contrast to the licensing and blocking structures defined in previous systems of equations, which only make reference to existing input or output predicates. This does not increase the expressive power of the BMRS formalism; these transformations are still subsequential (Bhaskar et al., 2020). However, it does highlight an interesting difference between output-local transformations—like the unbounded spreading definition in (16)—and truly long-distance transformations in Kikongo, namely that there is an explicitly recursive predicate that is part of the BMRS system of equations but is not an output predicate. This is a point that warrants further work.

Finally, recursive schemes are flexible enough that they can be defined to hew more closely to phonological representation. For example, we can define functions that return not boolean values, but featural functions $\mathcal{I} = \{[\text{F}](t), [\text{G}](t), ..., [\text{Z}](t), \#(t)\}$ that instead take values in $\{+, -, 0\}$.

For example, a word-final devoicing definition would work as in (67).

(67) $[\text{voi}]_\text{o}(x) = \texttt{if } \underline{\begin{bmatrix} -\text{son} \\ +\text{voi} \end{bmatrix}} \#(x) \texttt{ then } - \texttt{ else } [\text{voi}](x)$

This states: the value of [$\pm$voi] for x in the output is $-$ if it is a voiced obstruent, otherwise it takes the value x has for [$\pm$voi] in the input (i.e., it is output faithfully).

However, notice that in statements of the form

$$\texttt{if } E_1 \texttt{ then } E_2 \texttt{ else } E_3$$

the expression E_1 must evaluate to a boolean value—otherwise the if/then/else logical control cannot function. To deal with this, we can add the (standard; see, e.g. Enderton 1972) single boolean predicate

$$E_1 = E_2$$

to the syntax of BMRSs, and then require (as is also standard; Moschovakis 2019) that for a well-formed $\texttt{if } E_1 \texttt{ then } E_2 \texttt{ else } E_3$ statement E_1 must be boolean. Further boolean predicates of the form [+F] can then be defined as

$$[+\text{F}](x) = \texttt{if}\ [\text{F}](x) = +\ \texttt{then}\ \top\ \texttt{else}\ \bot$$

Note that this would change the details of the definitions of predicates such as $\begin{bmatrix} -\text{son} \\ +\text{voi} \end{bmatrix}(x)$, but the BMRS definitions of maps (such as the final devoicing map in §3.1.3) would not change at all. In other words, the logic of transductions defined with recursive schemes is not dependent on how the user-defined predicates are defined; it remains constant as long as the truth values of the user-defined predicates remain constant. Therefore, while considering predicates that return feature values instead of boolean values would make the formalism no longer *boolean*, it would still be *monadic*, and thus there would be no change in the expressiveness of the formalism.

Extending recursive schemes to representations that explicitly include binary relations, such as association in autosegmental representations (Goldsmith, 1976; Coleman and Local, 1991) or dominance in the prosodic hierarchy (Selkirk, 1980; Nespor and Vogel, 1986), would necessitate relaxing the restriction to monadic predicates and functions. The consequences of this remain to be explored; however, recent research on logical transductions in such representations (Strother-Garcia, 2017; Koser et al., 2019; Chandlee and Jardine, 2019) can serve as a starting point for this work.

6 Conclusion

Generative phonology, like all formal linguistic theories, aims to understand the nature of the computations that underlie patterns in natural language. This paper has illustrated how analyses with boolean monadic recursive schemes directly capture phonological generalizations, while adhering to results showing the computationally restrictive nature of phonological typology. While many open questions remain, the examples in this paper demonstrate that this formalism shows great promise as a theory of the computational nature of phonological transformations.

References

Ao, B. (1991). Kikongo nasal harmony and context-sensitive underspecification. *Linguistic Inquiry*, 22(2):193–196.

Baker, M. (2009). Language universals: Abstract not mythological. *Behavior and Brain Sciences*, 32:448–449.

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

Baković, E. (2006). Elsewhere effects in Optimality Theory. In Baković, E., Ito, J., and McCarthy, J. J., editors, *Wondering at the Natural Fecundity of Things: Essays in Honor of Alan Prince*, pages 23–70. University of California eScholarship Repository and BookSurge Publishing.

Bane, M. and Riggle, J. (2010). When more choice means less freedom: a note on candidate sets and typologies. ROA#1108. Available at http://roa.rutgers.edu/.

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., Martín-Vide, C., Shapira, D., and Zandron, C., editors, *Language and Automata Theory and Applications - LATA 2020*.

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

Chandlee, J. and Heinz, J. (2012). Bounded copying is subsequential: Implications for metathesis and reduplication. In *Proceedings of the 12th Meeting of the ACL Special Interest Group on Computational Morphology and Phonology*, pages 42–51, Montreal, Canada. Association for Computational Linguistics.

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

Chandlee, J., Heinz, J., and Jardine, A. (2018). Input strictly local opaque maps. *Phonology*, 35:1–35.

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

Chomsky, N. (1957). *Aspects of the theory of syntax.* The Hague: Mouton.

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

Choti, J. (2015). Phonological asymmetries of bantu nasal prefixes. In Kramer, R., Zsiga, E. C., and Boyer, O. T., editors, *Selected Proceedings of the 44th Annual Conference on African Linguistics*, pages 37–51. Somerville, MA: Cascadilla Press.

Coleman, J. and Local, J. (1991). The "No Crossing Constraint" in autosegmental phonology. *Linguistics and Philosophy*, 14:295–338.

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

Derbyshire, D. C. (1985). *Hixkaryana and Linguistic Typology.* Dallas: Summer Institute of Linguistics.

Durvasula, K. (2010). *Understanding Nasality.* PhD thesis, University of Delaware.

Enderton, H. (1972). *A mathematical introduction to logic.* Academic Press.

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

Frank, R. and Satta, G. (1998). Optimality Theory and the generative complexity of constraint violability. *Computational linguistics*, 24:307–315.

Friberg, T. and Friberg, B. (1991). Notes on Konjo phonology. In Sneddon, J. N., editor, *Studies in Sulawesi Linguistics, Part II*, volume 33 of *NUSA Series*, pages 71–117. Jakarta, Indonesia: NUSA – Linguistic Studies of Indonesian and Other Languages in Indonesia.

Gerdemann, D. and Hulden, M. (2012). Practical finite state Optimality Theory. In *Proceedings of the 10th International Workshop on FSMNLP*, pages 10–19. ACL.

Gold, M. E. (1967). Language identification in the limit. *Information and Control*, 10:447–474.

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

Gordon, M. (2002). A factorial typology of quantity-insensitive stress. *Natural Language and Linguistic Theory*, 20:491–552.

Halle, M. (1995). Comments on Burzio (1995). *Glot International*, 1:27–28.

Halle, M. and Idsardi, W. J. (1998). A response to Alan Prince's letter. *Glot International*, 3:1 & 22.

Halle, M. and Idsardi, W. J. (2000). Stress and length in Hixkaryana. *The Linguistic Review*, 17:199–218.

Hayes, B. (1995). *Metrical stress theory*. Chicago: The University of Chicago Press.

Heinz, J. (2009). On the role of locality in learning stress patterns. *Phonology*, 26:303–351.

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. (forthcoming). Doing computational phonology. Oxford University Press.

Heinz, J. and Lai, R. (2013). Vowel harmony and subsequentiality. In Kornai, A. and Kuhlmann, M., editors, *Proceedings of the 13th Meeting on Mathematics of Language*, Sofia, Bulgaria.

Herbert, R. K. (1986). *Language Universals, Markedness Theory, and Natural Phonetic Processes*. Berlin: Mouton de Gruyter.

Hinnebusch, T. J. (1975). A reconstructed chronology of loss: Swahili class 9/10. In Herbert, R. K., editor, *Proceedings of the Sixth Conference on African Linguistics*, pages 32–41.

Huffman, M. and Hinnebusch, T. (1998). The phonetic nature of 'voiceless' nasals in Pokomo: Implications for sound change. *Journal of African Languages and Linguistics*, 19:1–19.

Immerman, N. (1980). *First-order expressibility as a New Complexity Measure*. PhD thesis, Cornell University.

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

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

Jardine, A. (2017b). On the logical complexity of autosegmental representations. In Kanazawa, M., de Groote, P., and Sadrzadeh, M., editors, *Proceedings of the 15th Meeting on the Mathematics of Language*, pages 22–35, London, UK. Association for Computational Linguistics.

Jardine, A. (2019). The expressivity of autosegmental grammars. *Journal of Logic, Language, and Information*, 28:9–54.

Jardine, A., Chandlee, J., Eyraud, R., and Heinz, J. (2014). Very efficient learning of structured classes of subsequential functions from positive data. In *Proceedings of the 12th International Conference on Grammatical Inference (ICGI 2014)*, JMLR Workshop Proceedings, pages 94–108.

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

Kager, R. (1999). *Optimality Theory.* Cambridge University Press.

Kager, R. (2007). Feet and metrical stress. In de Lacy, P., editor, *The Cambridge Handbook of Phonological Theory*, chapter 9, pages 195–227. Cambridge University Press.

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

Kaplan, R. M. and Kay, M. (1981). Phonological rules and finite-state transducers. In *Linguistic Society of America Meeting Handbook, Fifty-Sixth Annual Meeting.* New York: LSA. Abstract.

Karttunen, L. (1993). Finite-state constraints. In Goldsmith, J., editor, *The Last Phonological Rule*, pages 173–194. Chicago: The University of Chicago Press.

Kenstowicz, M. (1994). *Phonology in Generative Grammar.* Blackwell Publishing.

Kimper, W. (2012). Harmony is myopic: Reply to Walker 2010. *LI*, 43(2):301–309.

Kiparsky, P. (1973). Abstractness, opacity, and global rules. In Fujimura, O., editor, *Three dimensions in linguistic theory*, pages 57–86. Tokyo: TEC.

Koser, N., Oakden, C., and Jardine, A. (2019). Tone association and output locality in non-linear structures. In Hout, K., Mai, A., McCollum, A., Rose, S., and Zaslansky, M., editors, *Supplemental Proceedings of the 2019 Annual Meeting on Phonology.* LSA.

Lamont, A. (2018). Precedence is pathological: The problem of alphabetical sorting. Poster, WCCFL 36.

Luo, H. (2017). Long-distance consonant agreement and subsequentiality. *Glossa*, 2(52).

McCollum, A., Baković, E., Mai, A., and Meinhardt, E. (2017). Conditional blocking in Tutrugbu requires non-determinism: implications for the subregular hypothesis. In *Paper presented at NELS 48*.

Mills, R. F. (1975). *Proto South Sulawesi and Proto Austronesian Phonology*. PhD thesis, University of Michigan.

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

Moschovakis, Y. N. (2019). *Abstract recursion and intrinsic complexity*, volume 48 of *Lecture Notes in Logic*. Cambridge University Press.

Myers, S. (1987). Vowel shortening in English. *Natural Language and Linguistic Theory*, 5:485–518.

Nespor, M. and Vogel, I. (1986). *Prosodic Phonology*. Dordrecht: Foris.

Odden, D. (1982). Tonal phenomena in Kishambaa. *Studies in African Linguistics*, 13(2):177–208.

Odden, D. (1994). Adjacency parameters in phonology. *Language*, 70(2):289–330.

Oncina, J., García, P., and Vidal, E. (1993). Learning subsequential transducers for pattern recognition tasks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15:448–458.

Onn, F. M. (1980). *Aspects of Malay Phonology and Morphology: A Generative Approach*. Kuala Lampur: Universiti Kebangsaan Malaysia.

Orr, C. (1962). Ecuador Quichua phonology. In Elson, B., editor, *Studies in Ecuadorian Indian Languages*, pages 60–77. Norman, OK: Summer Institute of Linguistics.

Pater, J. (2004). Austronesian nasal substitution and other NC effects. In McCarthy, J., editor, *Optimality Theory in Phonology: A Reader*, pages 271–289. Oxford and Malden, MA: Blackwell.

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

Payne, A. (2017). All dissimilation is computationally subsequential. *Language: Phonological Analysis*, 93(4):e353–371.

Prince, A. (1983). Relating to the grid. *Linguistic Inquiry*, 14:19–100.

Prince, A. and Smolensky, P. (1993). Optimality Theory: Constraint interaction in generative grammar. *Rutgers University Center for Cognitive Science Technical Report*, 2.

Prince, A. and Smolensky, P. (2004). *Optimality Theory: Constraint Interaction in Generative Grammar*. Blackwell Publishing.

Riggle, J. (2004). *Generation, Recognition, and Learning in Finite State Optimality Theory.* PhD thesis, UCLA.

Rogers, J., Heinz, J., Fero, M., Hurst, J., Lambert, D., and Wibel, S. (2013). Cognitive and sub-regular complexity. In *Formal Grammar*, volume 8036 of *Lecture Notes in Computer Science*, pages 90–108. Springer.

Rose, S. and Walker, R. (2004). A typology of consonant agreement as correspondence. *Language*, 80:475–531.

Schadeberg, T. (1982). Nasalization in Umbundu. *Journal of African Languages and Linguistics*, 4:109–32.

Selkirk, E. (1980). The role of prosodic categories in english word stress. *Linguistic Inquiry*, 11(3):563–605.

Selkirk, E. (1984). On the major class features and syllable theory. In Halle, M., Aronoff, M., and Oehrle, R., editors, *Language sound structure: Studies in phonology.* Cambridge, Mass.: MIT Press.

Strother-Garcia, K. (2017). Imdlawn Tashlhiyt Berber syllabification is quantifier-free. In *Proceedings of the first annual meeting of the Society for Computation in Linguistics*, volume 1, pages 145–153.

Strother-Garcia, K. (2019). *Using model theory in phonology: a novel characterization of syllable structure and syllabification.* PhD thesis, University of Delaware.

Strother-Garcia, K., Heinz, J., and Hwangbo, H. J. (2016). Using model theory for grammatical inference: a case study from phonology. In Verwer, S., van Zaanen, M., and Smetsers, R., editors, *Proceedings of The 13th International Conference on Grammatical Inference*, volume 57 of *JMLR: Workshop and Conference Proceedings*, pages 66–78.

Viljoen, J. and Amakali, P. (1978). *A handbook of Oshiwambo.* Pretoria: University of South Africa.

Walker, R. (1996). Prominence-driven stress. Rutgers Optimality Archive. ROA-172, http://roa/rutgers.edu.

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

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

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