Computational universals in linguistic theory: Using recursive programs for phonological analysis

Jane Chandlee, Adam Jardine

Language, Volume 97, Number 3, September 2021, pp. 485-519 (Article)

Published by Linguistic Society of America
DOI: *https://doi.org/10.1353/lan.2021.0045*

COMPUTATIONAL UNIVERSALS IN LINGUISTIC THEORY: USING RECURSIVE PROGRAMS FOR PHONOLOGICAL ANALYSIS

JANE CHANDLEE
Haverford College

ADAM JARDINE
Rutgers University

This article presents BOOLEAN MONADIC RECURSIVE SCHEMES (BMRSs), adapted from the mathematical study of computation, as a phonological theory that both explains the observed computational properties of phonological patterns and directly captures phonological substance and linguistically significant generalizations. BMRSs consist of structures defined as logical predicates and situated in an 'if … then … else' syntax in such a way that they variably LICENSE or BLOCK the features that surface in particular contexts. Three case studies are presented to demonstrate how these grammars (i) express conflicting pressures in a language, (ii) naturally derive ELSEWHERE CONDITION effects, and (iii) capture typologies of repairs for marked structures.*

Keywords: phonology, computation, logic, mathematical linguistics, elsewhere condition, feature-based representations

1. INTRODUCTION.

1.1. COMPUTATIONAL THEORIES OF LANGUAGE. 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 & 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. (This means that regular patterns can be computed by finite-state machines, whose finite numbers of states represent this fixed memory.) For example, progressive vowel harmony is regular because it requires only remembering the relevant feature value in the trigger vowel closest to a span of target vowels (Heinz & Lai 2013). In contrast, consider that the MAJORITY-RULES vowel harmony pathology, in which the output value of a feature in a target vowel is determined by the value that occurs most often in the underlying form (Baković 2000), is not regular (Heinz & Lai 2013). This is because it requires counting the number of feature values in the word, which requires more memory the longer the word is.

As a universal of natural language, the property of being regular was first explored in the domain of sentence well-formedness, with Chomsky (1956) concluding that long-distance syntactic dependencies are incompatible with a regular grammar.[1] This assessment of the expressive power needed by the syntax motivated instead a theory of phrase structure and transformational grammar.

When the regular property came back into the picture as an assessment of the phonological grammar, the difference between these two domains could then be identified in computational terms. The regular/nonregular divide between phonology and syntax enables varying predictions about what constitutes a possible pattern in each domain and points to a need for distinct learning mechanisms (Heinz & Idsardi 2011, 2013). It also introduces the question of where morphology fits in and whether a further computa-

* We sincerely thank Siddharth Bhaskar for introducing us to recursive schemes, as well as Steven Lindell, Jeffrey Heinz, Eric Baković, Adam McCollum, Anna Mai, Eric Meinhardt, and the Rutgers Math Ling research group for their input and suggestions.

[1] Chomsky's actual argument based on English nested embedding was in fact flawed (see Daly 1974, Pullum 2011), but the conclusion that natural language syntax is not regular was later confirmed by others, including Pullum and Gazdar (1982), Shieber (1985), and Kobele (2006).

tional distinction can or should be made between morphophonology and morphosyntax (Heinz & Idsardi 2013, Chandlee 2017).

Recent work has in fact pursued even stronger computational universals for phonology by positing SUBREGULARITY (i.e. properly less expressive than regular) as a better characterization of the phonological grammar (Heinz 2009, Heinz & Lai 2013, Chandlee & Heinz 2018). Such work argues that a subregular approach to phonology allows for restrictive yet robust typological generalizations about the shape and structure of phonology that are unavailable to existing rule- and constraint-based theories (Chandlee et al. 2018, Heinz 2018, Jardine 2019). These computational characterizations also help us make progress on the question of how children learn linguistic generalizations from positive examples (Heinz 2009, 2010, 2018). Furthermore, they have drawn even more attention to the interaction of computation and representation, as a given pattern will have different computational properties depending on how it is represented. Determining the best characterization of some patterns involves a tradeoff between computational and representational complexity; for example, a regular pattern over strings may be subregular over autosegmental representations (see Chandlee & Jardine 2019). As Chomsky's (1956) findings for syntax were based on string representations, it is perhaps not surprising that the work on subregularity in phonology has led some to revisit the earlier conclusions for syntax. A variety of current and recent studies are exploring tree-based equivalents of the subregular properties established for strings (Graf 2019, 2020, Graf & Shafiei 2019, Vu et al. 2019, Ikawa et al. 2020, Ji & Heinz 2020).

1.2. COMPUTATIONAL THEORIES AND LINGUISTIC GENERALIZATIONS. In the context of phonology, 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 (2018:156): 'it is a mistake to conceive of [such computational work] as an alternative to [optimality theory] (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. Indeed, prior work on the computational complexity of OPTIMALITY THEORY (OT) revealed that it can generate nonregular patterns such as majority-rules vowel harmony (for a simple proof that OT can generate nonregular maps, see Gerdemann & Hulden 2012). Various proposals followed for curbing OT's computational overgeneration by altering one or more of its core mechanisms, such as bounding the number of violations a constraint can assign (Frank & Satta 1998, Karttunen 1998), forcing local rather than global evaluation of constraints (Eisner 1997, 2000), or reducing the candidate set to only those that are not harmonically bounded by others (Riggle 2004). What is needed then is a formalism that has OT's ability to incorporate phonological substance but without the computational cost of optimization.

As a solution, we propose here 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 addresses problems of both rule-based, derivational frameworks and constraint-based grammars. Thus, the

BMRS formalism enables a theory of phonology that captures both intensional and computational generalizations about phonology in a way that is unavailable to previous frameworks.

1.3. BOOLEAN MONADIC RECURSIVE SCHEMES. 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, metrical structure building, etc.). And a single algorithm that does all of these things is central to theories like OT, 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) /σσ́σσσ/ → [σσ́σσ́σ]

The crucial question here in computing the output is determining whether an output syllable has the property of carrying an H tone. The BMRS predicate in 2 describes how to compute this. The variable x refers to any given input syllable, $\top$ means true, and $\bot$ means false. Subscript is and os indicate reference to the input and output representation, respectively.

(2) $\acute{\sigma}_o(x) =$ if $\text{final}_i(x)$ then $\bot$ else
if $\acute{\sigma}_o(p(x))$ then $\top$ else
$\acute{\sigma}_i(x)$

The predicate in 2 defines when $\acute{\sigma}_o(x)$ is true: 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 $\acute{\sigma}_o(x)$ evaluates to $\bot$ (false). This implements a nonfinality 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 ($\acute{\sigma}_o(p(x))$), then $\acute{\sigma}_o(x)$ evaluates to $\top$ (true): that is, x is H-toned in the output. This implements the iterative nature of spreading. The final line simply states that (nonfinal) 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, $\text{final}_i(x)$ is a BLOCKING STRUCTURE for an output H tone, because it blocks it from surfacing on x. In contrast, $\acute{\sigma}_o(p(x))$ 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 $\text{final}_i(x)$ supersedes the licensing structure $\acute{\sigma}_o(p(x))$, and thus any x that satisfies both will NOT surface with an H tone.

In contrast to OT, however, the evaluation of a BMRS hierarchy is necessarily LOCAL in nature, and therefore avoids the computational overgeneration that has been attributed to OT's global evaluation strategy (Frank & Satta 1998, Gerdemann & Hulden

2012, Lamont 2019). 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 output predicates such as $\acute{\sigma}_0(x)$ 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.

1.4. OUTLINE. The remainder of the article is structured as follows. We first situate BMRS analyses in the context of a theory that captures the computational nature of phonology (§2) and explain in detail how BMRSs are defined and how they represent a phonological input-output map (§3). Then in §4 we present BMRS 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 *N$\underset{\circ}{\text{C}}$ (§4.3). Section 5 discusses a few potential questions and points of interest raised by the analyses, and we conclude in §6.

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 result—that phonological generalizations which can be described with ordered rewrite rules are regular (Johnson 1972, Kaplan & Kay 1994, Heinz 2018)—distinguishes 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 that spreads to the centermost syllable of the word. An example mapping for this pattern is given in 3.

(3) /σσσσσσσ/ → [σσσσσσσ]

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 have only 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 both are 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, the deterministic FINITE-STATE TRANSDUCER (FST) in Figure 1 computes the attested unbounded H-tone spreading pattern given in 1. (Here and throughout the article we use the symbols $\rtimes$ and $\ltimes$ to represent the start and end of a string, respectively.)

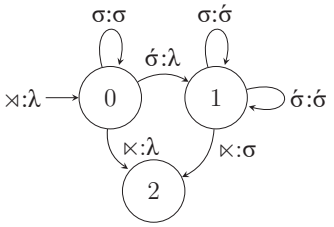

FIGURE 1. An FST computing unbounded H-tone spread to the penult. States are represented as circles, and transitions are represented as arrows, with labels of the form 'input:output'.

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 the empty string (λ). 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 'σ́:λ' is taken to state 1. This means that the syllable is first output as nothing ('σ́:λ'), because the machine does not yet know if that syllable is final (and thus should not receive an H 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 end-of-word boundary, the machine outputs a single nonhigh syllable ('⋉:σ'), which realizes the final syllable as not H-toned.[2]

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 nondeterministic ones (Mohri 1997), essentially because they have a 'bounded lookahead'. That is, the determinism forces any change to a target in the input to 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 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 require unbounded lookahead. This aligns with the independent characterization of spreading as myopic (Wilson 2003, 2006). 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 regular languages are not learnable). Exceptions to this hypothesis in tone (Jardine 2016) and vowel harmony (McCollum et al. 2020) have been noted, but the fact remains that a significant amount of phonology is subsequential.

Aside from typological concerns, however, another goal of a generative theory of phonology is to capture linguistically significant generalizations (Chomsky & Halle

[2] Note that this FST would delete the H tone from a monosyllabic form, under the assumption that final syllables cannot bear a tone. However, that assumption is not necessary for the map to be deterministic.

1968). A theory that both incorporates computational characterizations and intensionally captures linguistically significant generalizations has so far proved to be 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 nonfinality (Prince & Smolensky 1993, Walker 2001, Yip 2002) is not explicit anywhere in the formalism.

Furthermore, the feature-based representations prominent throughout phonological theorizing have not been widely pursued in finite-state analyses of phonological patterns (see Heinz & Koirala 2010 for one example), with transitions instead being labeled with unanalyzed segments. This is due in part to the need to preserve determinism. As noted above, determinism requires that there be exactly one possible transition from a given state for an input symbol, and this restriction is lost when processing feature bundles. For example, if a given state had outgoing transitions for [+voice], [−voice], [+son], [−son], [+labial], [+nasal], and so on, then more than one transition would be possible for an input segment like /b/. The FST would then be nondeterministic, which loses the desirable computational restrictions of subregularity. The use of segment transition labels maintains determinism, but at the cost of generalization. Natural classes of segments might be treated the same by the FST, but that fact is not made explicit in any way.

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 (Strother-Garcia et al. 2016, Jardine 2017a, Strother-Garcia 2017). However, previous work applying logical descriptions to phonological processes has been defined entirely on the input (Heinz 2021), thereby missing generalizations about output-based constraints motivating processes.

In this article we propose BMRSs as a means of addressing all of these limitations. We apply BMRSs to various phonological analyses and show that they can intensionally capture phonological generalizations—using features, reference to the output, and the interaction of constraints. Computational restrictiveness is still guaranteed, as Bhaskar et al. (2020) show that BMRSs are a logical description of the subsequential functions. In addition, 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 an element based on a fixed set of (unary) predicates that refer to the input and output structures local to that element. We illustrate with representations of words as strings of feature bundles, but this easily extends to other structures—for instance, the strings of syllables and tone values used in the introductory example above. (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 §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.

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 $\mathbb{F} = \{[(\pm)\text{F}], [(\pm)\text{G}], \ldots, [(\pm)\text{Z}]\}$ of valued features and boundary symbols $\rtimes$, $\ltimes$, we use a set $\mathcal{I} = \{[\text{F}]_i(t), [\text{G}]_i(t), \ldots, [\text{Z}]_i(t), \rtimes_i(t), \ltimes_i(t)\}$ of INPUT FEATURE PREDICATES (marked with a subscript i) as well as a set of OUTPUT FEATURE PREDICATES $\mathcal{O} = \{[\text{F}]_o(t), [\text{G}]_o(t), \ldots, [\text{Z}]_o(t), \rtimes_o(t), \ltimes_o(t)\}$ (marked with subscript o).

The argument t stands for some TERM that 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 Figure 2 as an example. Figure 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.

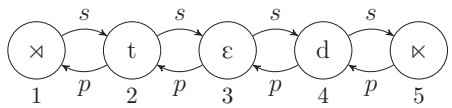

FIGURE 2. An example model of the word /tɛd/.

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. For example, $[\text{son}]_i(x)$ is true when x is interpreted as the [+son] element 3 in Fig. 2: in other words, the sole vowel [ɛ]. The predicate evaluates to false for all other segments. Similarly, $[\text{voi}]_i(x)$ is true only for 3 and 4, and $\ltimes_i(x)$ is true only for 5.

	⋊ 1	t 2	ɛ 3	d 4	⋉ 5
$[\text{son}]_i(x)$	⊥	⊥	⊤	⊥	⊥
$[\text{voi}]_i(x)$	⊥	⊥	⊤	⊤	⊥
$\ltimes_i(x)$	⊥	⊥	⊥	⊥	⊤
$\ltimes_i(s(x))$	⊥	⊥	⊥	⊤	⊥
$[\text{cor}]_i(p(p(x)))$	⊥	⊥	⊥	⊤	⊥

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

We assume the usual representation of segments as bundles of features in $\mathbb{F}$ and interpret the feature predicates as follows. When F is a binary feature, $[\text{F}]_i(t)$ 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. When F is a privative feature, $[\text{F}]_i(t)$ is true for position t when t is [F] in the input, and false otherwise. We do this because in a binary feature system, it is redundant to have separate predicates $[+\text{F}]_i(t)$ and $[-\text{F}]_i(t)$ (as the former is true if and only if the latter is false). While this conflates binary and privative features and does not explicitly represent unspecified feature values, this is not a crucial assumption for the main points in the article. To keep the exposition focused on the logic of BMRSs, we maintain this simplification throughout the article, 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 $\ltimes_i(s(x))$ as 'the successor of x is ⋉'; thus, this is true only when x is evaluated to 4. Likewise, $[\text{cor}]_i(p(p(x)))$ can be read 'the predecessor of the predeces-

sor of *x* is coronal'—in other words, the element two elements preceding *x* is coronal. In Fig. 2, this is also true only for 4.[3]

LOGICAL CONTROL AND DEFINING LOCAL STRUCTURES. The logical backbone of BMRSs is EXPRESSIONS built out of the basic predicates identified in the previous subsection 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 'if E_1 then E_2 else E_3' is an expression;
d. nothing else is an expression.

The definition in 4 makes explicit what is and what is not a BMRS 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, then E returns the result of evaluating E_3. The left-hand side of Figure 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.

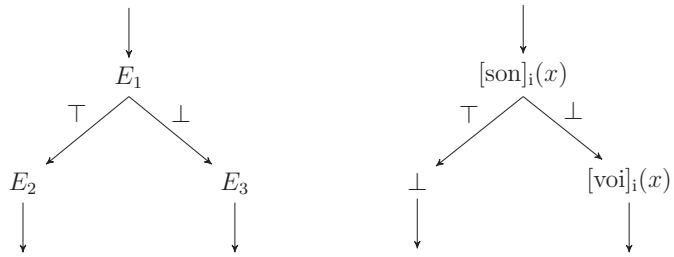

FIGURE 3. Evaluation of 'if … then … else' statements (left), with 5 as an example (right).

To illustrate, the expression in 5 defines what it means to be a voiced obstruent.

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

As shown in the right-hand side of Fig. 3, 5 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 $[\text{voi}]_i(x)$ (= E_3) is returned. If $[\text{voi}]_i(x)$ is true, then we know that *x* is both voiced and an obstruent. Thus, 5 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.[4] Note that in this and future tables, cells are filled in only if that row's predicate is evaluated

[3] 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.

[4] A referee asks how a predicate like $\begin{bmatrix}-\text{son}\\-\text{voi}\end{bmatrix}(x)$, with only negative feature values, would be defined. The 'if … then … else' structures can be embedded, such that this expression would be defined as follows.

(i) $\begin{bmatrix}-\text{son}\\-\text{voi}\end{bmatrix}_i(x) = \text{if } [\text{son}]_i(x) \text{ then } \bot \text{ else if } [\text{voi}]_i(x) \text{ then } \bot \text{ else } \top$

We will see more examples of embedded structures throughout the article.

for that column's element. Whether or not a predicate is evaluated is determined by the 'if … then … else' structure. So, for example, the cell for E_2 and element 1 is blank in Table 2: since 1 evaluates to $\bot$ for E_1, E_2 is skipped and E_3 is evaluated instead.

		⋊	t	ɛ	d	⋉
		1	2	3	4	5
E_1	$[\text{son}]_i(x)$	⊥	⊥	⊤	⊥	⊥
E_2	⊥			⊥		
E_3	$[\text{voi}]_i(x)$	⊥	⊥		⊤	⊥
	$\left[\begin{smallmatrix}-\text{son}\\+\text{voi}\end{smallmatrix}\right]_i(x)$	⊥	⊥	⊥	⊤	⊥

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 left is evaluated for that segment.

Before moving on, it is important to note that 5 names an expression with the shorthand predicate $\left[\begin{smallmatrix}-\text{son}\\+\text{voi}\end{smallmatrix}\right]_i(x)$. In this way, we can derive new predicates that are not in the original set of primitive predicates we started with. However, because the right-hand side of the equation is a BMRS expression, $\left[\begin{smallmatrix}-\text{son}\\+\text{voi}\end{smallmatrix}\right]_i(x)$ 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 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 (Chomsky & Halle 1968) or markedness constraints in OT.

For example, the expression defined in 6 identifies word-final voiced obstruents.[5] Note that in the name of the expression, $\left[\begin{smallmatrix}-\text{son}\\+\text{voi}\end{smallmatrix}\right]$ has been set in boldface to indicate x's position in the structure. This notation is used throughout the article.

(6) $\left[\begin{smallmatrix}\mathbf{-son}\\\mathbf{+voi}\end{smallmatrix}\right]\ltimes_i(x) = \text{if } \left[\begin{smallmatrix}-\text{son}\\+\text{voi}\end{smallmatrix}\right]_i(x) \text{ then } \ltimes_i(s(x)) \text{ else } \bot$

This expression is evaluated similarly to the definition for $\left[\begin{smallmatrix}-\text{son}\\+\text{voi}\end{smallmatrix}\right]_i(x)$ in 5. Here, $\left[\begin{smallmatrix}-\text{son}\\+\text{voi}\end{smallmatrix}\right]_i(x)$ 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 $\ltimes_i(s(x))$ is returned. Recall that $\ltimes_i(s(x))$ returns $\top$ if and only if the successor of x is the word boundary—in other words, if x is word-final. Thus, $\left[\begin{smallmatrix}\mathbf{-son}\\\mathbf{+voi}\end{smallmatrix}\right]\ltimes_i(x)$ 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.

The following section shows how local structures such as 6 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.

DEFINING OUTPUT FEATURE PREDICATES. Predicate logics can model maps by defining the output in terms of the input (Courcelle 1994, Engelfriet & Hoogeboom 2001).[6] This means that we can define a phonological map using featural representations

[5] Equivalently, we could use the standard logical conjunction (i.e. $[-\text{son},+\text{voice}]_i(x) \wedge \ltimes_i(s(x))$), as the 'if … then … else' syntax is equally expressive as the Boolean connectives (Moschovakis 2019). (Note, however, that there is nothing equivalent to quantifiers in the BMRS syntax.) For the sake of consistency, however, we use the 'if … then … else' syntax for all definitions in the article.

[6] 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).

		⋊ 1	b 2	æ 3	d 4	⋉ 5
E_1	$\left[\begin{smallmatrix}-\text{son}\\+\text{voi}\end{smallmatrix}\right]_i(x)$	⊥	⊤	⊥	⊤	⊥
E_2	$⋉_i(s(x))$		⊥		⊤	
E_3	⊥	⊥		⊥		⊥
	$\left[\begin{smallmatrix}\mathbf{-son}\\\mathbf{+voi}\end{smallmatrix}\right]⋉_i(x)$	⊥	⊥	⊥	⊤	⊥

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 left is evaluated for that segment.

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) [−son] → [−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 $[\text{voi}]_o(x)$ in 8.

(8) $[\text{voi}]_o(x) = \text{if } \left[\begin{smallmatrix}\mathbf{-son}\\\mathbf{+voi}\end{smallmatrix}\right]⋉_i(x) \text{ then } ⊥ \text{ else } [\text{voi}]_i(x)$

This states: the value of $[\text{voi}]_o(x)$ for x is false—that is, x is [−voi] in the output—if it is a word-final voiced obstruent. Otherwise, $[\text{voi}]_o(x)$ 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 as [−voi] if it was [−voi] in the input. This is illustrated in Table 4.

	⋊ 1	b 2	æ 3	d 4	⋉ 5
$\left[\begin{smallmatrix}\mathbf{-son}\\\mathbf{+voi}\end{smallmatrix}\right]⋉_i(x)$	⊥	⊥	⊥	⊤	⊥
⊥				⊥	
$[\text{voi}]_i(x)$	⊥	⊤	⊤		⊥
$[\text{voi}]_o(x)$	⊥	⊤	⊤	⊥	⊥

	⋊ 1′	b 2′	æ 3′	d 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.

The output values for other features, such as [±sonorant], [coronal], [labial], and so forth, do not change. We can then simply define the output feature predicates for these features to be faithful to their inputs, as in 9.

(9) $[\text{son}]_o(x) = [\text{son}]_i(x)$
$[\text{cor}]_o(x) = [\text{cor}]_i(x)$
$[\text{lab}]_o(x) = [\text{lab}]_i(x)$

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 in order 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 a set of input valued features $\mathbb{F} = \{[(\pm)\text{F}], [(\pm)\text{G}], \ldots, [(\pm)\text{Z}]\}$, a BMRS definition of a map is a list of definitions:

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

Here each E_i is a BMRS expression and out(x) is a special output predicate that determines whether x has a corresponding output segment (i.e. whether it is deleted).

Thus, 11 is a BMRS definition of word-final obstruent devoicing. We assume for now as a notational simplification that word boundaries are not included in the output (this allows us to write our output feature definitions without worrying about assigning features to boundaries). Table 5 shows how these predicates evaluate for each segment in the example input /bæd/, producing the correct output [bæt].

(11)
$$\begin{aligned} \text{out}(x) &= \text{if } \rtimes_\text{i}(x) \text{ then } \bot \text{ else} \\ &\quad \text{if } \ltimes_\text{i}(x) \text{ then } \bot \text{ else } \top \\ [\text{voi}]_\text{o}(x) &= \begin{bmatrix}\textbf{–son}\\ \textbf{+voi}\end{bmatrix}\ltimes_\text{i}(x) \text{ then } \bot \text{ else } [\text{voi}]_\text{i}(x) \\ [\text{son}]_\text{o}(x) &= [\text{son}]_\text{i}(x) \\ [\text{cor}]_\text{o}(x) &= [\text{cor}]_\text{i}(x) \\ [\text{lab}]_\text{o}(x) &= [\text{lab}]_\text{i}(x) \end{aligned}$$

	⋊	b	æ	d	⋉
	1	2	3	4	5
out(x)	⊥	⊤	⊤	⊤	⊥
$[\text{voi}]_\text{o}(x)$	⊥	⊤	⊤	⊥	⊥
$[\text{son}]_\text{o}(x)$	⊥	⊥	⊤	⊥	⊥
$[\text{cor}]_\text{o}(x)$	⊥	⊥	⊥	⊤	⊥
$[\text{lab}]_\text{o}(x)$	⊥	⊤	⊥	⊥	⊥
	1′	2′	3′	4′	5′
		b	æ	t	

TABLE 5. Evaluation of the BMRS definition for word-final obstruent devoicing in 11 for the example input /bæd/.

As none of the analyses to follow in §4 include epenthesis, we abstract away from how to implement the addition of an element. However, there are established ways of doing so in logical transductions, by specifying copies of input elements. We refer interested readers to Strother-Garcia 2017 for examples.

3.2. LICENSING AND BLOCKING. In the definition for $[\text{voi}]_\text{o}(x)$ in 8, the predicate $\begin{bmatrix}\textbf{–son}\\ \textbf{+voi}\end{bmatrix}\ltimes_\text{i}(x)$ acts as a markedness constraint: if x is in this position in this structure, then $[\text{voi}]_\text{o}(x)$ evaluates to ⊥. Likewise, $[\text{voi}]_\text{i}(x)$ acts like 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 the case when x satisfies $\begin{bmatrix}\textbf{–son}\\ \textbf{+voi}\end{bmatrix}\ltimes_\text{i}(x)$, which takes precedence over $[\text{voi}]_\text{i}(x)$ 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 $A_\text{o}(x)$ is defined as a BMRS expression with the structure in 12.

(12) $A_o(x)$ = if $\text{STRUCT}_1(x)$ then $\{\top, \bot\}$ else
if $\text{STRUCT}_2(x)$ then $\{\top, \bot\}$ else
…
if $\text{STRUCT}_n(x)$ then $\{\top, \bot\}$ else
$A_i(x)$

Here STRUCT_i for any line of the form 'if $\text{STRUCT}_i(x)$ then $\top$' is a LICENSING STRUCTURE, because if it evaluates to $\top$ it causes $A_o(x)$ to be TRUE in the output. Likewise, for any line of the form 'if $\text{STRUCT}_i(x)$ then $\bot$', STRUCT_i is a BLOCKING STRUCTURE, because if it evaluates to $\top$ it causes $A_o(x)$ to be FALSE in the output.

Importantly, the entire definition of $A_o(x)$ is an ORDERING of licensing structures and blocking structures. This is depicted graphically in Figure 4. 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 blocking condition expressed by $\text{STRUCT}_j(x)$ can be violated if and only if some STRUCT_i earlier in the order has been satisfied. The analyses that follow in this article give concrete examples of this logic of computation of the output.

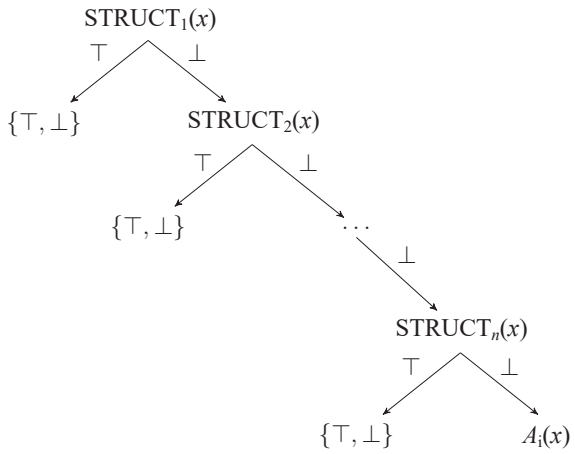

FIGURE 4. Ordering of structures in a BMRS definition of an output property $A_o(x)$.

3.3. RECURSION AND OUTPUT STRUCTURES. From the discussion so far, the BOOLEAN part of BMRSs should be clear.[7] 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 H-tone spreading in Shambaa (Odden 1982). In Shambaa, an underlying H tone spreads to the right, until it reaches the penult.

(13) Shambaa (Bantu; Odden 1982)

/ku-hand-a/	[ku-hand-a]	'to plant'
/ku-fúmbatiʃ-a/	[ku-fúmbátíʃ-a]	'to tie securely'
/ku-fúmbatiʃ-ij-an-a/	[ku-fúmbátíʃ-íj-án-a]	'to tie securely for each other'

[7] The MONADIC part comes from the limitation to a single variable in each term.

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

(14) a. $\sigma \rightarrow \acute{\sigma} \,/\, \acute{\sigma}$ __ σ (iterative)
b. /σόσσσ/ → σόόσσ → σόόόσ → [σόόόσ]

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 input predicates $\acute{\sigma}_i$ to represent the privative property of having an H tone and σ_i to indicate being a syllable.[8] The truth values of each of these predicates for the input /σόσσσ/ are given in Table 6.

	⋊	σ	ό	σ	σ	σ	⋉
	1	2	3	4	5	6	7
$\rtimes_i(x)$	⊤	⊥	⊥	⊥	⊥	⊥	⊥
$\ltimes_i(x)$	⊥	⊥	⊥	⊥	⊥	⊥	⊤
$\sigma_i(x)$	⊥	⊤	⊤	⊤	⊤	⊤	⊥
$\acute{\sigma}_i(x)$	⊥	⊥	⊤	⊥	⊥	⊥	⊥

TABLE 6. Values for the input predicates $\rtimes_i(x)$, $\ltimes_i(x)$, $\sigma_i(x)$, and $\acute{\sigma}_i(x)$ for the input /σόσσσ/.

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

(15) $\text{final}_i(x) = \text{if } \ltimes_i(s(x)) \text{ then } \top \text{ else } \bot$

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

(16) a. $\sigma_o(x) = \sigma_i(x)$
b. $\acute{\sigma}_o(x) = \text{if final}_i(x) \text{ then } \bot \text{ else}$
$\text{if } \acute{\sigma}_o(p(x)) \text{ then } \top \text{ else}$
$\acute{\sigma}_i(x)$

The definition of $\sigma_o(x)$ states that all syllables are mapped faithfully to the output, and the definition of $\acute{\sigma}_o(x)$ states three conditions that govern whether a syllable is H-toned in the output. The first is that, if a syllable is final, then it cannot be high ('if $\text{final}_i(x)$ then $\bot$'). The second states that if the preceding syllable is high IN THE OUTPUT, then x is high ('if $\acute{\sigma}_o(p(x))$ then $\top$'). The final line states that any underlying H-toned syllable is mapped faithfully with an H tone.

In the second condition of 16 we see that the output predicate $\acute{\sigma}_o(x)$ 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 Table 7. Recall that empty cells indicate expressions that are not evaluated for that particular element.

The top three lines of Table 7 give the truth values for the three conditions tested by $\acute{\sigma}_o(x)$. For example, element 2, the first syllable, satisfies none of these conditions: it is not final ($\text{final}_i(x)$ evaluates to $\bot$), its predecessor is not high in the output ($\acute{\sigma}_o(p(x))$ evaluates to $\bot$—see n. 3), and it itself is not high in the input ($\acute{\sigma}_i(x)$ evaluates to $\bot$). Thus $\acute{\sigma}_o(x)$ evaluates to $\bot$ for 2. Turning to element 3, it also evaluates to $\bot$ for $\acute{\sigma}_o(p(x))$, because its predecessor, 2, does. However, because 3 is itself high in the input and thus satisfies $\acute{\sigma}_i(x)$, it does satisfy $\acute{\sigma}_o(x)$.

[8] We use $\acute{\sigma}_i$ instead of ό to explicitly represent tone as a property of the syllable.

	⋊ 1	σ 2	σ́ 3	σ 4	σ 5	σ 6	⋉ 7
$\text{final}_{\text{i}}(x)$	⊥	⊥	⊥	⊥	⊥	⊤	⊥
$\acute{\sigma}_{\text{o}}(p(x))$	⊥	⊥	⊥	⊤	⊤		⊥
$\acute{\sigma}_{\text{i}}(x)$	⊥	⊥	⊤				⊥
$\acute{\sigma}_{\text{o}}(x)$	⊥	⊥	⊤	⊤	⊤	⊥	⊥
$\sigma_{\text{o}}(x)$	⊥	⊤	⊤	⊤	⊤	⊤	⊥
Output:		σ	σ́	σ́	σ́	σ	

TABLE 7. Evaluation of 16 given an input /σσσσσ/. Empty cells indicate predicates that are not evaluated for that element.

Because $\acute{\sigma}_{\text{o}}(p(x))$ is a licensing structure for $\acute{\sigma}_{\text{o}}$, $\acute{\sigma}_{\text{o}}(x)$ is also true for 4. Likewise, because $\acute{\sigma}_{\text{o}}(p(x))$ is true for 5, $\acute{\sigma}_{\text{o}}(x)$ 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 $\acute{\sigma}_{\text{o}}(p(x))$ is also true for element 6. Crucially, however, $\text{final}_{\text{i}}(x)$ is ALSO true for 6, and $\text{final}_{\text{i}}(x)$ is a blocking structure for $\acute{\sigma}_{\text{o}}$. Because this blocking structure OUTRANKS (i.e. takes priority over) the licensing structure $\acute{\sigma}_{\text{o}}(p(x))$, $\acute{\sigma}_{\text{o}}(x)$ evaluates to ⊥ for element 6. This captures the generalization that nonfinality is a crucial condition for H-tone spreading.

Because we allow output predicates in their own definitions, it is possible to write BMRS definitions that are circular, or nonterminating. 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 article (see discussion in Bhaskar et al. 2020). For present purposes, we consider such circular grammars to be noninterpretable 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 are 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 can embed multiple *p*s or *s*s but never combine them—BMRSs describe exactly the subsequential functions. Thus, as a theory of phonology BMRSs correctly exclude unattested patterns like the 'spread to center' pattern mentioned in 3, as well as other nonsubsequential pathologies (Heinz & Lai 2013, Jardine 2016).

We briefly explain why this is the case. Roughly, the recursively defined monadic predicates correspond to states in an FST. Determinism requires bounded lookahead in at least one direction, while recursion enables unbounded lookahead. If we recurse only over *p*, for example, the transduction can 'look' backward an unbounded amount, but its ability to look ahead will be bounded (see Smith & O'Hara 2019 for more on the relationship between lookahead and subsequentiality in the context of phonological maps). The analyses in the following sections all have this property of recursing in at most only one direction.

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 important aspects of phonological generalization 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 BMRSs 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's (1999) 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's (2000) criticisms of Kager's analysis. The following are taken from Kager's (1999) 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) Hixkaryana (Cariban; Kager 1999)
- a. [to.ró:.no] 'small bird'
- b. [ne.mó:.ko.tó:.no] 'it fell'
- c. [a.tʃó:.wo.wo] 'wind'
- d. [kʷá:.ja] 'red and green macaw'

Furthermore, as illustrated in 17c [a.tʃó:.wo.wo] 'wind', Hixkaryana has a nonfinality condition that prevents this iteration from reaching the end of the word (cf. *[a.tʃó:.wo.wó:]). In bisyllabic words such as 17d [kʷá:.ja] 'red and green macaw', this forces the initial syllable to be stressed (cf. *[kʷa.já:]).

Because stress and length in open syllables are entirely predictable, neither need be present in 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 the Kager analysis, this schematic abstracts away from the portion of the grammar that implements syllabification, and takes syllables as the input.[9]

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

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

[9] A referee points out that just marking syllables as H or L ignores a potential underlying distinction, if one adopts a RICHNESS-OF-THE-BASE-style analysis, between underlying closed syllables, which should interrupt the iteration of stress, and underlying long vowels, which should only surface exactly in the case in which they get stress. For simplicity, the analysis presented here assumes (closer to older derivational analyses) that long vowels do not appear in the underlying forms. A similar analysis adhering to richness of the base could be accomplished by replacing $H_i(x)$ with a predicate $closed_i(x)$, indicating a closed syllable, and $long_{i/o}(x)$, governing whether a syllable contains a long vowel.

(19) a. [ák.ma.tá:.ri] ‘branch’
b. [tóh.ku.rʲé:.ho.na] ‘to Tohkurye’
c. [nák.ɲóh.játʃ.ke.ná:.no] ‘they were burning it’
d. [kʰa.ná:.níh.no] ‘I taught you’
e. [mi.há:.na.níh.no] ‘you taught him’

As 19a [ák.ma.tá:.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 ↦ [ĹHLĹHL] (= 19a)
b. HLLLL ↦ [ĹHLĹHLL] (= 19b)
c. HHHLLL ↦ [ĹĹHLĹHL] (= 19c)
d. LLHL ↦ [LĹĹHL] (= 19d)
e. LLLHL ↦ [LĹHLĹHL] (= 19e)

A BMRS analysis of the Hixkaryana facts defines when a syllable is heavy in the output ($H_o(x)$), when it is light in the output ($L_o(x)$), and when it is stressed in the output ($\acute{\sigma}_o(x)$), given the input predicates $L_i(x)$ and $H_i(x)$. The following captures this with iteration governed by the notions of CLASH and LAPSE (Prince 1983, Selkirk 1984, Gordon 2002). A stress clash is when adjacent syllables are stressed, and a stress lapse is when adjacent syllables are unstressed.

The following define what a syllable is (essentially, any H or L)[10] and when a syllable x is potentially in a clash or lapse situation.

(21) a. $\sigma_i(x) = \text{if } L_i(x) \text{ then } \top \text{ else } H_i(x)$
b. $\text{clash}(x) = \text{if } \sigma_i(x) \text{ then } \acute{\sigma}_o(p(x)) \text{ else } \bot$
c. $\text{lapse}(x) = \text{if } \acute{\sigma}_o(p(x)) \text{ then } \bot \text{ else}$
$\text{if } \sigma_i(p(x)) \text{ then } \sigma_i(x) \text{ else } \bot$

The predicate 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 $\acute{\sigma}_o(x)$. The predicate 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.[11] Note that as both of these predicates are defined using both input ($\sigma_i(x)$) and output ($\acute{\sigma}_o(x)$) predicates, they receive neither an input nor an output subscript (a convention we follow throughout the remainder of the article).

We also make use of a predicate only(x), which identifies the first syllable of disyllabic words (monosyllables are disallowed; Hayes 1995:206), as these syllables are always stressed.

(22) $\text{only}_i(x) = \text{if final}_i(s(x)) \text{ then initial}_i(x) \text{ else } \bot$

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) $\acute{\sigma}_o(x) =$ $\text{if only}_i(x) \text{ then } \top \text{ else}$
$\text{if final}_i(x) \text{ then } \bot \text{ else}$
$\text{if } H_i(x) \text{ then } \top \text{ else}$
$\text{if initial}_i(x) \text{ then } \bot \text{ else}$
$\text{if clash}(x) \text{ then } \bot \text{ else}$
$\text{lapse}(x)$

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

[11] Note that the definition of lapse(x) is more complicated than that of clash(x) because of the use of $\acute{\sigma}$ to identify a stressed syllable. The absence of stress cannot be detected by just checking whether the predecessor is σ, because it could still also be stressed.

Setting clash(x) as a blocking structure and lapse(x) as a licensing structure captures exactly left-to-right binary stress. Setting initial$_i$(x) as a blocking structure makes the iteration iambic. This is illustrated with the derivation in Table 8 for an input LLLLL ([ne.móː.ko.tóː.no] 'it fell'; 17b).

	⋊	L	L	L	L	L	⋉
	1	2	3	4	5	6	7
initial$_i$(x)	⊥	⊤	⊥	⊥	⊥	⊥	⊥
clash(x)	⊥		⊥	⊤	⊥	⊤	⊥
lapse(x)	⊥		⊤		⊤		⊥
$\acute{\sqcap}_o$(x)	⊥	⊥	⊤	⊥	⊤	⊥	⊥

TABLE 8. Evaluation of 23 given the input /LLLLL/.

As shown in Table 8, element 2 in the representation—that is, the first L in LLLLL—satisfies initial$_i$(x). This is a blocking structure in the definition of $\acute{\sqcap}_o$(x), and so 2 evaluates to ⊥ for the entire expression and will not receive stress. The second L, 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 $\acute{\sqcap}_o$(p(x)) is true. Since element 2 evaluates to ⊥, clash(x) evaluates to ⊥ for element 3. Conversely, because element 2 evaluates to ⊥ for $\acute{\sqcap}_o$(x), lapse(x) evaluates to ⊤ for 3. As lapse(x) is a licensing structure for $\acute{\sqcap}_o$(x), $\acute{\sqcap}_o$(x) evaluates to ⊤ for 3. Thus, element 3 will be stressed in the output. In turn, this means that the blocking structure clash(x) evaluates to ⊤ for 4, so $\acute{\sqcap}_o$(x) evaluates to ⊥ for 4. This in turn means that the licensing structure lapse(x) evaluates to ⊤ for 5, and so it evaluates to ⊤ for $\acute{\sqcap}_o$(x), and so on.

We now turn to the second and third lines in the definition of $\acute{\sqcap}_o$(x) in 23, which are repeated below in 24.

(24) if final$_i$(x) then ⊥ else
if H$_i$(x) then ⊤ else

As with the unbounded spreading example in §3.3, nonfinality is implemented by setting final$_i$(x) as a blocking structure. Likewise, setting H$_i$(x) as a licensing structure implements the weight-to-stress principle (Prince 1980, 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 Table 9 for an input LLLL ([a.tʃóː.wo.wo] 'wind'; 17c).

	⋊	L	L	L	L	⋉
	1	2	3	4	5	6
final$_i$(x)	⊥	⊥	⊥	⊥	⊤	⊥
H$_i$(x)	⊥	⊥	⊥	⊥		⊥
initial$_i$(x)	⊥	⊤	⊥	⊥		⊥
clash(x)	⊥		⊥	⊤		⊥
lapse(x)	⊥		⊤			⊥
$\acute{\sqcap}_o$(x)	⊥	⊥	⊤	⊥	⊥	⊥

TABLE 9. 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$_i$(x) as a blocking structure above the licensing structure lapse(x). Thus, the final L in LLLL (element 5) would satisfy the licensing structure lapse(x) (note that its predecessor evaluates to ⊥), but it first satisfies final(x)$_i$. This means that $\acute{\sqcap}_o$(x) immediately evaluates to ⊥, leaving lapse(x) not evalu-

ated. Thus, any final L will not be assigned stress. (As final syllables cannot be underlyingly heavy (Hayes 1995:206), it is impossible to know whether final heavy syllables would receive stress, but we have also ranked $\text{final}_i(x)$ over $\text{H}_i(x)$.)

Conversely, any H syllable must receive stress. By ordering $\text{H}_i(x)$ as a licensing structure above the blocking structures $\text{initial}_i(x)$ and $\text{clash}(x)$, this assigns stress to Hs that are in either of these positions. This is illustrated in Table 10 for an input HLLLL ([tóh.ku.rʲéː.ho.na] 'to Tohkurye'; 19b) and in Table 11 for HHHLLL ([nák.ɲóh.játʃ.ke.náː.no] 'they were burning it'; 19c).

	⋊ 1	H 2	L 3	L 4	L 5	L 6	⋉ 7
$\text{final}_i(x)$	⊥	⊥	⊥	⊥	⊥	⊤	⊥
$\text{H}_i(x)$	⊥	⊤	⊥	⊥	⊥		⊥
$\text{initial}_i(x)$	⊥		⊥	⊥	⊥		⊥
$\text{clash}(x)$	⊥		⊤	⊥	⊤		⊥
$\text{lapse}(x)$	⊥			⊤			⊥
$\acute{\sqcap}_o(x)$	⊥	⊤	⊥	⊤	⊥	⊥	⊥

TABLE 10. Evaluation of 23 for the input /HLLLL/.

	⋊ 1	H 2	H 3	H 4	L 5	L 6	L 7	⋉ 8
$\text{final}_i(x)$	⊥	⊥	⊥	⊥	⊥	⊥	⊤	⊥
$\text{H}_i(x)$	⊥	⊤	⊤	⊤	⊥	⊥		⊥
$\text{initial}_i(x)$	⊥				⊥	⊥		⊥
$\text{clash}(x)$	⊥				⊤	⊥		⊥
$\text{lapse}(x)$	⊥					⊤		⊥
$\acute{\sqcap}_o(x)$	⊥	⊤	⊤	⊤	⊥	⊤	⊥	⊥

TABLE 11. Evaluation of 23 for the input /HHHLLL/.

In Table 10, the initial H (element 2) satisfies the licensing structure $\text{H}_i(x)$ and thus $\acute{\sqcap}_o(x)$ evaluates to ⊤. This is despite the fact that H is initial and would otherwise satisfy the blocking structure $\text{initial}_i(x)$. Ranking $\text{H}_i(x)$ over $\text{initial}_i(x)$ then allows initial Hs to receive stress. Likewise, ranking $\text{H}_i(x)$ over $\text{clash}(x)$ captures the generalization that assigning stress to heavy syllables outweighs the drive for placing stress on alternating syllables. Thus, the first three Hs in Table 11 evaluate to ⊤ for $\acute{\sqcap}_o(x)$, even though they would satisfy the blocking structure $\text{clash}(x)$.

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

As shown in Table 12, for an input LL, setting $\text{only}_i(x)$ as a licensing structure and ordering it before the blocking structures $\text{final}_i(x)$ and $\text{initial}_i(x)$ correctly assigns stress to the initial syllable. (The remaining parts of the definition of $\acute{\sqcap}_o(x)$ are omitted.)

	⋊ 1	L 2	L 3	⋉ 4
$\text{only}_i(x)$	⊥	⊤	⊥	⊥
$\text{final}_i(x)$	⊥		⊤	⊥
$\text{H}_i(x)$	⊥			⊥
$\text{initial}_i(x)$	⊥			⊥
$\acute{\sqcap}_o(x)$	⊥	⊤	⊥	⊥

TABLE 12. Evaluation of relevant statements in 23 for input /LL/.

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

(25) a. $H_o(x)$ = if $\acute{\sigma}_o(x)$ then $\top$ else $H_i(x)$
b. $L_o(x)$ = if $\acute{\sigma}_o(x)$ then $\bot$ else $L_i(x)$

Definition 25a for $H_o(x)$ 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 applies only to final H syllables that did not receive stress). Definition 25b 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 25 work together to produce the correct outputs for the inputs LLHL ([k^{h}a.ná:.níh.no] 'I taught you'; 19d) and HHHLLL ([nák.ɲóh.játʃ.ke.ná:.no] 'they were burning it'; 19c) is shown in the full derivations in Table 13. The individual lines of the definition of $\acute{\sigma}_o(x)$ are given for clarity.

	⋊ 1	L 2	L 3	H 4	L 5	⋉ 6
$only_i(x)$	⊥	⊥	⊥	⊥	⊥	⊥
$final_i(x)$	⊥	⊥	⊥	⊥	⊤	⊥
$H_i(x)$	⊥	⊥	⊥	⊤		⊥
$initial_i(x)$	⊥	⊤	⊥			⊥
$clash(x)$	⊥		⊥			⊥
$lapse(x)$	⊥		⊤			⊥
$\acute{\sigma}_o(x)$	⊥	⊥	⊤	⊤	⊥	⊥
$H_o(x)$	⊥	⊥	⊤	⊤	⊥	⊥
$L_o(x)$	⊥	⊤	⊥	⊥	⊤	⊥
Output:		L	Ĥ	Ĥ	L	

	⋊ 1	H 2	H 3	H 4	L 5	L 6	L 7	⋉ 8
$only_i(x)$	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥
$final_i(x)$	⊥	⊥	⊥	⊥	⊥	⊥	⊤	⊥
$H_i(x)$	⊥	⊤	⊤	⊤	⊥	⊥		⊥
$initial_i(x)$	⊥				⊥	⊥		⊥
$clash(x)$	⊥				⊤	⊥		⊥
$lapse(x)$	⊥					⊤		⊥
$\acute{\sigma}_i(x)$	⊥	⊤	⊤	⊤	⊥	⊤	⊥	⊥
$H_o(x)$	⊥	⊤	⊤	⊤	⊥	⊤	⊥	⊥
$L_o(x)$	⊥	⊥	⊥	⊥	⊤	⊥	⊤	⊥
Output:		Ĥ	Ĥ	Ĥ	L	Ĥ	L	

TABLE 13. Evaluation of 23 and 25 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 alternating stress pattern, 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's (1995) 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 $final_i(x)$ above the licensing structures (except for $only_i(x)$) 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. nonfinality, 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's (1999) analysis as *[LLĤL] (as opposed to the attested [LĤĤL], e.g. [k^{h}a.ná:.níh.no] 'I taught you' (19d)). This is due to the fact that, in OT, *[LLĤL] is directly compared to [LĤĤL], and the former turns out to be more har-

monic than the latter given Kager's (1999) high ranking of Uneven-Iamb, which penalizes iambs of the form (LĹ) and (Ĥ). The BMRS analysis obtains the correct output [LĤĤL] by assigning stress exactly as Halle and Idsardi (2000:202) 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'. The way the statements in a BMRS analysis are evaluated crucially limits its ability to look ahead.

Another point of divergence between the analysis presented here and those of Hayes (1995) and Kager (1999) is that, like that of 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 on how they can be extended to build structures. Logical characterizations can also capture structure building; see, for example, the work of Jardine (2017b) and Strother-Garcia (2019).

Finally, a potential weakness of this analysis is the need for the licensing structure $\text{only}_{\text{i}}(x)$ to identify the initial syllable in disyllabic forms. This licenses stress in exactly the case in which cumulativity of stress seems to force stress to appear, even if violating other constraints. However, while the use of this structure is somewhat stipulative, treating disyllabic forms differently is unavoidable, and as Halle and Idsardi (2000) point out, even Kager's (1999) analysis does not capture this by appealing to a cumulativity constraint on stress. 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 disyllabic forms. In Kager's (1999:160) analysis, the ranking Ft-Bin >> Dep-IO motivates lengthening in bisyllabic words, whereas the separate ranking Uneven-Iamb >> Dep-IO motivates lengthening in longer words. In Hayes's rule-based analysis, an additional rule is required, as pointed out by Kager (1999:149): 'When the entire metrical domain is a single light syllable, assign (L) to it' (ex. 17, step 4). Thus, in its special treatment of short forms, this BMRS analysis is not significantly different from other analyses.

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 & Smolensky 2004, Baković 2006, 2013), from which elsewhere condition (EC; Kiparsky 1973) effects can be derived.[12] We state this characterization as the STRICT SUBSTRUCTURE ORDERING THEOREM, or SSOT.

Theorem 1. STRICT SUBSTRUCTURE ORDERING THEOREM (SSOT): 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), if $i < j$ in the order, then $\text{STRUCT}_j(x)$ will never evaluate to $\top$ and thus never take effect.

The theorem follows directly from the logic of the 'if … then … else' syntax. The proof, given below, is therefore straightforward.

Proof. Let A and B be two structures such that whenever B is true, A is true, but the converse does not hold. Now assume A precedes B in a ranking of structures. For B to take effect there must be at least one x such that A evaluates to $\bot$ and B evaluates to $\top$. But since whenever B is true, A is true, no such case exists. Therefore for B to ever take effect it must precede A. □

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

To illustrate, consider the structures $C_i(x)$, indicating that x is a consonant, and $V\mathbf{C}V_i(x)$, indicating an intervocalic consonant. Intuitively, $V\mathbf{C}V_i(x)$ implies $C_i(x)$—an intervocalic consonant must, of course, be a consonant. Formally, $V\mathbf{C}V_i(x)$ can only be true for any x for which $C_i(x)$ is also true. In other words, $C_i(x)$ is a strict substructure of $V\mathbf{C}V_i(x)$.

According to theorem 1, the strict substructure relationship fixes the order in which $C_i(x)$ and $V\mathbf{C}V_i(x)$ can appear in a BMRS analysis. To see why, consider the definition for $[\text{voi}]_o(x)$ in 26, in which $V\mathbf{C}V_i(x)$ serves as a licensing structure ranked over $C_i(x)$ as a blocking structure. Table 14 shows the evaluation of this predicate for the inputs /papa/ and /baba/. (Evaluation of the vowels has been omitted.)

(26) $[\text{voi}]_o(x)$ = if $V\mathbf{C}V_i(x)$ then $\top$ else
if $C_i(x)$ then $\bot$ else
$[\text{voi}]_i(x)$

	p	a	p	a
$V\mathbf{C}V_i(x)$	$\bot$		$\top$	
$C_i(x)$	$\top$			
$[\text{voi}]_o(x)$	$\bot$		$\top$	
	p	a	b	a

	b	a	b	a
$V\mathbf{C}V_i(x)$	$\bot$		$\top$	
$C_i(x)$	$\top$			
$[\text{voi}]_o(x)$	$\bot$		$\top$	
	p	a	b	a

TABLE 14. Evaluation of 26 for inputs /papa/ and /baba/.

If $V\mathbf{C}V_i(x)$ evaluates to $\top$, then $[\text{voi}]_o(x)$ evaluates to $\top$. If not, then evaluation passes down the line, and if it reaches $C_i(x)$ and evaluates to $\top$, then $[\text{voi}]_o(x)$ evaluates to $\bot$. In both examples in Table 14, $[\text{voi}]_o(x)$ evaluates to $\bot$ for the initial consonant, and to $\top$ for the intervocalic consonant.

If the priority of the two were instead reversed, as in 27, then $V\mathbf{C}V_i(x)$ would never get a chance to serve as a licensing structure, because for any x for which $V\mathbf{C}V_i(x)$ is true, $C_i(x)$ will be true. In this case, $C_i(x)$ will always take effect and $V\mathbf{C}V_i(x)$ will never have a chance. This is illustrated in Table 15. Note that now the truth value for $[\text{voi}]_o(x)$ for all consonants is $\bot$, because even though the intervocalic consonants satisfy $V\mathbf{C}V_i(x)$, $C_i(x)$ always preempts it. In other words, $V\mathbf{C}V_i(x)$ might as well not be present in the definition at all.

(27) $[\text{voi}]_o(x)$ = if $C_i(x)$ then $\bot$ else
if $V\mathbf{C}V_i(x)$ then $\top$ else
$[\text{voi}]_i(x)$

	p	a	p	a
$C_i(x)$	$\top$		$\top$	
$V\mathbf{C}V_i(x)$				
$[\text{voi}]_o(x)$	$\bot$		$\bot$	
	p	a	p	a

	b	a	b	a
$C_i(x)$	$\top$		$\top$	
$V\mathbf{C}V_i(x)$				
$[\text{voi}]_o(x)$	$\bot$		$\bot$	
	p	a	p	a

TABLE 15. Evaluation of 27 for inputs /papa/ and /baba/.

To give a more concrete example, the reader can confirm that, with the structures $\text{only}_i(x)$ and $\text{initial}_i(x)$ in the BMRS definition for Hixkaryana stress in the previous section, $\text{initial}_i(x)$ is a strict substructure of $\text{only}_i(x)$. The SSOT then fixes the relative ranking of these two predicates, such that the ordering of $\text{only}_i(x)$ before $\text{initial}_i(x)$ in 23 is the only way in which the former can play a role in the grammar.

To see how the SSOT predicts EC effects, we turn to stress and length in English. As discussed by Baković (2006, 2013) (following Chomsky & 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 that the final suffixed syllables in these forms are extrametrical.

(28) a. (n**ă**tu)ral (cf. n**ā**ture)
b. di(v**ĭ**ni)ty (cf. div**ī**ne)
c. (r**ă**di)cal

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

(29) a. re(m**ē**di)al *re(m**ĕ**di)al
b. co(l**ō**ni)al *co(l**ŏ**ni)al
c. (r**ā**di)al *(r**ă**di)al

Stressed vowels in binary feet are then 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 give a BMRS analysis of these facts.

We assume a predicate $\text{brhd}_i(x)$ that is true exactly when x is the head of a binary branching foot, as well as a predicate **[−hi]**$\text{CiV}_i(x)$ that identifies a [−hi] segment followed by the sequence CiV. To identify the environment for lengthening, we combine these into the predicate in 30.

(30) $\text{brhd\&}$**[−hi]**$\text{CiV}_i(x) = \text{if brhd}_i(x)$ then **[−hi]**$\text{CiV}_i(x)$ else $\bot$

The BMRS definition for the feature [±long] in English is then defined as in 31.

(31) $[\text{long}]_o(x)$ = if $\text{brhd\&}$**[−hi]**$\text{CiV}_i(x)$ then $\top$ else
if $\text{brhd}_i(x)$ then $\bot$ else
$[\text{long}]_i(x)$

This definition involves one licensing structure, $\text{brhd\&}$**[−hi]**$\text{CiV}_i(x)$ (the lengthening environment), ordered before a blocking structure $\text{brhd}_i(x)$ (the shortening environment). Note that this order is fixed by the SSOT: $\text{brhd\&}$**[−hi]**$\text{CiV}_i(x)$ implies $\text{bhrd}_i(x)$, so the former must take priority over the latter. The definition in 31 is evaluated as shown in Table 16, with the examples *radical* and *radial*.

	(r**a**di)cal
$\text{brhd\&}$**[−hi]**$\text{CiV}_i(x)$	$\bot$
$\text{brhd}_i(x)$	$\top$
$[\text{long}]_o(x)$	$\bot$
Output:	r**ă**dical

	(r**a**di)al
$\text{brhd\&}$**[−hi]**$\text{CiV}_i(x)$	$\top$
$\text{brhd}_i(x)$	
$[\text{long}]_o(x)$	$\top$
Output:	r**ā**dial

TABLE 16. Evaluation of 31 for inputs *radical* and *radial*.

As illustrated in Table 16, the stressed vowel in *radical* fails $\text{brhd\&}$**[−hi]**$\text{CiV}_i(x)$, so evaluation passes along to $\text{brhd}_i(x)$. This evaluates to $\top$, and since $\text{bhrd}_i(x)$ is a blocking structure for $[\text{long}]_o(x)$, the entire predicate evaluates to $\bot$. This produces the correct output *rădical*. In contrast, the stressed vowel in *radial* satisfies $\text{brhd\&}$**[−hi]**$\text{CiV}_i(x)$. As this is a licensing structure for $[\text{long}]_o(x)$, the predicate immediately evaluates to $\top$, skipping the evaluation of $\text{brhd}_i(x)$. This produces the correct output *rādial*.

Note that if the order of the two structures were reversed, the licensing structure brhd&**[−hi]**CiV$_i$(x) would never have an effect. This is illustrated in Table 17.

	(**ra**di)cal
brhd$_i$(x)	⊤
brhd&**[−hi]**CiV$_i$(x)	
[long]$_o$(x)	⊥
Output:	**ră**dical

	(**ra**di)al
brhd$_i$(x)	⊤
brhd&**[−hi]**CiV$_i$(x)	
[long]$_o$(x)	⊥
Output:	***ră**dial

TABLE 17. Incorrect output due to violation of the SSOT.

Exactly as in the example with only$_i$(x) and initial$_i$(x), bhrd$_i$(x) is a strict substructure of brhd&**[−hi]**CiV$_i$(x); logically, the latter implies the former. This means that for any element for which brhd&**[−hi]**CiV(x)$_i$ is true, so is bhrd$_i$(x). Thus, if we were to order bhrd$_i$(x) before brhd&**[−hi]**CiV$_i$(x), the latter would never have an effect and would be extraneous in the grammar.

Note also that 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 behavior of BMRSs with a rule-based analysis, we briefly discuss how derivational analyses invoke the EC to effect this disjunctive behavior. Kenstowicz (1994:218, ex. 36) formulates the rules for lengthening and shortening as in 32.

(32) English lengthening and shortening (Kenstowicz 1994)

a. Shortening

$$\bar{V} \rightarrow \breve{V} / \underline{\quad} \begin{array}{c} C_0 \\ | \\ (\acute{\sigma} \end{array} \begin{array}{c} V \\ | \\ \sigma) \end{array}$$

b. Lengthening

$$\breve{V} \rightarrow \bar{V} / \underline{\quad} \begin{array}{c} C \\ | \\ (\acute{\sigma} \end{array} \begin{array}{c} i \\ | \\ \sigma) \end{array} \begin{array}{c} V \\ \\ \\ \end{array}$$

As Kenstowicz (1994) argues (see also Halle 1995, Halle & 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 33a), and ordering shortening before lengthening produces a 'Duke of York' effect in which shortening makes a change that is then undone (see 33b; and see Halle & Idsardi 1998 for discussion).

(33) Nondisjunctive 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 32b is strictly more specific than that of 32a. This is exactly the situation in which the EC states that shortening and lengthening should be disjunctively ordered, with lengthening (32b) applying in the specific situations in which its environment is met, and shortening (32a) elsewhere. The derivations in 34 illustrate.

(34)

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

This produces the correct forms. However, as pointed out by Baković (2006, 2013), the EC is an external principle independent of the operation of the SPE rule system.[13] 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 automatically from how the grammar is evaluated.

4.3. The typology of *NC̥. Our third case study is the well-known typology of *NC̥ 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 35 is responsible for the diverse set of repairs.

(35) *NC̥: No nasal/voiceless obstruent sequences (Pater 1999)

Pater (1999) demonstrates how the typology of repairs is captured through permutations of ranking various faithfulness constraints with *NC̥. In contrast, rule-based analyses assign each language 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 *NC̥. As in the previous analyses, the placement of these structures is constrained by the inherent logic of BMRSs.

We first address nasal place assimilation, which plays a role in all of these grammars. All analyses in this section assume a definition of the output place features as given for $[\text{lab}]_o(x)$ in 36.[14] We assume comparable definitions for $[\text{cor}]_o(x)$ and $[\text{dor}]_o(x)$.

(36) $[\text{lab}]_o(x)$ = if **[nas]**$[\text{lab}]_i(x)$ then $\top$ else
if **[nas]**$[\text{cor}]_i(x)$ then $\bot$ else
if **[nas]**$[\text{dor}]_i(x)$ then $\bot$ else
$[\text{lab}]_i(x)$

Next we define the two predicates in 37, which identify the two segments in an NC̥ structure. The reason two predicates are needed is because the repair strategies vary in whether they target the nasal or the obstruent in these structures. One might object to this need for two predicates, given that in OT only a single markedness constraint is

[13] A referee points out that the SSOT echoes the proper inclusion precedence principle of Sanders (1974) and Koutsoudas et al. (1974), which states that a rule A must apply before a rule B iff A's structural description properly includes that of B. Like the EC, proper inclusion precedence is an external constraint on the SPE system and cannot be derived from the system itself. We thank the referee for bringing this connection to our attention.

[14] There are at least two ways to make a more succinct statement that deviate slightly from the BMRS syntax we have been using. To directly state place assimilation to a following consonant, we could instead write (i).

(i) $[\text{lab}]_o(x)$ = if **[nas]**$C_i(x)$ then $[\text{lab}]_i(s(x))$ else $[\text{lab}]_i(x)$

This states that a nasal must take on the value for [lab] of a following consonant, if one exists. Another way is with a three-valued predicate $[\text{place}]_o(x)$ that takes one of the values in {lab, cor, dor}.

(ii) $[\text{place}]_o(x)$ = if **[nas]**$C_i(x)$ then $[\text{place}]_i(s(x))$ else $[\text{place}]_i(x)$

Both of these deviate from the BMRS structure established in §3.1, in that x takes on a place value rather than a Boolean value. However, this is still valid BMRS syntax.

needed. But the relationship between the two predicates is structured, in that they are mirror images of each other. They represent equally valid ways of referring to a pair of segments, so we can posit that whenever one is available, so is the other.[15]

(37) a. $\mathbf{NC}_{\circ i}(x) = \text{if } [\text{nas}]_i(x) \text{ then } \begin{bmatrix}-\text{son}\\-\text{voi}\end{bmatrix}_i(s(x)) \text{ else } \bot$

b. $\mathbf{NC}_{\circ i}(x) = \text{if } \begin{bmatrix}-\text{son}\\-\text{voi}\end{bmatrix}_i(x) \text{ then } [\text{nas}]_i(p(x)) \text{ else } \bot$

Given these predicates, we can then examine their various effects as licensing and blocking structures for the definitions of the predicates $\text{out}(x)$, $[\text{nas}]_o(x)$, and $[\text{voi}]_o(x)$ determining whether x is output, nasal, and voiced, respectively.

We show that, like Pater's (1999) factorial typology of *NC$_\circ$ and various faithfulness constraints, the variable placement of these structures in a BMRS definition describes the typology of nasal-voiceless consonant repairs. Furthermore, the following definition, similar to the SSOT in the previous section, governs where these structures can be placed. We use this definition to explain why certain logically possible repairs do not occur. Also like the SSOT, this definition is not an artificial restriction on BMRS grammars, but rather the description of a consequence of how they are evaluated.

Definition 1. MEANINGFUL STRUCTURES IN BMRSs: We consider BMRS transductions whose licensing and blocking structures are MEANINGFUL. In a BMRS of the structure 'if X then B 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 $\neg Y$.

As in the SSOT, 'X implies Y' means that whenever X is true, Y is true. For example, the default for $\text{out}(x)$ is $\top$. Thus, unless some other blocking structure intervenes in the order, using either $\mathbf{NC}_{\circ i}(x)$ or $\mathbf{NC}_{\circ i}(x)$ as a licensing structure is not meaningful, because $\text{out}(x)$ in this case would be evaluated to $\top$ anyway. To see this, in the definition of $\text{out}(x)$ in 38a, $\mathbf{NC}_{\circ i}(x)$ is a licensing structure. However, it also implies $\top$—whenever $\mathbf{NC}_{\circ i}(x)$ is true, $\top$ is true (somewhat vacuously, as $\top$ is always true). Therefore, by definition 1, $\mathbf{NC}_{\circ i}(x)$ is not meaningful.

(38) a. $\text{out}(x) = \text{if } \mathbf{NC}_{\circ i}(x) \text{ then } \top \text{ else } \top$

b. $\text{out}(x) = \text{if } \mathbf{NC}_{\circ i}(x) \text{ then } \top \text{ else if } [\text{nas}]_i(x) \text{ then } \bot \text{ else } \top$

However, if we add another intervening blocking structure, such as $[\text{nas}]_i(x)$ in 38b, then $\mathbf{NC}_{\circ i}(x)$ is now meaningful, because it does not imply all of 'if $[\text{nas}]_i(x)$ then $\bot$ else $\top$'.

Thus, $\mathbf{NC}_{\circ i}(x)$ on its own cannot function as a licensing structure for $\text{out}(x)$, and for similar reasons neither can $\mathbf{NC}_{\circ i}(x)$. However, they can act as blocking structures for $\text{out}(x)$, which results in deletion. Deletion of the consonant results from using $\mathbf{NC}_{\circ i}(x)$. This is attested in Indonesian (Onn 1980, Herbert 1986, Pater 1999), in which the infinitive 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.[16]

[15] We thank a referee for suggesting this response to such an objection.

[16] As noted in Pater 1999, this alternation occurs only 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 rt-internal(x) and assigning it as a higher-ranked licensing structure for $\text{out}(x)$ as below.

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

(ii) $\text{out}(x) = \text{if rt-internal}(x) \text{ then } \top \text{ else if } \mathbf{NC}_{\circ i}(x) \text{ then } \bot \text{ else } \top$

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

(39) Fusion in Indonesian (Austronesian; Onn 1980)
/məNpilih/ [məmilih] 'to choose, vote'

(40) $\text{out}(x) = \text{if } \mathbf{NC}_{\circ i}(x) \text{ then } \bot \text{ else } \top$
$[\text{nas}]_o(x) = [\text{nas}]_i(x)$
$[\text{voi}]_o(x) = [\text{voi}]_i(x)$

Note that the place assimilation is accomplished through the definition in 36, which applies even in the case of deletion of the consonant because it is defined in terms of the input. Note also that this analysis is similar in spirit to derivational analyses in which both assimilation and deletion occur (e.g. Onn 1980, Zaleska 2018), rather than Pater's (1999) 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. At the same time, this analysis does not occur in 'two steps', as Pater points out about analyses like that of Onn (1980)—the BMRS analysis describes a single mapping from input to output.

Deletion of the nasal instead is achieved by using $\mathbf{NC}_{\circ i}(x)$ as a blocking structure for $\text{out}(x)$, as in 41. This is attested in Swahili (Choti 2015), for example, as seen in 42.

(41) $\text{out}(x) = \text{if } \mathbf{NC}_{\circ i}(x) \text{ then } \bot \text{ else } \top$
$[\text{nas}]_o(x) = [\text{nas}]_i(x)$
$[\text{voi}]_o(x) = [\text{voi}]_i(x)$

(42) Nasal deletion in Swahili (Bantu; Choti 2015)
/Nbaya/ [mbaya] 'CLASS9/10.bad'
/Nkubwa/ [kubwa] 'CLASS9/10.big'

Repairs that change voicing and nasality then result from using $\text{N}\mathbf{C}_{\circ i}(x)$ or $\mathbf{NC}_{\circ i}(x)$ as licensing or blocking structures for $[\text{voi}]_o(x)$ and $[\text{nas}]_o(x)$, respectively. We go through these various options in turn. First, when $\text{N}\mathbf{C}_{\circ i}(x)$ appears as a licensing structure in $[\text{voi}]_o(x)$, as in 43, the result is postnasal voicing. This occurs in Puyo Pongo Quechua (Orr 1962), as shown in 44.

(43) $\text{out}(x) = \top$
$[\text{nas}]_o(x) = [\text{nas}]_i(x)$
$[\text{voi}]_o(x) = \text{if } \text{N}\mathbf{C}_{\circ i}(x) \text{ then } \top \text{ else } [\text{voi}]_i(x)$

(44) Postnasal voicing in Puyo Pungo Quechua (Quechua; Orr 1962)
/kampa/ [kamba] 'yours'

$\text{N}\mathbf{C}_{\circ i}(x)$ cannot, however, serve as a blocking structure for $[\text{voi}]_o(x)$, because it would not be meaningful according to definition 1. This is because $\text{N}\mathbf{C}_{\circ i}(x)$ identifies a voiceless segment and therefore it does imply $\neg[\text{voi}]_i(x)$.

We now consider $\mathbf{NC}_{\circ i}(x)$ as a licensing structure for $[\text{voi}]_o(x)$. First we recognize that in most languages, nasality implies voicing. This can be captured by adding $[\text{nas}]_i(x)$ as a licensing structure for the definition of $[\text{voi}]_o(x)$.

(45) $[\text{voi}]_o(x) = \text{if } [\text{nas}]_o(x) \text{ then } \top \text{ else } [\text{voi}]_i(x)$

Given 45, either $\mathbf{NC}_{\circ i}(x)$ is not meaningful as a licensing structure or it violates the SSOT.

(46) a. $[\text{voi}]_o(x) = \text{if } \mathbf{NC}_{\circ i}(x) \text{ then } \top \text{ else}$
$\text{if } [\text{nas}]_o(x) \text{ then } \top \text{ else}$
$[\text{voi}]_i(x)$

b. $[\text{voi}]_o(x) = \text{if } [\text{nas}]_o(x) \text{ then } \top \text{ else}$
$\text{if } \mathbf{NC}_{\circ i}(x) \text{ then } \top \text{ else}$
$[\text{voi}]_i(x)$

Consider first 46a. Whenever $\mathbf{NC}_{\circ i}(x)$ is true, $[\text{nas}]_o(x)$ is also true (unless it is blocked in that exact configuration). Therefore, $\mathbf{NC}_{\circ i}(x)$ implies 'if $[\text{nas}]_o(x)$ then $\top$ else $[\text{voi}]_i(x)$',

in violation of definition 1. So 46a is excluded by being meaningless. Likewise, the reverse ordering in 46b violates the SSOT, because $[\text{nas}]_i(x)$ is a proper substructure of $\mathbf{NC}_{\circ i}(x)$. Thus, both possible orderings of $\mathbf{NC}_{\circ i}(x)$ and $[\text{nas}]_i(x)$ as licensing structures are forbidden by either theorem 1 or definition 1.

However, $\mathbf{NC}_{\circ i}(x)$ can appear as a blocking structure for $[\text{voi}]_o(x)$, causing devoicing of the nasal. This is shown in 47, with a hypothetical example in 48.

(47) $\text{out}(x) = \top$
$[\text{nas}]_o(x) = [\text{nas}]_i(x)$
$[\text{voi}]_o(x) = \text{if } \mathbf{NC}_{\circ i}(x) \text{ then } \bot \text{ else } [\text{voi}]_i(x)$

(48) Nasal devoicing
/ampa/ [aṃ̥pa]

Whether this repair is attested is controversial, as noted by Pater (1999). Herbert (1986) cites Ndonga (Viljoen & Amakali 1978) and Pokomo (Hinnebusch 1975) as having nasal devoicing in this context, though Huffman and Hinnebusch (1998) argue that it is only a phonetic effect in Pokomo.

Finally, we consider $\mathbf{NC}_{\circ i}(x)$ and $\mathbf{NC}_{\circ i}(x)$ as licensing and blocking structures for $[\text{nas}]_o(x)$. As a licensing structure for $[\text{nas}]_o(x)$, $\mathbf{NC}_{\circ i}(x)$ predicts nasalization of the obstruent. This occurs in Konjo (Friberg & Friberg 1991:84–86), as shown in 49. The predicate definitions are given in 50.

(49) Nasalization in Konjo (Niger-Congo; Friberg & Friberg 1991)
a. /aŋpinahaŋ/ [əmminahaŋ] ‘to follow’
b. /aŋkanre/ [əŋŋanre] ‘to eat’

(50) $\text{out}(x) = \top$
$[\text{nas}]_o(x) = \text{if } \mathbf{NC}_{\circ i}(x) \text{ then } \top \text{ else } [\text{nas}]_i(x)$
$[\text{voi}]_o(x) = \text{if } [\text{nas}]_o(x) \text{ then } \top \text{ else } [\text{voi}]_i(x)$

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

As a blocking structure for $[\text{nas}]_o(x)$, $\mathbf{NC}_{\circ i}(x)$ 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 $[-\text{son}]_o(x)$ as a blocking structure for $[\text{nas}]_o(x)$.

(51) a. $[-\text{son}]_o(x) = \text{if } [\text{son}]_o(x) \text{ then } \bot \text{ else } \top$
b. $[\text{nas}]_o(x) = \text{if } [-\text{son}]_o(x) \text{ then } \bot \text{ else } [\text{nas}]_i(x)$

But whenever this implication is present for $[\text{nas}]_o(x)$, as a blocking structure, $\mathbf{NC}_{\circ i}(x)$ is either meaningless or is blocked by the SSOT, for reasons parallel to those given above for $\mathbf{NC}_{\circ i}(x)$ when nasality implied voicing. The predicates in 52 demonstrate.

(52) a. $[\text{nas}]_o(x) = \text{if } \mathbf{NC}_{\circ i}(x) \text{ then } \bot \text{ else}$
$\text{if } [-\text{son}]_o(x) \text{ then } \bot \text{ else}$
$[\text{nas}]_i(x)$
b. $[\text{nas}]_o(x) = \text{if } [-\text{son}]_o(x) \text{ then } \bot \text{ else}$
$\text{if } \mathbf{NC}_{\circ i}(x) \text{ then } \bot \text{ else}$
$[\text{nas}]_i(x)$

Turning to $\mathbf{NC}_{\circ i}(x)$, as a blocking structure for $[\text{nas}]_o(x)$ it causes denasalization, as in Mandar (Mills 1975).

(53) Gemination in Mandar (Austronesian; Mills 1975)
/mantunu/ [mattunu] ‘to burn’

(54) $\text{out}(x) = \top$
$[\text{nas}]_o(x) = \text{if } \mathbf{NC}_{\circ i}(x) \text{ then } \bot \text{ else } [\text{nas}]_i(x)$
$[\text{voi}]_o(x) = \text{if } [-\text{voi}]_i(s(x)) \text{ then } \bot \text{ else } [\text{voi}]_i(x)$

As a blocking structure for $[\text{nas}]_o(x)$ in 54, $\mathbf{NC}_{\circ i}(x)$ prevents the N in an 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 $[-\text{voi}]_i(s(x))$ as a blocking structure for $[\text{voi}]_o(x)$.) Finally, as a licensing structure for $[\text{nas}]_o(x)$, $\mathbf{NC}_{\circ i}(x)$ is meaningless. This is because whenever $\mathbf{NC}_{\circ i}(x)$ is true, then clearly the default predicate $[\text{nas}]_i(x)$ is also true.

Table 18 thus summarizes the processes predicted by varying $\mathbf{NC}_{\circ i}(x)$ and $\text{N}\mathbf{C}_{\circ i}(x)$ as licensing and blocking structures for $[\text{nas}]_o(x)$, $[\text{voi}]_o(x)$, and $\text{out}(x)$.

	NC̥		NC̥	
	LICENSING	BLOCKING	LICENSING	BLOCKING
out	(meaningless)	nasal deletion (Swahili)	(meaningless)	consonant deletion (Indonesian)
$[\text{nas}]_o$	(meaningless)	denasalization (Mandar)	nasalization (Konjo)	(meaningless when $[-\text{son}] \rightarrow \neg[\text{nas}]$)
$[\text{voi}]_o$	(meaningless when $[\text{nas}] \rightarrow [\text{voi}]$)	nasal devoicing (Ndonga(?), Pokomo(?))	voicing (Quechua)	(meaningless)

TABLE 18. Summary of *NC̥ typology.

As a final example, we consider a case where both structures appear in the same grammar. In Umbundu, roots beginning with a voiceless obstruent surface with an initial nasal at the same place of articulation when inflected for first-person singular. A systematic exception is roots that begin with fricatives, which never nasalize. Schadeberg (1982) accounts for the pattern with an unspecified nasal prefix, as in the examples in 55.

(55) Umbundu (Bantu; Schadeberg 1982)
/Ntuma/ [numa] 'I send'
/Nseva/ [seva] 'I cook'

In a BMRS analysis, the pattern is captured by placing $\mathbf{NC}_{\circ i}(x)$ as a blocking structure for $\text{out}(x)$ AND $\text{N}\mathbf{C}_{\circ i}(x)$ as a licensing structure for $[\text{nas}]_o(x)$, as in 56.

(56) $\text{out}(x) = \text{if } \mathbf{NC}_{\circ i}(x) \text{ then } \bot \text{ else } \top$
$[\text{nas}]_o(x) = \text{if } [\text{cont}]_i(x) \text{ then } \bot \text{ else if } \text{N}\mathbf{C}_{\circ i}(x) \text{ then } \top \text{ else } [\text{nas}]_i(x)$
$[\text{voi}]_o(x) = [\text{voi}]_i(x)$

In /Ntuma/ 'I send', the initial /N/ returns $\bot$ for $\text{out}(x)$ and is thus deleted; additionally, the /t/ is also nasalized. This yields the output [numa]. The fact that noncontinuant consonants nasalize after a nasal is captured in 56 by placing $\text{N}\mathbf{C}_{\circ i}(x)$ as a licensing structure for $[\text{nas}]_o(x)$. That continuants exceptionally do NOT nasalize is captured by placing the blocking structure $[\text{cont}]_i(x)$ above the licensing structure $\text{N}\mathbf{C}_{\circ i}(x)$. Thus, for /Nseva/ 'I cook', the nasal deletes, but the continuant does not nasalize, yielding [seva].

In this section we have demonstrated how BMRSs can derive typological generalizations by varying a marked structure as a blocking or licensing structure for different output predicates. Of course, as in OT, this approach faces a 'too many solutions' problem. As Pater (1999) points out, epenthesis is never attested as a repair for NC̥ se-

quences, 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.

5. DISCUSSION. The analyses presented in the previous sections have demonstrated several of the strengths of the BMRS formalism: namely, it 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 article 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 & Kay 1981, 1994) and OT grammars (e.g. Karttunen 1993, Frank & Satta 1998, Riggle 2004, Gerdemann & Hulden 2012). 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 & Walker 2004). In Kikongo, a suffix liquid becomes nasal following a nasal in the root, regardless of how many other segments intervene. Examples are given in 57.

(57) Kikongo (Niger-Congo; Ao 1991, Odden 1994)

a.	/sakid-ila/	[sakid-ila]	‘congratulate for’
b.	/ku-toot-ila/	[ku-toot-ila]	‘to harvest for’
c.	/ku-kin-ila/	[ku-kin-ina]	‘to dance for’
d.	/ku-dumuk-ila/	[ku-dumuk-ina]	‘to jump for’
e.	/ku-dumuk-is-ila/	[ku-dumuk-is-ina]	‘to make jump for’

To capture this long-distance pattern with a BMRS system of equations, we first need a recursively defined predicate follows-$[\text{nas}]_i(x)$, as in 58.

(58) follows-$[\text{nas}]_i(x)$ = if $\rtimes_i(x)$ then $\bot$ else
if $[\text{nas}]_i(p(x))$ then $\top$ else
follows-$[\text{nas}]_i(p(x))$

Note that follows-$[\text{nas}]_i(x)$ is explicitly recursive as it refers to itself: follows-$[\text{nas}]_i(x)$ is $\bot$ if x is the starting word boundary, $\top$ if the immediate predecessor of x is nasal, or $\top$ if follows-$[\text{nas}]_i(x)$ is true of x’s predecessor. Thus, it will recurse backward in a word until it finds a nasal segment (and then returns $\top$) or finds the initial word boundary (and then returns $\bot$).

We can use this predicate to define a predicate N…$\mathbf{L}_i(x)$, which identifies a liquid that follows a nasal anywhere in the word.

(59) $\mathrm{N}\ldots\mathbf{L}_i(x) = \text{if } [\text{lat}]_i(x) \text{ then follows-}[\text{nas}]_i(x) \text{ else } \bot$

The transformation then is captured with the output predicates in 60, in which the predicate in 59 serves as a licensing structure for nasality.

(60) $[\text{nas}]_o(x) = \text{if } \mathrm{N}\ldots\mathbf{L}_i(x) \text{ then } \top \text{ else } [\text{nas}]_i(x)$

$[\text{lat}]_o(x) = \text{if } [\text{nas}]_o(x) \text{ then } \bot \text{ else } [\text{lat}]_i(x)$

Note that, because it is explicitly recursive, follows-$[\text{nas}]_i(x)$ 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 make reference only 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 like in Kikongo. Namely, there is an explicitly recursive predicate that is part of the BMRS system of equations but is not itself an output predicate. This is a point that warrants further work, particularly in the context of how to best characterize the distinction between local and long-distance phonology.

Finally, recursive schemes are flexible enough that they can be defined to hew more closely to phonological representation. For example, instead of functions that return Boolean values, we can define featural functions $\mathcal{J} = \{[\text{F}]_i(t), [\text{G}]_i(t), \ldots, [\text{Z}]_i(t)\}$ that take values in $\{+, -, 0\}$. Doing so would allow featural predicates to make distinctions between binary and privative features.

For example, a word-final devoicing definition would work as in 61.

(61) $[\text{voi}]_o(x) = \begin{bmatrix}\mathbf{-son}\\ \mathbf{+voi}\end{bmatrix}\ltimes_i(x) \text{ then } - \text{ else } [\text{voi}]_i(x)$

This predicate states that the value of the feature voice for x in the output is '−' if it is a word-final voiced obstruent; otherwise it takes whatever value x has for voice in the input (i.e. it is output faithfully).

We would, however, still need expressions that evaluate to Boolean values: the evaluation of the 'if E_1 then E_2 else E_3' syntax requires E_1 to be a Boolean expression (otherwise the if/then/else logical control cannot function). We can accomplish this by adding to our formalism the (standard; see e.g. Enderton 1972) single Boolean expression $E_1 \approx E_2$, which takes two expressions that evaluate to one of $\{+, -, 0\}$ and returns $\top$ iff E_1 and E_2 return the same value, $\bot$ otherwise. For example, $[\text{F}]_i(x) \approx +$ returns $\top$ if $[\text{F}]_i(x)$ evaluates to +; otherwise it returns $\bot$. From this we can define more complex Boolean expressions such as in 62.

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

While this approach would change the details of the definitions of predicates such as $\begin{bmatrix}-\text{son}\\ +\text{voi}\end{bmatrix}_i(x)$, the definitions of maps (such as the final devoicing map in 8) 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 (i.e. the predicates still take a single argument), and thus there would be no change in the expressiveness of the formalism.

Finally, extending recursive schemes to representations that explicitly include binary relations, such as association in autosegmental representations (Goldsmith 1976, Coleman & Local 1991) or dominance in the prosodic hierarchy (Selkirk 1980, Nespor & Vogel 1986), would necessitate allowing binary predicates and functions (such as e.g. $A(x, y)$) to be true iff x and y are associated. The consequences of this remain to be explored; however, recent research on logical transductions in such representations (Strother-Garcia 2017, Chandlee & Jardine 2019, Koser et al. 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 article 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 article demonstrate that this formalism shows great promise as a theory of the computational nature of phonological transformations.

REFERENCES

AO, BENJAMIN. 1991. Kikongo nasal harmony and context-sensitive underspecification. *Linguistic Inquiry* 22.193–96. Online: https://www.jstor.org/stable/4178713.

BAKER, MARK C. 2009. Language universals: Abstract but not mythological. *Behavior and Brain Sciences* 32.448–49. DOI: 10.1017/S0140525X09990604.

BAKOVIĆ, ERIC. 2000. *Harmony, dominance and control*. New Brunswick, NJ: Rutgers University dissertation. DOI: 10.7282/T3TQ60BJ.

BAKOVIĆ, ERIC. 2006. Elsewhere effects in optimality theory. *Wondering at the natural fecundity of things: Essays in honor of Alan Prince*, ed. by Eric Baković, Junko Ito, and John J. McCarthy, 23–70. Santa Cruz: University of California, Santa Cruz. Online: https://escholarship.org/uc/item/1m56m1ht.

BAKOVIĆ, ERIC. 2013. *Blocking and complementarity in phonological theory*. (Advances in optimality theory.) London: Equinox.

BHASKAR, SIDDHARTH; JANE CHANDLEE; ADAM JARDINE; and CHRISTOPHER OAKDEN. 2020. Boolean monadic recursive schemes as a logical characterization of the subsequential functions. *Language and Automata Theory and Applications (LATA 2020)*, 157–69. DOI: 10.1007/978-3-030-40608-0_10.

CHANDLEE, JANE. 2017. Computational locality in morphological maps. *Morphology* 27. 599–641. DOI: 10.1007/s11525-017-9316-9.

CHANDLEE, JANE, and JEFFREY HEINZ. 2018. Strict locality and phonological maps. *Linguistic Inquiry* 49.23–60. DOI: 10.1162/LING_a_00265.

CHANDLEE, JANE; JEFFREY HEINZ; and ADAM JARDINE. 2018. Input strictly local opaque maps. *Phonology* 35.171–205. DOI: 10.1017/S0952675718000027.

CHANDLEE, JANE, and ADAM JARDINE. 2019. Autosegmental input strictly local functions. *Transactions of the Association for Computational Linguistics* 7.157–68. DOI: 10.1162/tacl_a_00260.

CHOMSKY, NOAM. 1956. Three models for the description of language. *IRE Transactions on Information Theory* 2.113–24. DOI: 10.1109/TIT.1956.1056813.

CHOMSKY, NOAM. 1957. *Aspects of the theory of syntax*. The Hague: Mouton.

CHOMSKY, NOAM, and MORRIS HALLE. 1968. *The sound pattern of English*. New York: Harper & Row.

CHOTI, JONATHAN. 2015. Phonological asymmetries of Bantu nasal prefixes. *Annual Conference on African Linguistics* 44.37–51. Online: http://www.lingref.com/cpp/acal/44/paper3125.pdf.

COLEMAN, JOHN, and JOHN LOCAL. 1991. The 'no crossing constraint' in autosegmental phonology. *Linguistics and Philosophy* 14.295–338. DOI: 10.1007/BF00627405.

COURCELLE, BRUNO. 1994. Monadic second-order definable graph transductions: A survey. *Theoretical Computer Science* 126.53–75. DOI: 10.1016/0304-3975(94)90268-2.

DALY, RICHARD T. 1974. *Applications of the mathematical theory of linguistics*. The Hague: Mouton.
DERBYSHIRE, DESMOND C. 1985. *Hixkaryana and linguistic typology*. Dallas: SIL International.
DURVASULA, KARTHIK. 2010. *Understanding nasality*. Newark: University of Delaware dissertation.
EISNER, JASON. 1997. Efficient generation in primitive optimality theory. *35th annual meeting of the Association for Computational Linguistics and 8th Conference of the European Chapter of the Association for Computational Linguistics*, 313–20. DOI: 10.3115/976909.979657.
EISNER, JASON. 2000. Directional constraint evaluation in optimality theory. *Proceedings of the 18th Conference on Computational Linguistics (COLING 2000)*, 257–63. DOI: 10.3115/990820.990858.
ENDERTON, HERBERT. 1972. *A mathematical introduction to logic*. Cambridge, MA: Academic Press.
ENGELFRIET, JOOST, and HENDRIK JAN HOOGEBOOM. 2001. MSO definable string transductions and two-way finite-state transducers. *ACM Transactions on Computational Logic* 2.216–54. DOI: 10.1145/371316.371512.
FRANK, ROBERT, and GIORGIO SATTA. 1998. Optimality theory and the generative complexity of constraint violability. *Computational Linguistics* 24.307–15. Online: https://www.aclweb.org/anthology/J98-2006.
FRIBERG, TIMOTHY, and BARBARA FRIBERG. 1991. Notes on Konjo phonology. *Studies in Sulawesi linguistics, part II*, ed. by James Neil Sneddon, 71–117. (NUSA linguistic studies of Indonesian and other languages in Indonesia 33.) Jakarta: Universitas Katolik Indonesia.
GERDEMANN, DALE, and MANS HULDEN. 2012. Practical finite state optimality theory. *Proceedings of the 10th International Workshop on Finite State Methods and Natural Language*, 10–19. Online: http://www.aclweb.org/anthology/W12-6202.
GOLD, MARK E. 1967. Language identification in the limit. *Information and Control* 10.447–74. DOI: 10.1016/S0019-9958(67)91165-5.
GOLDSMITH, JOHN. 1976. *Autosegmental phonology*. Cambridge, MA: MIT dissertation. Online: http://www.ai.mit.edu/projects/dm/theses/goldsmith76.pdf.
GORDON, MATTHEW. 2002. A factorial typology of quantity-insensitive stress. *Natural Language and Linguistic Theory* 20.491–552. DOI: 10.1023/A:1015810531699.
GRAF, THOMAS. 2019. A subregular bound on the complexity of lexical quantifiers. *Proceedings of the 22nd Amsterdam colloquium*, ed. by Julian J. Schlöder, Dean McHugh, and Floris Roelofsen, 455–64.
GRAF, THOMAS. 2020. Curbing feature coding: Strictly local feature assignment. *Proceedings of the Society for Computation in Linguistics* 3:35. DOI: 10.7275/f7y5-xz32.
GRAF, THOMAS, and NAZILA SHAFIEI. 2019. C-command dependencies as TSL string constraints. *Proceedings of the Society for Computation in Linguistics* 2:22. DOI: 10.7275/4rrx-x488.
HALLE, MORRIS. 1995. Comments on Burzio (1995). *Glot International* 1.27–28.
HALLE, MORRIS, and WILLIAM J. IDSARDI. 1998. A response to Alan Prince's letter. *Glot International* 3.1–22.
HALLE, MORRIS, and WILLIAM J. IDSARDI. 2000. Stress and length in Hixkaryana. *The Linguistic Review* 17.199–218. DOI: 10.1515/tlir.2000.17.2-4.199.
HAYES, BRUCE. 1995. *Metrical stress theory*. Chicago: The University of Chicago Press.
HEINZ, JEFFREY. 2009. On the role of locality in learning stress patterns. *Phonology* 26.303–51. DOI: 10.1017/S0952675709990145.
HEINZ, JEFFREY. 2010. Learning long-distance phonotactics. *Linguistic Inquiry* 41.623–61. DOI: 10.1162/LING_a_00015.
HEINZ, JEFFREY. 2018. The computational nature of phonological generalizations. *Phonological typology*, ed. by Larry Hyman and Frans Plank, 126–95. Berlin: De Gruyter Mouton. DOI: 10.1515/9783110451931-005.
HEINZ, JEFFREY. 2021. *Doing computational phonology*. Oxford: Oxford University Press, to appear.
HEINZ, JEFFREY, and WILLIAM IDSARDI. 2011. Sentence and word complexity. *Science* 333.295–97. DOI: 10.1126/science.1210358.

HEINZ, JEFFREY, and WILLIAM IDSARDI. 2013. What complexity differences reveal about domains in language. *Topics in Cognitive Science* 5.111–31. DOI: 10.1111/tops.12000.

HEINZ, JEFFREY, and CESAR KOIRALA. 2010. Maximum likelihood estimation of feature-based distributions. *Proceedings of the 11th Meeting of the ACL Special Interest Group on Computational Morphology and Phonology*, 28–37. Online: https://www.aclweb.org/anthology/W10-2204.

HEINZ, JEFFREY, and REGINE LAI. 2013. Vowel harmony and subsequentiality. *Proceedings of the 13th Meeting on Mathematics of Language*, 52–63. Online: https://www.aclweb.org/anthology/W13-3006.

HERBERT, ROBERT K. 1986. *Language universals, markedness theory, and natural phonetic processes*. Berlin: Mouton de Gruyter.

HINNEBUSCH, THOMAS J. 1975. A reconstructed chronology of loss: Swahili class 9/10. *Ohio State Working Papers in Linguistics (Proceedings of the sixth Conference on African Linguistics*) 20.32–41. Online: http://hdl.handle.net/1811/81394.

HUFFMAN, MARIA, and THOMAS HINNEBUSCH. 1998. The phonetic nature of 'voiceless' nasals in Pokomo: Implications for sound change. *Journal of African Languages and Linguistics* 19.1–19. DOI: 10.1515/jall.1998.19.1.1.

IKAWA, SHIORI; AKANE OHTAKA; and ADAM JARDINE. 2020. Quantifier-free tree transductions. *Proceedings of the Society for Computation in Linguistics* 3:49. DOI: 10.7275/k1gj-bq24.

IMMERMAN, NEIL. 1980. *First-order expressibility as a new complexity measure*. Ithaca, NY: Cornell University dissertation. Online: https://hdl.handle.net/1813/6272.

JARDINE, ADAM. 2016. Computationally, tone is different. *Phonology* 33.247–83. DOI: 10.1017/S0952675716000129.

JARDINE, ADAM. 2017a. The local nature of tone-association patterns. *Phonology* 34.363–84. DOI: 10.1017/S0952675717000185.

JARDINE, ADAM. 2017b. On the logical complexity of autosegmental representations. *Proceedings of the 15th Meeting on the Mathematics of Language*, 22–35. DOI: 10.18653/v1/W17-3403.

JARDINE, ADAM. 2019. The expressivity of autosegmental grammars. *Journal of Logic, Language, and Information* 28.9–54. DOI: 10.1007/s10849-018-9270-x.

JARDINE, ADAM; JANE CHANDLEE; RÉMI EYRAUD; and JEFFREY HEINZ. 2014. Very efficient learning of structured classes of subsequential functions from positive data. *Proceedings of Machine Learning Research (The 12th International Conference on Grammatical Inference)* 34.94–108. Online: http://proceedings.mlr.press/v34/jardine14a.html.

JI, JING, and JEFFREY HEINZ. 2020. Input strictly local tree transducers. *Language and Automata Theory and Applications (LATA 2020)*, 369–81. DOI: 10.1007/978-3-030-40608-0_26.

JOHNSON, C. DOUGLAS. 1972. *Formal aspects of phonological description*. The Hague: Mouton.

KAGER, RENÉ. 1999. *Optimality theory*. Cambridge: Cambridge University Press.

KAGER, RENÉ. 2007. Feet and metrical stress. *The Cambridge handbook of phonological theory*, ed. by Paul de Lacy, 195–227. Cambridge: Cambridge University Press.

KAPLAN, RONALD M., and MARTIN KAY. 1981. Phonological rules and finite-state transducers. Abstract of paper presented at the 56th annual meeting of the Linguistic Society of America, New York. Online: https://www.linguisticsociety.org/sites/default/files/1981_searchable.pdf.

KAPLAN, RONALD M., and MARTIN KAY. 1994. Regular models of phonological rule systems. *Computational Linguistics* 20.331–78. Online: https://www.aclweb.org/anthology/J94-3001.

KARTTUNEN, LAURI. 1993. Finite-state constraints. *The last phonological rule*, ed. by John Goldsmith, 173–94. Chicago: The University of Chicago Press.

KARTTUNEN, LAURI. 1998. The proper treatment of optimality in computational phonology. *Finite State Methods in Natural Language Processing* 1998.1–12. Online: https://www.aclweb.org/anthology/W98-1301.

KENSTOWICZ, MICHAEL. 1994. *Phonology in generative grammar*. Hoboken, NJ: Blackwell.

KIPARSKY, PAUL. 1973. Abstractness, opacity, and global rules. *Three dimensions in linguistic theory*, ed. by Osamu Fujimura, 57–86. Tokyo: TEC.

KOBELE, GREGORY M. 2006. *Generating copies: An investigation into structural identity in language and grammar*. Los Angeles: University of California, Los Angeles dissertation. Online: https://linguistics.ucla.edu/general/dissertations/Kobele06GeneratingCopies.pdf.

KOSER, NATHAN; CHRISTOPHER OAKDEN; and ADAM JARDINE. 2019. Tone association and output locality in non-linear structures. *Proceedings of the 2018 Annual Meeting on Phonology*. DOI: 10.3765/amp.v7i0.4476.

KOUTSOUDAS, ANDREAS; GERALD SANDERS; and CRAIG NOLL. 1974. The application of phonological rules. *Language* 50.1–28. DOI: 10.2307/412007.

LAMONT, ANDREW. 2019. Precedence is pathological: The problem of alphabetical sorting. *West Coast Conference on Formal Linguistics (WCCFL)* 36.243–49. Online: http://www.lingref.com/cpp/wccfl/36/paper3468.pdf.

LUO, HUAN. 2017. Long-distance consonant agreement and subsequentiality. *Glossa: A Journal of General Linguistics* 2(1):52. DOI: 10.5334/gjgl.42.

MCCOLLUM, ADAM G.; ERIC BAKOVIĆ; ANNA MAI; and ERIC MEINHARDT. 2020. Unbounded circumambient patterns in segmental phonology. *Phonology* 37.215–55. DOI: 10.1017/S095267572000010X.

MILLS, ROGER F. 1975. *Proto South Sulawesi and proto Austronesian phonology*. Ann Arbor: University of Michigan dissertation.

MOHRI, MEHRYAR. 1997. Finite-state transducers in language and speech processing. *Computational Linguistics* 23.269–311. Online: https://www.aclweb.org/anthology/J97-2003.

MOSCHOVAKIS, YIANNIS N. 2019. *Abstract recursion and intrinsic complexity*. (Lecture notes in logic 48.) Cambridge: Cambridge University Press. DOI: 10.1017/9781108234238.

MYERS, SCOTT. 1987. Vowel shortening in English. *Natural Language and Linguistic Theory* 5.485–518. DOI: 10.1007/BF00138987.

NESPOR, MARINA, and IRENE VOGEL. 1986. *Prosodic phonology*. Dordrecht: Foris.

ODDEN, DAVID. 1982. Tonal phenomena in Kishambaa. *Studies in African Linguistics* 13.177–208. Online: https://journals.linguisticsociety.org/elanguage/sal/article/view/1117.html.

ODDEN, DAVID. 1994. Adjacency parameters in phonology. *Language* 70.289–330. DOI: 10.2307/415830.

ONCINA, JOSÉ; PEDRO GARCÍA; and ENRIQUE VIDAL. 1993. Learning subsequential transducers for pattern recognition tasks. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 15.448–58. DOI: 10.1109/34.211465.

ONN, FARID M. 1980. *Aspects of Malay phonology and morphology: A generative approach*. Kuala Lumpur: Universiti Kebangsaan Malaysia.

ORR, CAROLYN. 1962. Ecuador Quichua phonology. *Studies in Ecuadorian Indian languages*, ed. by Benjamin Elson, 60–77. Norman, OK: Summer Institute of Linguistics.

PATER, JOE. 1999. Austronesian nasal substitution and other NÇ effects. *The prosody-morphology interface*, ed. by René Kager, Harry van der Hulst, and Wim Zonneveld, 310–43. Cambridge: Cambridge University Press.

PATER, JOE. 2018. Substance matters: A reply to Jardine (2016). *Phonology* 35.151–56. DOI: 10.1017/S0952675717000409.

PAYNE, AMANDA. 2017. All dissimilation is computationally subsequential. *Language* 93.e353–e371. DOI: 10.1353/lan.2017.0076.

PRINCE, ALAN S. 1980. A metrical theory for Estonian quantity. *Linguistic Inquiry* 14.511–62. Online: https://www.jstor.org/stable/4178178.

PRINCE, ALAN S. 1983. Relating to the grid. *Linguistic Inquiry* 14.19–100. Online: https://www.jstor.org/stable/4178311.

PRINCE, ALAN, and PAUL SMOLENSKY. 1993. Optimality theory: Constraint interaction in generative grammar. RuCCS Technical Report. New Brunswick, NJ: Rutgers University.

PRINCE, ALAN, and PAUL SMOLENSKY. 2004. *Optimality theory: Constraint interaction in generative grammar*. Hoboken, NJ: Blackwell.

PULLUM, GEOFFREY K. 2011. On the mathematical foundations of *Syntactic structures*. *Journal of Logic, Language and Information* 20.277–96. DOI: 10.1007/s10849-011-9139-8.

PULLUM, GEOFFREY K., and GERALD GAZDAR. 1982. Natural languages and context-free languages. *Linguistics and Philosophy* 4.471–504. DOI: 10.1007/BF00360802.
RIGGLE, JASON. 2004. *Generation, recognition, and learning in finite state optimality theory*. Los Angeles: University of California, Los Angeles dissertation. Online: https://linguistics.ucla.edu/general/dissertations/riggle/riggle04_1up.pdf.
ROGERS, JAMES; JEFFREY HEINZ; MARGARET FERO; JEREMY HURST; DAKOTAH LAMBERT; and SEAN WIBEL. 2013. Cognitive and sub-regular complexity. *Formal Grammar: FG 2013, FG 2012*, ed. by Glyn Morrill and Mark-Jan Nederhof, 90–108. New York: Springer. DOI: 10.1007/978-3-642-39998-5_6.
ROSE, SHARON, and RACHEL WALKER. 2004. A typology of consonant agreement as correspondence. *Language* 80.475–531. DOI: 10.1353/lan.2004.0144.
SANDERS, GERALD A. 1974. Precedence relations in language. *Foundations of Language* 11.361–400. Online: https://www.jstor.org/stable/25000783.
SCHADEBERG, THILO C. 1982. Nasalization in Umbundu. *Journal of African Languages and Linguistics* 4.109–32. DOI: 10.1515/jall.1982.4.2.109.
SELKIRK, ELISABETH O. 1980. The role of prosodic categories in English word stress. *Linguistic Inquiry* 11.563–605. Online: https://www.jstor.org/stable/4178179.
SELKIRK, ELISABETH O. 1984. On the major class features and syllable theory. *Language sound structure: Studies in phonology*, ed. by Morris Halle, Mark Aronoff, and Richard T. Oehrle, 107–13. Cambridge, MA: MIT Press.
SHIEBER, STUART M. 1985. Evidence against the context-freeness of natural language. *Linguistics and Philosophy* 8.333–43. DOI: 10.1007/BF00630917.
SMITH, CAITLIN, and CHARLIE O'HARA. 2019. Formal characterizations of true and false sour grapes. *Proceedings of the Society for Computation in Linguistics* 2:41. DOI: 10.7275/vd79-kt51.
STROTHER-GARCIA, KRISTINA. 2017. Imdlawn Tashlhiyt Berber syllabification is quantifier-free. *Proceedings of the Society for Computation in Linguistics* 1:16. DOI: 10.7275/R5J67F4D.
STROTHER-GARCIA, KRISTINA. 2019. *Using model theory in phonology: A novel characterization of syllable structure and syllabification*. Newark: University of Delaware dissertation. Online: http://udspace.udel.edu/handle/19716/25084.
STROTHER-GARCIA, KRISTINA; JEFFREY HEINZ; and HYUN JIN HWANGBO. 2016. Using model theory for grammatical inference: A case study from phonology. *Proceedings of Machine Learning Research (Proceedings of the 13th International Conference on Grammatical Inference)* 57.66–78. Online: http://proceedings.mlr.press/v57/strother-garcia16.html.
VILJOEN, JOHANNES, and P. AMAKALI. 1978. *A handbook of Oshiwambo*. Pretoria: University of South Africa.
VU, MAI HA; NAZILA SHAFIEI; and THOMAS GRAF. 2019. Case assignment in TSL syntax: A case study. *Proceedings of the Society for Computation in Linguistics* 2:28. DOI: 10.7275/sywz-xw23.
WALKER, RACHEL. 2001. Mongolian stress, licensing, and factorial typology. Santa Cruz: University of California, Santa Cruz, MS. Online: http://roa.rutgers.edu/article/view/183, version date 27 July 2001.
WILSON, COLIN. 2003. Analyzing unbounded spreading with constraints: Marks, targets, and derivations. Baltimore: Johns Hopkins University, MS.
WILSON, COLIN. 2006. Unbounded spreading is myopic. Paper presented at the Current Perspectives on Phonology workshop, Phonology Fest, Indiana University.
YIP, MOIRA. 2002. *Tone*. Cambridge: Cambridge University Press.
ZALESKA, JOANNA. 2018. *Coalescence without coalescence*. Leipzig: Universität Leipzig dissertation.

[jchandlee@haverford.edu]
[adam.jardine@rutgers.edu]

[Received 24 March 2020;
revision invited 14 July 2020;
revision received 7 October 2020;
accepted pending revisions 29 November 2020;
revision received 28 December 2020;
accepted 6 January 2021]