THE COMPUTATIONAL NATURE OF STRESS ASSIGNMENT

By

NATHAN DAVID KOSER

A dissertation submitted to the

School of Graduate Studies

Rutgers, The State University of New Jersey

In partial fulfillment of the requirements

For the degree of

Doctor of Philosophy

Graduate Program in Linguistics

Written under the direction of

Dr. Adam Jardine

And approved by

New Brunswick, New Jersey

October, 2022

ABSTRACT OF THE DISSERTATION

The Computational Nature of Stress Assignment

by NATHAN DAVID KOSER

Dissertation Director:

Dr. Adam Jardine

This dissertation articulates a restrictive theory of stress based on formal language theoretic complexity. It demonstrates that stress patterns that appear more complex as a single step from input to output can be profitably broken down into less-complex stress primitives or "atoms" that combine to derive the observed surface pattern. Though the idea that surface stress patterns are the combination of more fine-grained generalizations is not new, the theory proposed here uses independently motivated categories from computational complexity to provide a well-defined notion of the nature of these generalizations and the exact computational power at which they operate. An assertion regarding computational power is a hypothesis as to why stress patterns take the form that they do – they are beholden to certain computational restrictions that limits the range of patterns that may be defined. Thus, defining the formal properties of stress atoms provides a metric for evaluating what an attested stress pattern can be, improving the predicted typology.

This dissertation models stress as a string-to-string mapping from input to output, where the input is a bare string of syllables and the output is marked with stress. Patterns which are formally more complex when viewed as a single function break down into atoms corresponding to much simpler function classes. Iteration of stress is handled by a single *output strictly local* function (OSL; Chandlee 2014; Chandlee and Heinz 2018), while other stress properties such as non-finality are handled by an "edge-oriented" (EO) function, a novel function class defined in this thesis. EO functions encode the fact that stress generaliza-

tions such as non-finality are tethered to a word edge, and thus provide a restriction based on substantive stress properties couched in computational terms. Long-distance patterns break down into combinations of EO and *strictly piecewise* functions (SP; Rogers et al. 2010; Burness and McMullin 2020), which characterize long-distance processes without the overgeneration of more complex function classes. The stress component of the phonological grammar is thus comprised of atoms that are maximally OSL, EO, or SP, which are composed in a specific sequence. This hypothesis on the computational nature of stress is expressive enough to describe attested patterns, but sufficiently restrictive to eliminate many patterns that are logically possible but non-phonological, such as patterns that require explicit parity counting.

Contents

CHAPTER 1: INTRODUCTION

1.1 Introduction

This dissertation presents a restrictive theory of the stress patterns of the world's languages based on their computational properties in an effort to answer the question – what is a possible stress pattern? What separates an impossible, unattested pattern from the attested patterns? I demonstrate that a profitable way to address this question is by factoring stress patterns into their simplest composite parts, or "atoms". Specifically, the theory proposed here is that stress generalizations are composed of one or two simple individual *functions* that are maximally output strictly local (OSL; Chandlee et al. 2015; Chandlee and Heinz 2018), strictly piecewise (Rogers et al. 2010; Burness and McMullin 2020), or "edge-oriented" (EO), a novel contribution of this work. The surface patterns we observe are thus combinations of these individual atoms composed in sequence. By stating computational limitations on the possible atoms that may participate in a stress generalization, this work offers clear typological predictions of possible and impossible patterns. I show that this restriction on the stress primitives themselves based on their computational properties leads to measurable improvements in the predicted typology compared to a theory that allows for more expressive atomic generalizations. For example, I show in detail why allowing properly subsequential atoms (Schutzenberger 1977; Mohri 1997) overgenerates in a way that the more restrictive theory pursued here does not.

The question of what properties differentiate possible and impossible stress patterns has been approached from many different angles, including rule-based and parametric approaches (Chomsky and Halle 1968; Booij 1983; Halle and Vergnaud 1987; Idsardi 1992; Hayes 1995), constraint-based approaches (Bailey 1995; Hyde 2002; Gordon 2002; Kager 2005; Buckley 2009; Kager 2012), and computational approaches (Heinz 2007a; Rogers et al. 2013; Heinz 2014; Baek 2018; Hao and Anderson 2019; Koser and Jardine 2020b; Rogers and Lambert 2019). This vast body of previous work shares several important insights. First, they identify a need for restrictiveness in phonological theory. This stems from the observation that not all logically possible patterns that can be imagined are in fact attested phonological patterns. A stress pattern that stresses each syllable corresponding to a number in the Fibonacci sequence, for example, or a pattern that sorts all stressed syllables to the front of the word – these are both logically possible patterns that any phonologist will agree are undesirable predictions for phonological typology. So, while a hypothesis that any possible pattern is phonologically natural includes all attested patterns, it does so in way that egregiously overgenerates and leaves the fascinating properties that separate attested and unattested linguistic patterns completely unexplored.

Previous work on phonology from the computational perspective offers an insightful account of how the presence of some patterns and absence of others can be explained. Relying on the explicitly-defined notion of complexity offered by formal language theory, research in this vein seeks to identify the computational properties that linguistic patterns have. This allows for restrictive hypotheses based on formal complexity that directly addresses not just what form surface patterns have, but *why* they appear in the form that they do. For example, previous work has identified that phonological patterns are at most *regular* (Johnson 1972; Kaplan and Kay 1994). Regular patterns are those describable with a finite state machine. Intuitively, phonology is regular because calculation of a phonological pattern relies on a fixed, finite amount of memory. Further work has shown that even finer distinctions can be made, as the majority of phonological processes are in fact *subregular* – they belong to an even more restrictive complexity class (Rogers et al. 2013; Heinz 2018).

Connecting phonological processes to subregular classes provides a stronger hypothesis for those processes and improves the accuracy of the predicted typology.

Studies of stress and subregular complexity attempt to describe the upper bound of complexity in stress generalizations, as well as characterize the computational properties of stress in a way that is maximally restrictive (Heinz 2014; Rogers et al. 2013; Baek 2018; Rogers and Lambert 2019; Hao and Anderson 2019; Koser and Jardine 2020b). Thus, a proposal about the complexity of stress or some subcategory of stress patterns is a hypothesis that all patterns of that type fall within that complexity boundary. This thesis argues that the best characterization of the complexity of stress patterns is one that breaks the surface patterns down into the individual atoms that comprise them. This atomic approach to stress thus adopts another intuition shared by both subregular phonology and more traditional analyses – that more complex patterns can be broken down into simpler primitives, exposing the fundamental similarities of those patterns.

For example, on top of the theoretical devices that have been used to analyze stress, such as feet or metrical grids (Liberman and Prince 1977; Halle and Vergnaud 1978; Selkirk 1980; Prince 1983; Hammond 1984; Halle and Vergnaud 1987; Idsardi 1992; Hayes 1995), previous work often proposes special mechanisms to account for the patterns we observe and their particular details, such as rules of extrametricality (Liberman and Prince 1977; Hayes 1981) or the NON-FINALITY constraint of OT grammars (Prince and Smolensky 1993). This productive body of earlier work on stress thus shares the intuition of this dissertation that the surface patterns we observe are the sum of more fine-grained, primitive stress generalizations. It encapsulates a classic idea in phonology expressed succinctly by McCarthy (1988)'s claim that: "the goal of phonology is the construction of a theory in which cross-linguistically common and well-established processes emerge from very simple combinations of the descriptive parameters of the model" (p.84). Studies of formal complexity in linguistics help us evaluate how a preference for combinations of simpler generalizations – rather than a more complex single-step approach – directly affects the predicted typology of phonological patterns.

Joining this body of previous work on stress, the proposal here uses independently motivated categories from the theory of computation to provide a well-defined notion of the formal nature of mechanisms such as non-finality, as well as the exact computational power at which they operate. Proposing that stress patterns adhere to certain computational restrictions thus offers a hypothesis not only on what stress maps ought to look like, but *why* they take the form that they do – they are beholden to certain computational restrictions that can be expressed in the terms of formal language theory. It also contributes a precise computational characterization of the individual "atoms" present in stress typology and demonstrates that these properties only become apparent when stress is broken down into a series of steps. That a generalization like non-finality fits within the hypothesized computational limits for iterative stress thus lends support to such substantive mechanisms that have been previously proposed in the stress literature, but does so from the novel perspective of rigorously-defined computational complexity that formal language theory offers.

In this way, computational studies of stress invite a mutually beneficial relationship with work in other formalisms to provide a more holistic understanding of the nature of stress. For example, if some pathological pattern is not ruled out by a system of rules or constraints, but examination of its complexity indicates that it exceeds the hypothesized boundary for complexity in stress, then this offers an explanation to the pattern's absence, and may give insight into what restrictions can be implemented to remove it from the predicted typology. The reverse relationship can also obtain – hypotheses on complexity do not by themselves rule out all inaccurate typological predictions. For example, a stress pattern that stresses every syllable in the word is maximally simple from a computational perspective, but is unattested and a bad prediction. In such cases, an explanation that makes use of substantive aspects of phonological theory can and should be pursued. Reference to foot structure, culminativity, and other well-known theoretical devices provide their own notion of restrictiveness that can exist in tandem with purely computational restrictions to outline a maximally restrictive – yet descriptively adequate – typology.

1.2 Restrictiveness

As highlighted in (Heinz 2009), one goal of research in this vein is to determine what basic properties the stress assignment function has, and what separates a possible stress pattern from an impossible one. A main result of this thesis with regards to this goal is demonstrating that patterns that appear to require *subsequential* power (Schutzenberger 1977; Mohri 1997) as a single-function map from input to output are better analyzed as the composition of two functions that are formally less complex and subject to certain restrictions. Subsequential functions give rise to pathological predictions because, while they are subregular, they are still quite powerful. Any function that can be computed deterministically on its input in a single direction is a subsequential function. This allows for pathological patterns relying on overt numerical counts to be defined. A subsequential function could, for example, stress every third heavy syllable present in a string of light and heavy syllables. As such, subsequentiality is too *weak* of a hypothesis for the atomic functions comprising stress maps. Reference to more restrictive function classes is necessary.

For local, iterative patterns, an *output strictly local* function (OSL; Chandlee 2014; Chandlee et al. 2015; Chandlee and Heinz 2018) describes basic iteration of stress. The behavior of an OSL function is determined entirely by some local window in the output string – iteration of stress depends on the placement of previous stresses in the output, and so is OSL. These iteration functions are then composed with an "edge-oriented" (EO) function – a novel contribution of this dissertation. EO functions are local in that they are limited to apply only within a limited span anchored at a word edge. They implement a "cleanup" step that compensates when basic iteration is insufficient to describe the total pattern. For example, if the OSL function in the map iterates stress onto a final syllable, but the specific language is subject to a non-finality requirement, then the following EO function removes this final stress and completes the generalization. I show that not all subsequential functions can be described as the composition of an OSL and EO function, meaning that such compositions represent a stronger hypothesis for iterative stress.

For long-distance patterns such as default-to-same, default-to-opposite, or “suprabinary” patterns such as Pulaar (Niang 1997), a composition using one or two *strictly piecewise* functions (SP; Burness and McMullin 2020) is sufficient. SP functions model long distance patterns while avoiding issues with overgeneration that are possible with subsequential functions because the SP class is less expressive. Thus, the hypothesis that stress is the composition of OSL, EO, and SP atoms is preferable to one allowing subsequential power because it provides a more restrictive characterization of stress generalizations that eliminates pathological stress patterns which can be defined with just a single subsequential function. The potential for overgeneration via these compositions, and how it might be constrained, is discussed in Chapter 4. All stress patterns examined break down into compositions of individual atoms as follows:

(1)

Local:	**single stress**	**iterative stress**	**bidirectional stress**
	EO	OSL + EO	OSL + OSL
Non-local:	**DTO**	**DTS**	**suprabinary**
	EO + SP	SP + SP	SP + SP

Whereas previous work from formal language theory on the composition of stress primitives has focused on stress as formal language sets and the intersection of those sets (Heinz 2014; Rogers et al. 2013; Rogers and Lambert 2019), here the focus is on the composition of FLT *functions*. The analysis of stress as a mapping from the input through a series of functions to the output resembles classic analyses of stress that view it as a map from an underlying representation to a surface representation. Many of these analyses date back to the earliest days of phonology (Chomsky and Halle 1968; Liberman and Prince 1977; Halle and Vergnaud 1987; Idsardi 1992; Hayes 1995). What the approach taken here contributes is an exact formal characterization of the individual "atoms" of stress assignment and their typological consequences based on their computational properties. It helps explain *why* stress appears in the way that it does on the surface – the computational restrictions on stress mappings determine what is and is not a possible stress generalization.

As an example of the importance of restrictiveness, consider that while stress systems such as, "stress every other syllable starting from the first" or "stress every other syllable starting from the penult" appear perfectly reasonable and are in fact attested, there is no known stress system that follows a rule like "stress every other syllable starting from the middle". Pathological patterns that require locating the middle of the word have been referred to as the Midpoint Pathology (Eisner 1997; Hyde 2008; Kager 2012), an example of which is shown here:

(2) $\sigma\acute{\sigma}\sigma$

$\sigma\sigma\acute{\sigma}\sigma$

$\sigma\sigma\acute{\sigma}\sigma\sigma$

$\sigma\sigma\sigma\acute{\sigma}\sigma\sigma$

$\sigma\sigma\sigma\acute{\sigma}\sigma\sigma\sigma$

$\sigma\sigma\sigma\sigma\acute{\sigma}\sigma\sigma\sigma...$

Stress is fixed in the middle of the word. Though phonologists would generally agree that such a pattern is unattested and pathological, the formal notion of complexity available in FLT offers one explanation as to why that is the case: the generalization "find the middle" exceeds the proposed complexity threshold for phonological functions – it is not even a *regular* function (Eisner 1997)[1]. If phonology is at most regular, and the atoms of stress patterns belong to an even more restrictive class than that, it is natural that no stress patterns based on the middle of the word should exist.

However, a sufficiently restrictive theory of stress will enforce even more stringent requirements than being formally regular, as there are clearly pathological patterns that fall

[1] Stanton (2016) also offers an explanation from learnability.

within the regular boundary. One example is the following "sour-grapes"-like pattern (SG; Wilson 2003, 2006) for stress (Koser and Jardine 2020a,b). In such patterns, if a condition such as spread of a feature cannot be satisfied to the fullest possible extent, the condition is abandoned instead of applying partially:

(3) $\sigma\sigma\sigma$

$\acute{\sigma}\sigma\acute{\sigma}\sigma$

$\sigma\sigma\sigma\sigma\sigma$

$\acute{\sigma}\sigma\acute{\sigma}\sigma\acute{\sigma}\sigma$

$\sigma\sigma\sigma\sigma\sigma\sigma\sigma$

$\acute{\sigma}\sigma\acute{\sigma}\sigma\acute{\sigma}\sigma\acute{\sigma}\sigma$...

Here, stress only iterates through words of even parity. Though such a pattern is clearly pathological in that it relies on the parity of the entire word, it will be shown in Chapter 4 that it is not ruled out by the assumption that phonology itself is regular. However, I demonstrate that if iterative stress maps are restricted to the composition of OSL and EO functions, then this sour-grapes stress pattern *is* eliminated from the predicted typology. Thus, an important result of this thesis is a restrictive theory of stress based on its computational properties that makes testable predictions about what we should observe in the typology of natural language stress.

Though overgeneration occurs to some extent no matter the hypothesized level of complexity, making the most restrictive claim with regards to the expressive power of a category of phonological patterns leads to directly observable improvements in the hypothesized typology of those patterns. Where computational restrictions alone fail to eliminate unwanted patterns, arguments from particular theories of representation and phonological grammar are welcomed.

That iterative and long-distance patterns are subject to different computational requirements suggests a conceptual separation of stress typology into different categories with long-distance atoms and categories with only local atoms. I argue that this is a natural choice given the divergent nature of their computational properties – different computational characteristics lead to distinct typological predictions and differing notions of how to constrain overgeneration. If a given formalism treats two patterns with distinct computational properties as the same, then that is a facet of the patterns as they exist in that formalism, not of the abstract properties of the patterns themselves. On the other hand, the complexity of a pattern cannot be examined at all without adopting some representational assumptions, highlighting the fact that studies of computation in linguistics and more substantive proposals with regards to theoretical devices should enjoy a mutually beneficial interchange of insight that improves the quality of work being done in both domains.

1.3 Outline

Chapter 2 has two objectives. First, it establishes the representational assumptions that are made with regards to stress throughout the thesis. These assumptions are then justified based on their consequences for the computational study of stress patterns in later chapters. Second, the chapter sets the empirical background for the study of the world's stress patterns. This involves a detailed description of the patterns that are central to the proceeding analyses in the dissertation. Patterns are categorized based on their iterativity, directionality, and locality.

Chapter 3 provides the necessary formal background for the computational analyses throughout the dissertation. It begins with an overview of finite state transducers and how they can be used to study phonological processes such as stress as a mapping from an input string to an output string. It then gives an overview of the complexity classes of formal language theory that are relevant to work in this thesis. These include the input and output strictly local classes (Chandlee 2014; Chandlee and Heinz 2018), the strictly piecewise function class (Burness and McMullin 2020), and the subsequential class (Schutzenberger 1977; Mohri 1997). Special attention is paid to the ability of subsequential functions to overgenerate when adopted as a hypothesis for atoms of stress generalizations.

Chapter 4 outlines the main proposal of the dissertation, introducing the edge-oriented (EO) class. EO functions are more restrictive than either the OSL or ISL functions as they are limited to apply only within some fixed distance of the word edge. I demonstrate that they provide a better hypothesis for the cleanup step of compositional analyses of stress. Chapter 4 also provides a detailed overview of function composition and the typological consequences of adopting a compositional analysis of the necessary function classes.

Chapter 5 turns to an analysis of unidirectional iterative stress patterns, both quantity insensitive and quantity sensitive. It is shown that, though some patterns require a subsequential power as a single function, decomposition of the pattern into its primitive atoms allows it to be described in a manner that is formally local and provides better typological predictions. The work presented here emphasizes that, despite an array of surface differences, the full range of iterative patterns are beholden to the same computational restrictions. This suggests a new conception of stress typology based on the computational properties of their composite stress atoms, rather than what we observe on the surface.

Chapter 6 provides an analysis of bidirectional iterative patterns and long-distance patterns. Like their local, unidirectional counterparts, these are also subject to an atomic analysis based on their computational properties that results in a more restrictive theory than one allowing the expressive power of the corresponding single-function map by invoking SP functions.

This combined with the work of Chapter 5 shows that stress generalizations are composed of relatively simple individual functions that are maximally OSL, EO, or SP. The differing computational properties of long-distance patterns suggests that stress typology may be fruitfully divided into local and long-distance counterparts, as has been suggested for segmental phonology.

Chapter 7 demonstrates the effect that foot structure has on a series of segmental and morphological alternations from beyond the domain of stress. Patterns that appear properly subsequential in the absence of foot structure are rendered ISL instead when foot structure is present. Without foot structure, the patterns can only be calculated by an overt parity count of the syllables in the word. Foot structure, however, provides local input structure that can be used to determine the alternations with no reference to counting instead. It is shown that avoiding subsequential power has measurable, desirable implications for the predicted typology of such patterns. Thus, the work presented in Chapter 6 is an argument for a specific substantive proposal from the perspective of formal complexity.

Chapter 8 recapitulates the main findings of the thesis and outlines several open questions that work on the thesis has generated. One area for future research concerns how compositions of local functions can be restricted further to increase their typological accuracy while still allowing attested patterns to be described. Another promising avenue for future work is a formal comparison of different representations for stress – the metrical grid and metrical tree structures.

CHAPTER 2: STRESS

Stress is realized as acoustic prominence on one or more syllables in a word. The study of stress is as old as the study of phonology in generative grammar. Because of the wide variety of patterns in the typology of stress, it makes for a lucrative testing ground for formal theories of linguistics. For some previous work that engages large tracts of the stress typology, see: Hyman (1977); Booij (1983); Halle and Vergnaud (1987); Dresher and Kaye (1990); Bailey (1995); Hayes (1995); Gordon (2002); Heinz (2009). These are also the main sources from which examples of stress patterns were mined for the current work.

This dissertation examines patterns where stress is predictable given the length of the word and weight of the syllables in the word. Languages where stress is part of the lexical entry for the word i.e. lexical stress languages are not discussed. Languages with predictable stress make a good object of study because the input-output map can be described as a transduction that applies to an input string of any length and provides the correct output string. The formal properties of these mappings tell us about the nature of stress as it occurs in linguistic typology, and provide hypotheses as to what a possible and – importantly – *im*possible stress pattern is. Note that most, if not all such patterns contain exceptional forms. I set these forms aside and focus on the main, most general pattern in the language.

The typology of stress can be split broadly between languages that are *quantity insensitive* (QI) and *quantity sensitive* (QS). In QI patterns, determination of stress placement is not affected by the presence of heavy or light syllables. QS patterns are the opposite –

placement of stress is sensitive to syllable weight. Languages show a great deal of variation with regards to what constitutes a heavy syllable. Some treat both CVC and CV: syllables as heavy, while others behave as if only one is heavy. Some languages make even more fine-grained distinctions. For example, stress in Yana (Sapir and Swadesh 1960) equates CVC and CV: syllables as equally heavy, while Mauritanian Pulaar (Niang 1997) behaves as though CVC is heavier than CV, while CV: is heavier than both.

Despite the descriptive differences between QI and QS patterns, the computational analysis presented in this work reveals a surprising amount of overlap in the formal properties of the stress map. Despite a small difference in the alphabet of symbols the functions use, the computation of the stress function is analogous in QI and QS patterns. While some past work has attempted to account for similarities in QI and QS typology (Prince 1983; Kager 1992), the analyses in this dissertation suggest a new classification for stress typology based on the computational properties of their atomic elements, rather than the descriptive surface properties that result from the computational characteristics that the stress atoms have. One area where QI and QS stress do differ substantially is in the area of long-distance patterns, which – it is shown in Chapter 4 – can only be described with a QS pattern.

2.1 Representation

In this dissertation, stress is studied as a string to string mapping, where a series of functions apply to an input of bare syllables and return an output of syllables marked with stress in some way. For example, an "initial stress" function would provide the following mapping:

(4) $\sigma\sigma\sigma\sigma\sigma \mapsto \acute{\sigma}\sigma\sigma\sigma\sigma$

The analyses presented here make no reference to foot structure or the metrical grid (Liberman and Prince 1977; Halle and Vergnaud 1978; Selkirk 1980; Prince 1983; Hammond 1984; Halle and Vergnaud 1987; Idsardi 1992; Hayes 1995). Not committing to a specific instantiation of grids or tree structures in representation of stress allows for the right level of abstraction to study the fundamental computational properties that stress patterns have. It also avoids results that are overly tied to any one particular theory of the representation of stress. Even within a single mode of representation, there is lively theoretical debate on the fine details of the structure. For example, alongside proposals that feet are strictly binary (Hammond 1990; Kager 1993; Hayes 1995) are other proposals such as ternary feet (Halle and Vergnaud 1987; Dresher and Lahiri 1991; Rice 1992) and overlapping feet (Hyde 2002). Thus, by not committing to a particular theory or subtheory of representation, the current work avoids staking its claims on the accuracy or popularity of that particular theory. This choice is not a denial of hierarchical structure for stress – in fact, the mappings described throughout the dissertation are congruent with the designation of heads in such serial analyses, where syllables are marked for stress and thus promoted to the next level on the tree or grid. As such, in addition to the conclusions drawn with regards to computation, the results presented here can be taken as a formal analysis of an integral step in classic serial accounts of stress. The abstract computational properties of stress patterns also hold no matter what representational theory ends up being "correct", and future work on representation of stress can refer to the results presented here as a guide to the computational complexity of their own models.

It should also be acknowledged that the conception of stress assignment in this work is arguably more akin to a grid structure, as it lacks horizontal constituency groups such as feet or prosodic words. So, while this work attempts to abstract away from specific proposals with regards to representation of stress as much as possible, it is not the case that *no* theoretical commitments are made, and the results presented here are indeed tied to those theoretical commitments. For example, the analyses presented here adopt the syllable as a basic unit that is created before stress applies which, while reasonably non-controversial, is still a theoretical commitment. Nevertheless, I argue that remaining agnostic to the extent that it is possible with regards to specific models of representation enables the appropriate level of abstraction for the study of the computational properties of iterative stress pursued in this paper. This is also useful because it allows for an evaluation of the complexity that a particular model *itself* adds to stress generalizations.

Additionally, this thesis does not address cases of transformations that operate on lexical stresses, such as stress shift rules in English (see, for example, Kenstowicz (1994, p.237)). The computational properties of such rules is an important research area to provide a more complete conception of the computation of stress phenomena, though it falls beyond the purview of the current work. A brief discussion can be found in Chapter 7. I now give an overview of the typology of stress patterns.

2.2 Stress Typology

2.2.1 Bounded non-iterative

The simplest types of stress patterns, both descriptively and computationally, are "single stress" or "bounded non-iterative" patterns. These patterns place a single stress at a fixed distance from a word edge. The term "bounded" has been used inconsistently in the literature in reference to various, sometimes diametrically opposed, stress-related phenomena. For instance, single stress patterns traditionally featured one foot that parsed the entire word, and were thus "unbounded" – while iterative patterns required multiple binary "bounded" feet. Throughout this dissertation I use the terms in a way that is naive to foot structure and more related to computation, referring instead to the placement of stress itself. If a stress pattern places stress locally, it is bounded. If it is not, and placement of stress can occur in principle anywhere in the word, it is unbounded.

QI

Single stress patterns account for about 75% of the total number of languages in (Gordon 2002)'s typology of QI stress. An example of a QI single stress pattern is that of Afrikaans (Donaldson 1993), which places one stress on the initial syllable:

(5) $\acute{\sigma}, \acute{\sigma}\sigma, \acute{\sigma}\sigma\sigma, \acute{\sigma}\sigma\sigma\sigma, \acute{\sigma}\sigma\sigma\sigma\sigma, \acute{\sigma}\sigma\sigma\sigma\sigma\sigma...$

A function that applies single stress depends only on a short count of syllables from a word boundary symbol, and so can be described solely with reference to the input. These patterns are attested for every position up to three syllables away from the left or right word edge. Examples are shown in the following table:

initial	$\acute{\sigma}, \acute{\sigma}\sigma, \acute{\sigma}\sigma\sigma, \acute{\sigma}\sigma\sigma\sigma,$ $\acute{\sigma}\sigma\sigma\sigma\sigma, \acute{\sigma}\sigma\sigma\sigma\sigma\sigma...$ Afrikaans (Donaldson 1993)	$\acute{\sigma}, \sigma\acute{\sigma}, \sigma\sigma\acute{\sigma}, \sigma\sigma\sigma\acute{\sigma},$ $\sigma\sigma\sigma\sigma\acute{\sigma}, \sigma\sigma\sigma\sigma\sigma\acute{\sigma}...$ Abun (Berry and Berry 1999)	final
peninitial	$\acute{\sigma}, \sigma\acute{\sigma}, \sigma\acute{\sigma}\sigma, \sigma\acute{\sigma}\sigma\sigma,$ $\sigma\acute{\sigma}\sigma\sigma\sigma, \sigma\acute{\sigma}\sigma\sigma\sigma\sigma...$ Lakota (Gordon 2002)	$\acute{\sigma}, \acute{\sigma}\sigma, \sigma\acute{\sigma}\sigma, \sigma\sigma\acute{\sigma}\sigma,$ $\sigma\sigma\sigma\acute{\sigma}\sigma, \sigma\sigma\sigma\sigma\acute{\sigma}\sigma...$ Mohawk (Gordon 2002)	penultimate
postpeninitial	$\acute{\sigma}, \acute{\sigma}\sigma, \sigma\acute{\sigma}\sigma, \sigma\sigma\acute{\sigma}\sigma,$ $\sigma\sigma\acute{\sigma}\sigma\sigma, \sigma\sigma\acute{\sigma}\sigma\sigma\sigma...$ Azkoitia Basque (Hualde 1998)	$\acute{\sigma}, \acute{\sigma}\sigma, \acute{\sigma}\sigma\sigma, \sigma\acute{\sigma}\sigma\sigma$ $\sigma\sigma\acute{\sigma}\sigma\sigma, \sigma\sigma\sigma\acute{\sigma}\sigma\sigma...$ Macedonian (Lunt 1952)	antepenultimate

Table 2.1: QI single stress patterns

Note that not all single stress patterns are equally distributed in the observed typology. Among single stress patterns, Hyman (1977) and Gordon (2002) report 37% and 31% representation for initial stress, respectively, but only 2% and 4% for antepenultimate patterns, while Azkoitia Basque appears to be the only known case of postpeninitial stress. It is possible that learnability plays a role in this disparity, as forms that differentiate initial and antepenultimate stress, for example, are necessarily at least four syllables in length, and so plausibly less common than shorter forms. The case of postpeninitial stress in Basque seems to reinforce this idea, as stress in the language is paired with a non-finality requirement, and so the pattern is indistinguishable from penultimate stress until words of at least five syllables.

QS

QS cases of single stress patterns that are local are bounded to an edge, just like their QI counterparts. They differ only in that syllable weight sometimes affects the placement of stress. One example is Maidu (Shipley 1964), which stresses the initial syllable if heavy and otherwise stresses the peninitial syllable. In this sense the peninitial is the default location for stress, and a heavy initial pulls stress away from the default position:

(6)
Ĥ́HH
H́LL
LĹL
LH́L
LĹH

All QS single stress patterns feature some form of this weight-related variation in the placement of stress. In Chapter 4 it is shown that, despite this minor variation, the complexity of the function computing QS or QI single stress patterns is the same, as it relies only on the syllables a small, fixed distance from an input word boundary. In Chapter 5, it is demonstrated that Maidu-like patterns cannot be defined as a QI pattern because of the lack of a heavy-light distinction. In other words, the limited input alphabet QI patterns constrains the range of patterns that can be described.

Almost all combinations of edge-orientation and weight-related variation with a two-syllable window are attested. Examples of two-syllable window patterns are given here, where the description "heavy 1, else 2" can be interpreted as "heavy initial, else peninitial" and "heavy -2, else -1" can be read as "heavy penult, else final":

heavy 1, else 2	LĹL, H́LL, H́HH, LH́L Maidu (Shipley 1964)	LĹL, LLH́, HHH́, LH́L Kawaiisu (Zigmond et al. 1990)	heavy -1, else -2
heavy 2, else 1	ĹLL, H́LL, HH́H, LH́L Panamint (Dayley 1989)	LLĹ, LLH́, HH́H, LH́L Javanese (Herrfurth 1964)	heavy -2, else -1
heavy 2, else heavy 1, else 2	LĹL, H́LL, HH́H, LH́L *unattested*	LĹL, LLH́, HH́H, LH́L Awadhi (Saksena 1971)	heavy -2, else heavy -1, else -2
heavy 1, else heavy 2, else 1	ĹLL, H́LL, H́HH, LH́L Malayalam (Mohanan 1986)	LLĹ, LLH́, HHH́, LH́L Yapese (Hayes 1981)	heavy -1, else heavy -2, else -1

Table 2.2: QS single stress patterns

The only missing pattern is the left-edge mirror of Awadhi (Saksena 1971). Such a language would stress a heavy peninitial, else a heavy initial, else the peninitial. Among logically possible patterns involving a three-syllable window, only right-edge oriented patterns appear in the sources surveyed, and in very small numbers. Latin displays a "heavy penult else antepenult pattern" that may be shared with Malay (Lewis (1947) but cf. Winstedt (1927)). Maithili (Subdara 1944) stresses a heavy penult, else a heavy final, else a heavy antepenult, with default stress on the penult. Pirahã (Everett 1988) features a multi-tiered weight system that stresses the final, penult, or antepenult depending on which syllable is heaviest, with priority given in that order in case weight is equal. This exhausts the inventory of bounded trisyllabic window systems in the sources surveyed.

2.2.2 Bounded iterative

Another relevant type of stress pattern are the bounded iterative patterns. These languages place stress iteratively away from a word edge or other morphological marker in one or both directions. Though stress occurs in more than one place in the word, these patterns are still "bounded" in the sense that they rely on *output* locality – further stresses are placed based on the location of previous stress in the output string. While this is distinct from the input-oriented non-iterative patterns, it is still formally local, and thus bounded.

QI

Bounded iterative languages account for around 15% of all languages in Gordon (2002)'s typology of QI stress. An example is Murinbata (Street and Mollinjin 1981):

(7) $\acute{\sigma}\sigma, \acute{\sigma}\sigma\grave{\sigma}, \acute{\sigma}\sigma\grave{\sigma}\sigma, \acute{\sigma}\sigma\grave{\sigma}\sigma\grave{\sigma}, \acute{\sigma}\sigma\grave{\sigma}\sigma\grave{\sigma}\sigma, \ldots$

Main stress is located on the initial syllable, and secondary stresses are applied to alternating syllables thereafter. Patterns where iteration occurs without exceptions are attested for all combinations of directionality and locus of iteration:

L-R initial	L-R peninitial	R-L final	R-L penultimate
$\acute{\sigma}\sigma, \acute{\sigma}\sigma\grave{\sigma}, \acute{\sigma}\sigma\grave{\sigma}\sigma,$ $\acute{\sigma}\sigma\grave{\sigma}\sigma\grave{\sigma}, \acute{\sigma}\sigma\grave{\sigma}\sigma\grave{\sigma}\sigma, \ldots$ Murinbata (Street and Mollinjin 1981)	$\sigma\acute{\sigma}, \sigma\acute{\sigma}\sigma, \sigma\acute{\sigma}\sigma\grave{\sigma},$ $\sigma\acute{\sigma}\sigma\grave{\sigma}\sigma, \sigma\acute{\sigma}\sigma\grave{\sigma}\sigma\grave{\sigma}, \ldots$ Araucanian (Echeverria and Contreras 1965)	$\sigma\acute{\sigma}, \grave{\sigma}\sigma\acute{\sigma}, \sigma\grave{\sigma}\sigma\acute{\sigma},$ $\grave{\sigma}\sigma\grave{\sigma}\sigma\acute{\sigma}, \sigma\grave{\sigma}\sigma\grave{\sigma}\sigma\acute{\sigma}, \ldots$ Weri (Boxwell and Boxwell 1966)	$\acute{\sigma}\sigma, \sigma\acute{\sigma}\sigma, \grave{\sigma}\sigma\acute{\sigma}\sigma,$ $\sigma\grave{\sigma}\sigma\acute{\sigma}\sigma, \grave{\sigma}\sigma\grave{\sigma}\sigma\acute{\sigma}\sigma, \ldots$ Warao (Osborn 1966)

Table 2.3: QI iterative patterns

As iteration is the only factor determining placement of stress in these languages, reference to the output to check the location of the previous stress is all that is needed to compute the stress map.

There are, however, cases where exceptions to basic iteration require additional machinery. The distinctions between these iterative cases are a main focus of Chapter 4. Languages that allow lapses are an example of this type of pattern. In Pintupi (Hansen and Hansen 1969), stress iterates from left to right, but adheres to a strict non-finality requirement:

(8) $\acute{\sigma}\sigma, \acute{\sigma}\sigma\sigma, \acute{\sigma}\sigma\grave{\sigma}\sigma, \acute{\sigma}\sigma\grave{\sigma}\sigma\sigma, \acute{\sigma}\sigma\grave{\sigma}\sigma\grave{\sigma}\sigma, \ldots$

In odd-parity forms, stress does not iterate to the final syllable as it would in Murinbata, creating a lapse at the edge of the word. There are also cases where the lapse occurs word-internally, such as Garawa (Furby 1974):

(9) $\acute{\sigma}\sigma, \acute{\sigma}\sigma\sigma, \acute{\sigma}\sigma\grave{\sigma}\sigma, \acute{\sigma}\sigma\sigma\grave{\sigma}\sigma, \acute{\sigma}\sigma\grave{\sigma}\sigma\grave{\sigma}\sigma, \acute{\sigma}\sigma\sigma\grave{\sigma}\sigma\grave{\sigma}\sigma\ldots$

In odd-parity forms, leftward iteration of stress would create a clash with the main, initial stress, but this is prevented, leading to a word-internal lapse. There are some gaps in the typology of lapse patterns:

left edge	left internal	right edge	right internal
$\sigma\acute{\sigma}, \sigma\sigma\acute{\sigma}, \sigma\grave{\sigma}\sigma\acute{\sigma},$ $\sigma\sigma\grave{\sigma}\sigma\acute{\sigma}, \sigma\grave{\sigma}\sigma\grave{\sigma}\sigma\acute{\sigma}, \ldots$ *unattested*	$\acute{\sigma}\sigma, \acute{\sigma}\sigma\sigma, \acute{\sigma}\sigma\grave{\sigma}\sigma,$ $\acute{\sigma}\sigma\sigma\grave{\sigma}\sigma, \acute{\sigma}\sigma\grave{\sigma}\sigma\grave{\sigma}\sigma, \ldots$ Garawa (Furby 1974)	$\acute{\sigma}\sigma, \acute{\sigma}\sigma\sigma, \acute{\sigma}\sigma\grave{\sigma}\sigma,$ $\acute{\sigma}\sigma\grave{\sigma}\sigma\sigma, \acute{\sigma}\sigma\grave{\sigma}\sigma\grave{\sigma}\sigma, \ldots$ Pintupi (Hansen and Hansen 1969)	$\acute{\sigma}\sigma, \sigma\acute{\sigma}\sigma, \grave{\sigma}\sigma\acute{\sigma}\sigma,$ $\grave{\sigma}\sigma\sigma\acute{\sigma}\sigma, \grave{\sigma}\sigma\grave{\sigma}\sigma\acute{\sigma}\sigma, \ldots$ Piro (Matteson 1965)

Table 2.4: QI iterative lapse patterns

There is no known iterative pattern that places a lapse at the left edge of the word. This cannot be construed as a general ban on initial lapses, as both systems with fixed right-edge stress and unbounded systems where stress may fall anywhere in the word will feature forms with initial lapses.

Another group of exceptions to basic iteration of stress are the clash languages, which allow adjacent stresses. Just like their lapse counterparts, output-centered iteration does not account for the patterns alone. An example is Ojibwe (Kaye 1973):

(10) $\sigma\acute{\sigma}, \sigma\acute{\sigma}\grave{\sigma}, \sigma\acute{\sigma}\sigma\grave{\sigma}, \sigma\acute{\sigma}\sigma\grave{\sigma}\grave{\sigma}, \sigma\acute{\sigma}\sigma\grave{\sigma}\sigma\grave{\sigma}, ...$

The final syllable always bears stress, even when this creates a stress clash in odd-parity forms. Most attested cases fix the clash stress at the edge, though Southern Paiute (Harms 1966) is an example of a language that could be described as “internal clash”:

(11) $\acute{\sigma}\sigma, \sigma\acute{\sigma}\sigma, \sigma\acute{\sigma}\grave{\sigma}\sigma, \sigma\acute{\sigma}\sigma\grave{\sigma}\sigma, \sigma\acute{\sigma}\sigma\grave{\sigma}\grave{\sigma}\sigma, \sigma\acute{\sigma}\sigma\grave{\sigma}\sigma\grave{\sigma}\sigma...$

Here the fixed stress is penultimate, and other stresses iterate from the main peninitial position, even when this creates a clash between the antepenult and penultimate syllables. While clashes at both edges are attested, Paiute appears to be the only known example of internal clash.

left edge	left internal	right edge	right internal
$\acute{\sigma}\sigma, \grave{\sigma}\acute{\sigma}\sigma, \grave{\sigma}\sigma\acute{\sigma}\sigma,$ $\grave{\sigma}\grave{\sigma}\sigma\acute{\sigma}\sigma, \grave{\sigma}\sigma\grave{\sigma}\sigma\acute{\sigma}\sigma, ...$ Biangai (Dubert and Dubert 1973)	$\sigma\acute{\sigma}, \sigma\grave{\sigma}\acute{\sigma}, \sigma\grave{\sigma}\sigma\acute{\sigma},$ $\sigma\grave{\sigma}\grave{\sigma}\sigma\acute{\sigma}, \sigma\grave{\sigma}\sigma\grave{\sigma}\sigma\acute{\sigma}, ...$ *unattested*	$\sigma\acute{\sigma}, \sigma\acute{\sigma}\grave{\sigma}, \sigma\acute{\sigma}\sigma\grave{\sigma},$ $\sigma\acute{\sigma}\sigma\grave{\sigma}\grave{\sigma}, \sigma\acute{\sigma}\sigma\grave{\sigma}\sigma\grave{\sigma}, ...$ Ojibwe (Kaye 1973)	$\acute{\sigma}\sigma, \sigma\acute{\sigma}\sigma, \sigma\acute{\sigma}\grave{\sigma}\sigma$ $\sigma\acute{\sigma}\sigma\grave{\sigma}\sigma, \sigma\acute{\sigma}\sigma\grave{\sigma}\grave{\sigma}\sigma, ...$ Southern Paiute (Harms 1966)

Table 2.5: QI iterative clash patterns

Tauya (MacDonald 1990) is another case that resembles Biangai, but with iteration from the final syllable, rather than penultimate syllable. No other QI iterative clash patterns were found in the sources consulted. Worth noting is that in all attested patterns, the fixed clash stress is always located at the end of an iterative stress chain – rather than the beginning – and it is never the main stress.

While iteration of stress is typically binary, there are some cases of ternary stress. An example is Ioway-oto (Whitman 1947), which stresses every third syllable from the peninitial counting to the right:

(12) $\sigma\acute{\sigma}, \sigma\acute{\sigma}\sigma, \sigma\acute{\sigma}\sigma\sigma, \sigma\acute{\sigma}\sigma\sigma\grave{\sigma}, \sigma\acute{\sigma}\sigma\sigma\grave{\sigma}\sigma, \sigma\acute{\sigma}\sigma\sigma\grave{\sigma}\sigma\sigma, \sigma\acute{\sigma}\sigma\sigma\grave{\sigma}\sigma\sigma\grave{\sigma}...$

Here iteration requires consideration of a larger span of syllables i.e. a bigger window, but it is local to the output in the same way as patterns where iteration is binary.

A small number of languages display bidirectional stress assignment – iteration proceeds in both directions through the word[1]. This is distinct from the internal lapse patterns, which are sometimes referred to as bidirectional (Kager 2007). As stress in internal lapse patterns only proceeds in one direction, the term "bidirectional" may be inappropriate in these cases. Languages like Cahuilla (Seiler 1977; Levin 1988a) however, provide an unambiguous example of stress iterating in two directions. Main stress is placed to the right of the root boundary, and secondary stress iterates away in both directions:

(13)
$$\grave{\sigma}\sigma + \acute{\sigma}\sigma\grave{\sigma}$$
$$\sigma\grave{\sigma}\sigma + \acute{\sigma}\sigma\grave{\sigma}\sigma$$
$$\grave{\sigma}\sigma\grave{\sigma}\sigma + \acute{\sigma}\sigma\grave{\sigma}\sigma\grave{\sigma}$$
$$\sigma\grave{\sigma}\sigma\grave{\sigma}\sigma + \acute{\sigma}\sigma\grave{\sigma}\sigma\grave{\sigma}\sigma$$
$$\ldots$$

The count for stress in the prefix could not be carried out left to right, as it depends on the location of the root-initial syllable. Other examples of bidirectional patterns come from Auca (Pike 1964) and Iñapari (Parker 1999)[2]. From a formal perspective, bidirectional patterns are interesting because they require not one output local function to compute, but two – one for each direction of iteration. As such, they provide insight into the nature of compositions of contradirectional functions in general, as well as expanding our understanding of what is and is not possible in stress. Such patterns are addressed in Chapter 5.

[1]Thanks to Bill Idsardi for pointing these out.
[2]Though it should be noted that Iñapari's prosodic system is difficult to classify – it exhibits many features of a typical stress language, but also some features that are arguably more typical of a tone language.

QS

For QS patterns, there is considerable variation in terms of where main stress falls and how iterative stress behaves in relation to main stress and heavy syllables. The vast majority of patterns stress every heavy syllable, which results in a blurring of the neat boundary between simple iterative, lapse, and clash patterns that exists for QI patterns. Thus, instead of cataloguing every known case of QS iterative stress, I focus on several that provide important points of comparison with their QI counterparts:

QS main, QI iteration	L̀LL̀LH́L, L̀HH̀HH́L, LL̀HL̀LH́, LL̀HH̀HH́, L̀LL̀LĹL Romanian (Chitoran 1996)
QS main, QS iteration	L̀LL̀LH́L, LH̀H̀H̀H́L, L̀LH̀L̀LH́, L̀LH̀H̀H̀H́, L̀LL̀LĹL Fijian (Schütz 1985)
Possible internal clash	L̀LL̀LĹL, L̀LL̀H̀H́L, H̀LL̀LH̀H́, L̀LL̀H̀LH́ Romansh (Kamprath 1987)
QS non-finality lapse	ĹLL̀LH̀, ĹLH̀HH̀, ĹLL̀LL, ĹLH̀HL Wergaia (Hercus 1986)
Main from QS iteration	ĹLL̀LH̀L, LH́H̀H̀H̀L, ĹLH̀L̀LH̀, ĹLH̀H̀H̀H̀, ĹLL̀LL̀L Nywawaygi (Dixon 1983)
QS main with parity	LL̀LĹL, HL̀LH́L, LL̀LL̀H́L, LL̀LĹLL Cayuga (Foster 1982)
QS ternary iteration	LLL̀LLH́, LH̀LH́L, LH̀LHH́L, L̀LLL̀LLĹL Sentani (Cowan 1965)

Table 2.6: Some relevant QS iterative patterns

In some languages, placement of main stress is QS, but iteration of further stress is not. An example is Romanian (Chitoran 1996), which stresses a heavy final, else the penult, and then iterates stress leftward to alternating syllables regardless of weight. The majority of QS iterative patterns, however, include weight sensitive iteration in addition to weight sensitive main stress. Fijian (Schütz 1985) is an example. Main stress is the same as Romanian, but iterative stress falls on every heavy syllable in the string, in addition to alternating light syllables. This can create numerous clashes in a single word, depending on its form.

When main stress is anchored at the opposite edge from which iteration proceeds, this can result in an internal clash in certain words. An example comes from Romansh (Kamprath 1987), which has the same main stress pattern as Romanian and Fijian, but iterates stress with weight sensitivity from the *left* edge. This means that iteration can create an internal clash with the main stress, but only when the configuration of syllables allows it.

Lapse in QS languages appears to be limited to non-finality cases – all other patterns minimally stress alternating light syllables or every heavy syllable, eliminating the possibility of internal lapse. An example comes from Wergaia (Hercus 1986), which iterates stress left to right regardless of weight, but avoids stressing a final syllable only in the case that it is light. Final heavy syllables in the iterative chain surface with stress.

Patterns like those found in Fijian, Wergaia, and Romansh demonstrate that the fixed rules for clash and lapse that characterize the QI iterative patterns are absent in QS iterative typology. For example, while Ojibwe features a clash at the end of every odd-parity word, clashes in Fijian occur as a property of the syllables in each individual string i.e. whenever a sequence of two heavies happens to be present. In other words, for QS patterns, clash and lapse are *not* predictable given the length of the string, whereas this predictability is a hallmark of clash and lapse in QI patterns.

Other cases demonstrate that placement of main stress can depend on QS iteration of secondary stress. In Nywawaygi (Dixon 1983), stress applies iteratively right to left with weight sensitivity. Main stress then falls at the left edge, on whichever of the initial or peninitial syllable bears secondary stress, prioritizing the initial syllable if both are stressed. As the count for iteration starts from the right, no function starting from the right could place main stress properly – QS iteration from the opposite edge is crucial to determine the output.

Some patterns also place main stress at the end of an iterative chain, but combine weight sensitivity and parity-counting to determine where it should fall. An example comes from Cayuga (Foster 1982). Secondary stress in Cayuga iterates left to right to every other syllable regardless of weight, modulo non-finality. Main stress falls on the penult whenever it is even or if it is odd and heavy. When the penult is odd and light, stress is antepenultimate instead. This can result in an internal clash when iteration of secondary stress bumps up against an odd heavy penult that bears main stress.

As with QI patterns, ternary QS patterns are uncommon, but attested. One example comes from Sentani (Cowan 1965), which stresses a heavy final syllable, else the penult. From here, stresses iterate leftwards onto every third syllable. Ternary iteration is disrupted by a heavy syllable that occurs two syllables away from a stress. This can be interpreted as a ban on clash in the language.

Despite the wide variation among QS patterns and what appear to be significant surface differences between QS and QI languages, the analysis in Chapter 4 demonstrates that all iterative patterns share fundamental computational similarities, whether they are QI or QS. Both are subject to the same analysis, and the difference between the two amounts to a difference in the labels on syllables, not the complexity of the input-output maps of iterative patterns.

2.2.3 Unbounded patterns

The patterns discussed thus far all place stress in a fixed location, whether that location is local to the input or output string. However, there are patterns that are non-local – they are long-distance or "unbounded" in the sense that stress may apply to any syllable in the word if it meets certain conditions. The formal study of these long-distance patterns allows for a restrictive hypothesis of long-distance typology based on their computation.

Interestingly, examples of unbounded patterns appear to be limited to QS languages. Chapter 5 demonstrates why this is the case – a long distance generalization requires an input alphabet of at least two symbols. QI patterns, with an input alphabet of just σ^n, have no way to define such patterns.

DTO and DTS

The first relevant group of patterns are the default to opposite (DTO) and default to same (DTS) patterns (terminology due to Prince (1985)). These patterns stress a left (or rightmost) heavy syllable if any are present, or a light syllable at the edge of the word. There is only one stress per word. For example, the “leftmost heavy or rightmost” (LHOR) pattern of Kwakw’ala (Boas and Deloria 1941) stresses the leftmost heavy syllable in the word if one is present, or the rightmost syllable in forms with no heavies. All combinations of directionality are attested:

LHOR	LLLLĹ, LLH́LL, LH́LHL, H́HHHH Kwakw’ala (Boas and Deloria 1941)
LHOL	ĹLLLL, LLH́LL, LH́LHL, H́HHHH Lushootseed (Hess 1976)
RHOL	ĹLLLL, LLH́LL, LHLH́L, HHHHH́ Huasteco (Larsen and Pike 1949)
RHOR	LLLLĹ, LLH́LL, LHLH́L, HHHHH́ Western Cheremis (Itkonen 1955)

Table 2.7: DTO and DTS patterns

In a counter-intuitive result, Hao and Anderson (2019) and Koser and Jardine (2020b) demonstrate that DTS patterns are provably more complex that DTO patterns, despite the descriptive nature of the patterns, where one might assume that a “same-edge” orientation implies a simpler stress map. This is not the case, however – while DTO languages search the string for a stressable target in one direction, DTS languages must search in *both* directions. In LHOR, for example, a heavy is stressed if no heavies precede it, and the same holds true for a final light syllable. But in LHOL, a heavy is also stressed if no other heavies precede it and an initial light syllable is stressed if no heavy syllables *follow* it. Chapter 5 presents a restrictive theory of such patterns based on the computational properties of the individual stress atoms that comprise them.

Suprabinary patterns

DTO and DTS patterns employ a binary scale for weight – heavy and light – and the bidirectionality of DTS makes it more complex than DTO. There also exist patterns that are sensitive to a suprabinary scale. An example comes from Mauritanian Pulaar (Niang 1997), which employs the following quaternary weight scale: V:C > V: > VC > V. Main stress is assigned to the leftmost heaviest syllable in the word:

(14)	a. á.du.na	'world'	e. bá:.wa:.do	'weak person'
	b. bá.la.be	'shoulders'	f. jol.ti.nó:.wo	'person who removes out of'
	c. da.dór.de	'waist'	g. ha:l.pu.lá:r.ʔen	'speakers of Pulaar'
	d. tál.lor.de	'place for rolling over'	h. ka:.sa.má:s.na:.jo	'person from Casamance'

As with DTO and DTS, the location of stress varies significantly depending on the composition of the word. Another example comes from Nanti (Crowhurst and Michael 2005), which is sensitive to four weight levels CV:N > CV: > CVN > CV and a three-level sonority distinction in vowels, low > mid > high. This results in a twelve-step scale for weight, with a Ca:N syllable being the best target, and a Ci syllable being the worst. In general, Nanti places main stress (subject to non-finality) on the most stressable target in a word or, if two or more are tied, the rightmost one:

(15)	a. o.ko.ri.kʃi.tá.ka	'she wore nose discs'	e. i.tin.ka.ráa.ʃii.gʒi	'they picked thatch'
	b. já.nɯi.ti	'he walked'	f. nóo.ka.na.kse.ro	'I discarded it'
	c. non.kan.táa.ga.kse	'we will have said'	g. pi.ká.bi.ri.ti	'you get going
	d. noo.gái.ga.ro	'we ate it'	h. non.ka.mán.te	'I will tell'

In Chapter 5 it is demonstrated that these patterns can also be described as the composition of individual stress atoms in a way that constrains the predicted typology of long-distance stress, though they require slightly different machinery than DTO or DTS patterns. Despite the increased number of weight levels between Pulaar and Nanti, the computation of the two patterns is the same. This indicates that, though crossing into suprabinarity in the scale of an unbounded pattern has measurable implications for the computation of a stress pattern, further increases beyond the suprabinary level do not. In the next chapter, I turn to the necessary formal background for the framing of the dissertation.

Chapter 3: Formal Background

3.1 Introduction

This chapter gives the necessary formal background for the proceeding analyses in this dissertation. This is done by demonstrating how the complexity classes of formal language theory (FLT) can be connected to patterns found in natural language phonology. This helps provide an explanation as to *why* some patterns are attested and some are not, and so the conventions of FLT as it relates to natural language are discussed in detail. As the patterns are studied here as *functions* that map an input string to an output string, finite state transducers (FSTs) are used to provide insight into the formal properties of stress.

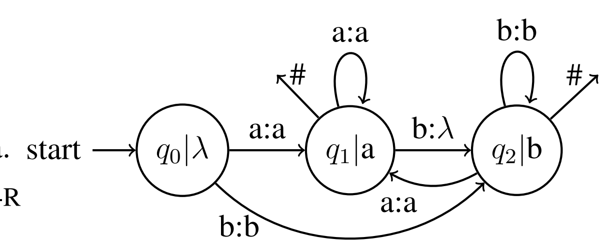
The chapter also provides an overview of relevant existing complexity classes. These include the input and output strictly local classes (Chandlee 2014; Chandlee et al. 2015; Chandlee and Heinz 2018) and subsequential classes (Schutzenberger 1977). Local functions do not suffice for long-distance stress patterns, and so the strictly piecewise functions (Burness and McMullin 2020), which implement long-distance generalizations, are introduced as well.

3.2 Finite state representations of stress

In this dissertation I study stress as a mapping from an input string of syllables that are unmarked for stress to an output string of syllables that are marked for stress. To study the formal properties of these mappings, I represent them using finite state transducers (FST;

see Sakarovitch (2006) for an introduction). An FST is a kind of directed graph where a set of states are connected by transitions between those states. FSTs provide a medium to study the abstract computational properties a function has by making it clear what kind of information the function is sensitive to. Here, the relevant information encoded by a certain state is given as a label appearing on the state. A transducer reads a string by starting with the first input symbol and taking the transition matching that input symbol to the state that is reached by that transition. The corresponding output symbol for that input symbol and transition is written to the output string. Subsequent input symbols are read one at a time until the entire string has been processed. The transduction may apply in either direction – from left to right, beginning with the first symbol in the string, or right to left, beginning with the last symbol in the string. In this way, transducers can model phonological processes that necessarily apply in a specific direction, such as iteration of stress. Here the direction in which the transduction applies is noted with a 'L-R' or 'R-L' under the transducer. This is for clarity for the reader, and is not a part of the formal definition of an FST. The transitions are labeled with input-output pairs, where the input symbol is given to the left of a colon and the output symbol appears to the right. When moving through a machine, the current state of the transduction encodes information that is relevant to the function it represents. Consider the following example transducer, which represents a function that deletes an input b that follows an input a:

(16)

a. 

L-R

The first state, q_0, indicates the start of the transduction. When the transduction is in state q_1, this indicates that the previous symbol seen was an a – all transitions leading to the state have an input a. Thus, we say that a is the minimal *suffix* leading to the state. To make the relevant suffix leading to a state explicitly clear, the transducers in this paper feature state labels, such as 'q_1|a' on the state q_1 in (16). State q_2 is the 'b state' – all transitions leading to q_2 have an input b, and so it is labeled b. When in q_1, if the next input symbol is a b, the transition to q_2 is taken and the empty string λ is written to the output. This type of transition with λ can be used to model deletion, or when the function needs to wait for more information before deciding what to output for a given input. The transitions leaving q_1 and q_2 labeled with # indicate reading of the word boundary and the end of the word. Throughout the paper, I exclude these where they are irrelevant.

The function represented by (16) deletes an input b that appears immediately after an input a, providing mappings such as in the following examples:

(17) a. $aaabbb \mapsto aaabb$

b. $bbbbb \mapsto bbbbb$

c. $ababab \mapsto aaa$

d. $bbbaaa \mapsto bbbaaa$

Example (18) demonstrates the application of a transduction to a string using (17). Diagrams such as the ones shown in (18) are helpful in interpreting transducers, and so feature throughout the dissertation. The input and output lines represent the respective input and output strings. The line of states indicates what transition was taken in the machine. For instance, the sequence "$q_0 \rightarrow q_1$" can be interpreted as starting in state zero and taking a transition to state one, with the corresponding pair of input and output symbols. The transducer in (16) reads the string left to right, and so the function starts at the beginning of the word.

(18)

input:		a		a		a		b		b		b	
states:	q_0	$\rightarrow$	q_1	$\rightarrow$	q_1	$\rightarrow$	q_1	$\rightarrow$	q_2	$\rightarrow$	q_2	$\rightarrow$	q_2
output:		a		a		a		λ		b		b	

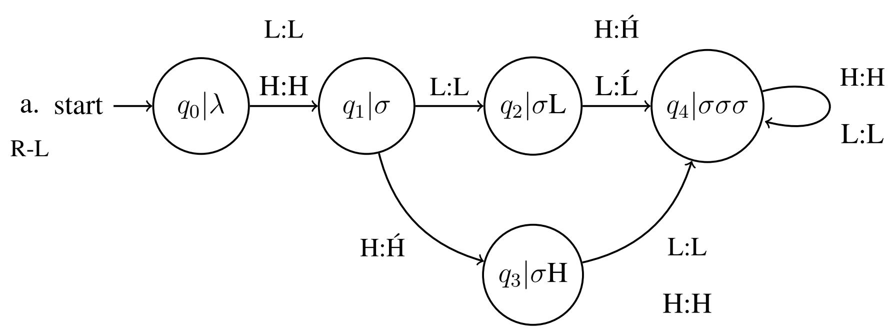
Reading the string left to right, the initial three input symbols are a, and so the transduction loops in q_1. The fourth symbol is a b that follows an a, and so the transduction moves to q_2 after outputting λ. The remaining input symbols are all b, and so the transduction remains in q_2. The string $aaabbb$ is thus mapped to $aaabb$. Note that FSTs such as this one describe not just a single input-output pair, but will map any input to the correct output for a particular pattern.

Studying phonological patterns in this way reveals the abstract computational properties that phonology has. It indicates which patterns are more, less, or equally complex in a rigorous, mathematically defined way. FSTs are thus an invaluable tool in the pursuit of the most restrictive characterization of linguistic phenomena such as stress. As an example of a stress map, consider the following examples from Latin, which stresses the antepenult unless the penult is heavy, in which case the penult is stressed intstead:

(19)	LLL	$\mapsto$	ĹLL
	HLL	$\mapsto$	H́LL
	HHH	$\mapsto$	HH́H
	LHL	$\mapsto$	LH́L

An FST for Latin would be as shown here in (20):

(20)



The transducer reads the input string right to left. State q_0 represents the beginning of the transduction, indicating that the only preceding material is the empty string λ. The final syllable is never stressed, and so it is output faithfully whether it is light or heavy in the transition from q_0 to q_1. That syllable weight is irrelevant at this point is reflected in the state labels, where the q_1 represents an input heavy *or* light syllable. From there, if a light syllable is seen, it is output as unstressed, as stress avoids penult light syllables. Intuitively, this is the information that is encoded in q_2, as indicated in the label (remembering that the string is read right to left) – a light penult has just been read in the input. The next syllable

– the antepenult – will then be marked as stressed no matter its weight. Alternatively, going back to q_1, if a heavy penult is encountered instead, it is stressed in the transition from q_1 to q_3. Once q_4 is reached, all other input symbols are left unchanged, as the end of the three syllable window has been reached. Interestingly, the state information makes explicit that some QS languages are sensitive to weight in certain positions, rather than the entire word – the penult is the only syllable where weight matters in Latin, for example.

To demonstrate the application of a stress function, consider the following derivations for input LLL and LHL. This transducer reads the string right to left, and so the function starts at the end of the word. The function correctly maps Latin inputs to Latin outputs:

(21)

input:		L		L		L		input:		L		H		L	
states:	q_4	$\leftarrow$	q_2	$\leftarrow$	q_1	$\leftarrow$	q_0	states:	q_4	$\leftarrow$	q_3	$\leftarrow$	q_1	$\leftarrow$	q_0
output:		Ĺ		L		L		output:		L		H́		L	

It should be noted that finite state analyses of linguistic patterns are not an assertion that the phonological grammar as it is instantiated neurologically is a series of finite state machines. Rather, FSTs are merely a useful analytical tool for representing phonological functions. This is because the properties of the transducers tell us about the properties of the functions themselves with regards to computational complexity. For the purposes of this paper, the most important property is what kind of information determines movement through the states of the transducer. In (20), for example, the states encode *input* information – though some states correspond to the same output symbols, each state encodes unique information about a local portion of the input string. FSTs thus help us answer relevant questions about the complexity and computation of phonological functions – is it local to the input or output, or does it use non-local information? Does it apply in any part of the word or is it tethered to an edge? These properties, encoded by formally distinct classes of transducers, play important roles in the theory of stress presented here.

3.3 FLT and Phonology

Formal language theory provides a well-defined measure of complexity in the form of complexity classes. The nested hierarchy of complexity classes divides the space of possible functions based on the expressive power of those functions. As applied to natural language, the study of FLT complexity delineates function classes that are relevant to natural language processes, helping to establish testable hypotheses about what a possible linguistic generalization is. One important result from this area of research is that phonological processes do not exceed the power of the *regular* class of functions (Johnson 1972; Kaplan and Kay 1994). Intuitively, this is because phonological functions are computed using a finite amount of memory and a finite alphabet. Further research has shown that the vast majority of phonological processes are *subregular* – belonging to some more restrictive subclass of the regular functions (Rogers et al. 2013; Heinz 2018). The study of the relation of phonological processes to subregular complexity classes is an ongoing program that delivers precise, mathematically-defined characterizations of phonology. I provide descriptions of relevant existing complexity classes here.

3.3.1 Subsequential

One relevant class of functions is the subsequential class (Schutzenberger 1977; Mohri 1997). Subsequential functions have well-understood automata-theoretic and learnability properties, and represent an important division in the space of possible functions that includes many phonological patterns and excludes many non-phonological ones (Oncina et al. 1993; Heinz and Lai 2013; Chandlee 2014; Jardine 2016; Payne 2017; Luo 2017). Though there are notable exceptions (Jardine 2016; McCollum et al. 2020; Hao and Anderson 2019; Koser and Jardine 2020b), most phonological processes are subsequential. While they are subregular, subsequential functions are more powerful than EO, ISL, OSL, or SP functions.

Subsequential transducers are *deterministic* – a property that separates them from more complex classes. In a deterministic transducer, for each input symbol, there is at most one transition leaving a given state for that input symbol. For example, q_0 in (16) has one exit transition for both inputs a and b. A hypothetical transition to q_1 using input a would entail non-determinism, as it would give the state multiple transitions for a single input. Allowing non-determinism has measurable consequences for the complexity of the functions that can be defined. Specifically, non-deterministic transducers correspond to the class of properly *regular* functions (Johnson 1972; Kaplan and Kay 1994). If phonology is actually *sub*regular, then an adherence to determinism based on the observed properties of phonological patterns will provide a better characterization of phonological typology.

An example of a subsequential stress pattern comes from the DTO languages, discussed above in §1.4.1. An example of a "leftmost heavy or right" (LHOR) pattern is found in Kwakw'ala (Bach 1975; Hayes 1995):

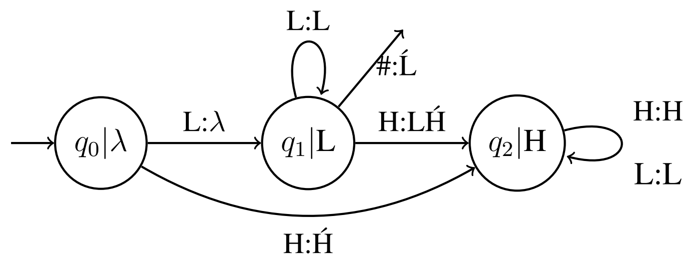
(22) LLLL $\mapsto$ LLLĹ
HHHH $\mapsto$ H́HHH
LHLLLL $\mapsto$ LH́LLLL
LHLLHL $\mapsto$ LH́LLHL

Reading the string left to right, the first heavy encountered is stressed. If no heavies are encountered, the final light syllable is stressed instead. This type of pattern loses the property of locality that is characteristic of EO, ISL, or OSL classes. This is because, to determine if any given H or a final L should be stressed, the function must keep track of the lack or presence of heavy syllables for the entire length of the word up to that point, as shown in the following diagram:

(23) $\overleftarrow{\mathrm{LL}}\boxed{\acute{\mathrm{H}}}\mathrm{LHL}$ $\quad$ $\overleftarrow{\mathrm{LLLLLL}}\boxed{\acute{\mathrm{L}}}$

A transducer is as follows:

(24) a. start L-R



The transduction proceeds left to right. The transition from q_0 to q_1 takes an input L and "waits" with λ – no output can be written until more input symbols are seen. If the word is light syllables only, the transduction loops in q_1, outputting Ĺ when the word ends, as seen in the exit transition on input #. If at any point a heavy is seen, it is output with stress and the transduction moves to q_2, where no further changes are made. In intuitive terms, being in q_1 means that only light syllables have been read, whereas being in q_2 means at least one heavy has been read. Note that this is not local information – LHOR is a long-distance pattern. That the transducer encodes this information about an arbitrary previous number of symbols it has seen makes the function properly subsequential. The following derivations for LLLL and LLHL demonstrate the LHOR mapping:

(25)

input:		L		L		L		L	
states:	q_0	$\rightarrow$	q_1	$\rightarrow$	q_1	$\rightarrow$	q_1	$\rightarrow$	q_1
output:		λ		L		L		L	Ĺ

input:		L		L		H		L	
states:	q_0	$\rightarrow$	q_1	$\rightarrow$	q_1	$\rightarrow$	q_2	$\rightarrow$	q_2
output:		λ		L		LH́		L	

When evaluating whether any position in the string, light or heavy, should surface with stress, the function depends on information about the – in principle – unbounded sequence of symbols preceding the current one. The loop in q_1 allows for an arbitrarily large sequence of L inputs before stress is ultimately applied. q_1 corresponds to only having read input L for the duration of the transduction, while q_2 corresponds to having seen at least one H somewhere in the string. As such, the pattern cannot be determined locally, and so is not OSL, ISL, or EO.

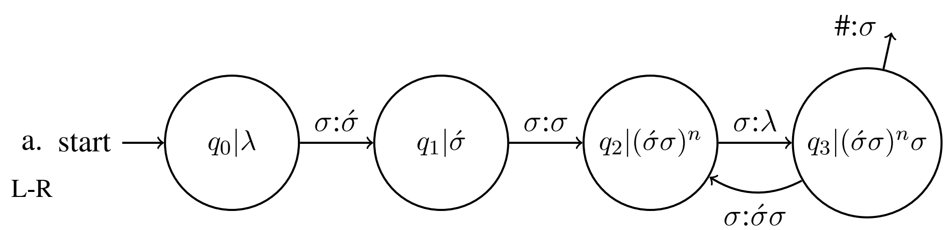
One property of properly subsequential functions that separates them from OSL functions in terms of their expressive power is lookahead. Subsequential functions may contain lookahead – the "waiting" as in (114) for some coming input before making a decision on what to output. OSL functions, however, must write a symbol to the output at any point that there is a transition from one state to another. They cannot provide lookahead.

As a consequence of this, some patterns that appear local in an intuitive sense require a properly subsequential function to describe. One example is the stress pattern of Pintupi (Hansen and Hansen 1969):

(26) $\acute{\sigma}\sigma, \acute{\sigma}\sigma\sigma, \acute{\sigma}\sigma\acute{\sigma}\sigma, \acute{\sigma}\sigma\acute{\sigma}\sigma\sigma, \acute{\sigma}\sigma\acute{\sigma}\sigma\acute{\sigma}\sigma, \ldots$

Stress iterates left to right starting from the initial syllable and avoids the final syllable. The pattern combines iteration, demonstrated to be OSL below, with the addition of a non-finality requirement. Though this may seem inconsequential in terms of the computation of the pattern, this non-finality requirement necessitates a properly subsequential function because it requires lookahead. In the trisyllable, for example, the third syllable is unstressed. However, in the four syllable form, the third syllable *does* bear stress. This means that a transducer for the pattern cannot simply alternate between outputting stressed and unstressed syllables, because the same syllable may be output differently if it is final or not. The transducer is given here:

(27)



The transducer must wait whenever it takes an odd syllable (other than the first) as its input. This waiting behavior was also observed for LHOR stress in (114) above. In order to know what to output for a given odd syllable, the stress function needs to know – is this the end of the word? As such, q_2 corresponds to some number of output $\acute{\sigma}\sigma$ sequences, but q_3 is a waiting state – an "odd syllable" state. It corresponds to some odd-numbered input syllable, and does not correspond to any additional output beyond that encoded in q_2. This means that information other than recent local outputs is needed to compute the function, and so it is not OSL. A derivation for five and six-syllable words is given here:

(28) 5 syllable

input:		σ	σ	σ	σ	σ
states:	q_0	$\rightarrow q_1$	$\rightarrow q_2$	$\rightarrow q_3$	$\rightarrow q_2$	$\rightarrow q_3$
output:		$\acute{\sigma}$	σ	λ	$\acute{\sigma}\sigma$	$\lambda\ \ \sigma$

6 syllable

input:		σ	σ	σ	σ	σ	σ
states:	q_0	$\rightarrow q_1$	$\rightarrow q_2$	$\rightarrow q_3$	$\rightarrow q_2$	$\rightarrow q_3$	$\rightarrow q_2$
output:		$\acute{\sigma}$	σ	λ	$\acute{\sigma}\sigma$	λ	$\acute{\sigma}\sigma$

An important question then, is whether or not this properly subsequential characterization of Pintupi is appropriate. It suggests that the subsequential boundary bisects the typology of iterative stress patterns, and that adding a non-finality generalization to basic iteration of stress somehow makes the pattern “long-distance”, like the pattern of LHOR. It is also an implicit claim that any subsequential pattern may appear in the typology of iterative stress. However, it is not the case that any subsequential function is also a possible stress pattern. For example, a pattern that stresses every odd heavy syllable in a word is definable with a subsequential function:

(29)

HHLLLHH	$\mapsto$	ḢHLLLḢH
HHHHHH	$\mapsto$	ḢHḢHḢH
LLLLHHH	$\mapsto$	LLLLḢHḢ
LḢHLLLLḢL	$\mapsto$	LḢHLLLLḢL

Such a pattern is clearly pathological and should not be included in the predicted typology of stress patterns. To provide a better hypotheses regarding the typology of iterative stress, I argue that patterns like Pintupi are fundamentally local in a way that properly subsequential patterns are not. To make this fact of the stress computation explicit, it is proposed in Chapter 4 that the best characterization of iterative stress is as the composition of an OSL and EO function into a combined map. The apparent need for lookahead in some iterative patterns suggests subsequential functions, but the compositional analysis shows that reference to this level of power is unnecessary – iterative patterns are merely the composition of different stress primitives that are formally local. I demonstrate that the OSL plus EO composed map is a better hypothesis for iterative stress than one that claims subsequential power, as it excludes patterns like those in (29).

3.3.2 OSL

Another relevant class of functions is the output strictly local (OSL) class (Chandlee 2014; Chandlee et al. 2015; Chandlee and Heinz 2018). Intuitively, an OSL function is a function that is calculated based entirely on information in the output string. Processes where application of a rule is iterative, such as spreading of a feature, are in general OSL (Chandlee 2014; Dolatian et al. 2021). OSL functions are separated into left and right counterparts (L-OSL and R-OSL). L-OSL functions can express different generalizations than R-OSL functions, as they process the string in opposite directions and so describe iterative patterns only in their respective directions. This output-centeredness is reflected in OSL transducers, where the states in the machine correspond to sequences of symbols read in the output that are relevant to the changes imposed by the function.

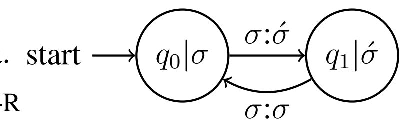
An example of an L-OSL function is left to right iteration of stress, as found in Murinbata (Street and Mollinjin 1981):

(30)
$$\begin{array}{lll} \sigma\sigma & \mapsto & \acute{\sigma}\sigma \\ \sigma\sigma\sigma & \mapsto & \acute{\sigma}\sigma\acute{\sigma} \\ \sigma\sigma\sigma\sigma & \mapsto & \acute{\sigma}\sigma\acute{\sigma}\sigma \\ \sigma\sigma\sigma\sigma\sigma & \mapsto & \acute{\sigma}\sigma\acute{\sigma}\sigma\acute{\sigma} \\ \sigma\sigma\sigma\sigma\sigma\sigma & \mapsto & \acute{\sigma}\sigma\acute{\sigma}\sigma\acute{\sigma}\sigma \\ & \ldots & \end{array}$$

Placement of iterative stress depends on the output string. Intuitively this is because the next application of stress depends on where it was last placed, and so this information can only be located in the output – the input contains no stresses. As such, the states of an OSL transducer correspond to the most recent output. A transducer for iteration of stress in Murinbata is as follows:

(31)

a.



L-R

The transducer outputs the first syllable it encounters with stress, moving to state q_1. The next syllable is output as unstressed, and the transduction moves back to q_0. In other words, the current state in an OSL transducer encodes information about the *output* string. Reliance on the output is a defining feature of OSL transducers. Formally, a process is called OSL_k if there is some k such that any two strings with the same k–1 suffixes in the output string arrive in the same state. The function represented in (38) is thus OSL_2 – any string for which the transduction has just output $\acute{\sigma}$ arrives in q_1, and any string for which the transduction has just output σ arrives in q_2. The process repeats for the remainder of the word, applying stress iteratively. The following shows a derivation for a six-syllable word:

(32) 5 syllable

input:	$\sigma \quad \sigma \quad \sigma \quad \sigma \quad \sigma$
states:	$q_0 \rightarrow q_1 \rightarrow q_0 \rightarrow q_1 \rightarrow q_0 \rightarrow q_1$
output:	$\acute{\sigma} \quad \sigma \quad \acute{\sigma} \quad \sigma \quad \acute{\sigma}$

6 syllable

input:	$\sigma \quad \sigma \quad \sigma \quad \sigma \quad \sigma \quad \sigma$
states:	$q_0 \rightarrow q_1 \rightarrow q_0 \rightarrow q_1 \rightarrow q_0 \rightarrow q_1 \rightarrow q_0$
output:	$\acute{\sigma} \quad \sigma \quad \acute{\sigma} \quad \sigma \quad \acute{\sigma} \quad \sigma$

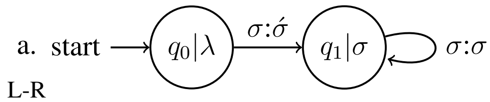
Any output string ending in σ will land in q_0, while any output string ending in $\acute{\sigma}$ will be in state q_1. This makes the output-oriented nature of OSL functions explicitly clear – the states do not encode any information about input syllables, but all output strings ending in σ or $\acute{\sigma}$ will land in their respective state.

A consequence of this is that OSL transducers cannot have a "waiting" transition between two states in which the empty string λ is written to the output, as seen in the LHOR transducer in (114). This configuration in a transducer means that there are infinite pairs of strings with the same output suffix that will land in different states – q_2 or q_3 in (114) – which is an explicit indication that the function itself is not OSL.

As they pertain to stress, OSL functions correspond to some iterative stress patterns, but not all. As will be shown in Chapter 4, for stress assignment where iteration is the only factor, an OSL mapping is sufficient. Placement of further stresses in an iterative chain depends on the previous stress in the output – the input of bare syllables is not enough information. However, if there are other stress phenomena at play, such as clash or nonfinality, an OSL function alone is insufficient. This is because such patterns require a small amount of lookahead – information other than the most recent output, which is not available to OSL functions. The current work proposes that in such cases, the stress map is the composition of an OSL and EO function, where the first handles the iteration of stress and the second accounts for other factors in the pattern. This provides a unified account of iterative stress patterns that is based on the properties of their computation.

3.3.3 ISL

Another relevant class of functions is the input strictly local (ISL) class (Chandlee 2014; Chandlee and Heinz 2018). The input-centered counterpart of OSL functions, ISL functions are calculated based entirely on information in the input string. These functions are extremely relevant to phonology, including common processes such as deletion and epenthesis. This input-centeredness is reflected in ISL transducers, where the states in the machine correspond to sequences of symbols read in the input that are relevant to the changes imposed by the function, just as the states in an OSL function encode output information. Any bounded non-iterative pattern can be described using an ISL function. The following transducer places initial stress, as in the pattern of Afrikaans (Donaldson 1993):

(33)


The first syllable is stressed in the transition from q_0 to q_1. All other syllables are left unstressed in the self-loop at q_1. An example derivation for an input $\sigma\sigma\sigma\sigma$ is given here:

(34)

input:	σ	σ	σ	σ
states:	$q_0 \rightarrow q_1$	$\rightarrow q_1$	$\rightarrow q_1$	$\rightarrow q_1$
output:	$\acute{\sigma}$	σ	σ	σ

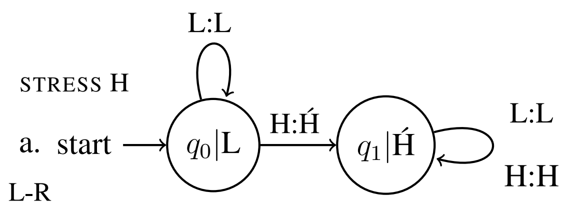
As single stress patterns such as Afrikaans depend on a small window of information at a word edge, they can all be calculated using only input information, whether they are QI or QS. While many phonological functions are ISL, I demonstrate in Chapter 4 that further restricting the power of certain atomic stress functions to be EO, rather than ISL, provides a better typological fit for what is observed in stress typology.

3.3.4 SP

In §3.1, it was demonstrated that the DTO “leftmost-heavy or rightmost” (LHOR), found in Kwakw’ala (Hayes 1995), can be calculated by a subsequential function. Remember, however, that full subsequential power overgenerates by predicting pathological counting patterns such as the “stress every odd heavy syllable” pattern described in (29). A legitimate question then is whether reference to subsequential functions is necessary, or if some more restrictive characterization of the “atoms” of long-distance patterns can be provided. The answer to this question comes from the *strictly piecewise* (SP) class of functions (Burness and McMullin 2020), an extension of the strictly piecewise languages (Heinz 2009; Rogers et al. 2010). This class of functions is sufficient to capture many long-distance phonological patterns but is more restrictive than full subsequential power. Intuitively, in an SP function, each input symbol has a *consistent effect* on the behavior of other inputs that appear further along in the string. An example comes from sibilant harmony in Aari (Hayward 1990), where an underlying /s/ surfaces as [ʃ] if another [ʃ] appears at any earlier point in the word. This means that [ʃ] determines the behavior of all following underlying /s/ in a consistent, long-distance manner – only [ʃ] will ever surface.

SP functions can describe the long-distance aspect of unbounded patterns. For example, consider the following transducer for a STRESS H function, that stresses the first heavy syllable in a word and leaves other syllables unchanged:

(35)



The first heavy syllable STRESS H encounters is stressed in the transition from q_0 to q_1. Thus, q_1 corresponds to having output a stress at some previous point in the word, and all further inputs are output as stressless. In other words, outputting of a stress has a consistent effect on how all other input symbols further along in the word surface. Note that this is not local – the stressed H maintains this effect for all other symbols, no matter how far they occur from the stress. The function is SP. Below it is demonstrated that an SP hypothesis for long-distance stress patterns is preferable to a hypothesis that allows full subsequential power because, just as with EO functions in bounded patterns, a better fit to the observed typology is achieved. A more in-depth description of SP functions and their relevance to the atomic properties of long-distance stress patterns is given in Chapter 6.

Chapter 4: An Atomic Theory of Stress

4.1 Introduction

This chapter explicitly details the formal properties of the theory of stress proposed in this dissertation. I propose that the surface stress patterns we observe can be factored into one or two simple individual steps or "atoms" that then compose together via function composition to create the complete input-output map. As described in Chapter 1, this follows conceptually in the long tradition of serial analyses of stress, but adds insight regarding computational complexity as defined in formal language theory. Specifically, the well-defined notion of complexity from FLT allows us to state the class of potential atoms that could occur in stress, placing important restrictions on the composition of those atoms. This leads to a predicted typology that is tangibly improved over a comparable theory that does not enforce the same restrictions. This is not to say that other theories of stress do not pursue restrictiveness or concern themselves with typological predictions. Rather, the difference is that the method of measuring complexity in this approach comes from the mathematical definition of complexity provided by FLT. Ultimately, notions of restrictiveness from substantive phonological theories are needed as well to reign in expressivity when pure computation alone is insufficient. The two conceptions of restrictiveness can then exist in tandem – computational restrictions constraining the overgeneration of formal theoretical devices, and the typological predictions of the theoretical devices providing additional restrictiveness in areas where computation alone makes inaccurate predictions.

I start by introducing the novel "edge-oriented" (EO) class of functions. EO functions enforce a notion of locality by limiting application to some fixed distance from a word edge. As many stress generalizations are tethered to a morphological boundary, edge-oriented functions provide a formal computational instantiation of a typologically real, substantive aspect of phonological grammar.

The use of compositions of functions to describe stress maps is further justified by demonstrating the restrictiveness of the predicted stress typology. Individual stress atoms such as iteration and non-finality belong to relatively simple function classes, but unless stress maps are analyzed as a stepwise application of these atoms, a more powerful function is required – a single-function analysis cannot capture the total patterns without use of a function belonging to a more expressive class. It is demonstrated that a hypothesis of more powerful stress atoms is both unnecessary and leads to unwanted typological predictions. The atoms that coalesce to produce the patterns that are observed in stress typology need not exceed the input or output strictly local, edge-oriented, or strictly piecewise boundary. All known stress patterns can be described via the composition of functions from these complexity classes.

Finally, I describe the range of possible primitives of stress patterns and how they may be combined, as well as the typological predictions that result from their combination. I examine in detail both accurate and *in*accurate predictions made by the theory, and attempt an assessment of what the inaccuracies have to say about the theory and its situation within phonology in general. This is done for each possible combination of atom types. This highlights that the areas in which the theory overgenerates can be constrained with reference to substantive theoretical devices from phonology such as foot structure. What this approach instead contributes is a detailed analysis of how computation *in particular* can constrain the predicted typology of stress in an effective manner.

4.2 Edge-oriented patterns

In this section I introduce the "edge-oriented" (EO) functions, which play an important role in the stress patterns examined in this dissertation. They provide a more restrictive hypothesis for non-iterative stress than ISL, implement the cleanup functions necessary for iterative patterns when iteration of stress alone is unable to capture a pattern, and also appear in the composition of unbounded patterns.

4.2.1 EO definition

Intuitively, EO functions are those for which any changes made to the input string only occur in a fixed window at the edge of the word. Prioritizing the word edge is a property of stress patterns in general – single stress patterns (final stress, penultimate stress, etc.) never place stress further than three syllables away from an edge, nor do iterative patterns begin iteration of stress from further than that distance (Hyman 1977; Gordon 2002).

More formally, in an FST for an EO function, only a finite number of strings can reach a non-identity transition. As such, given a string of sufficient length, the transduction always reaches a state where no further changes can be made to the word. The transitions from these states are self-loops that provide identity mappings – the restriction prevents a return to a state with a non-identity transition. In graph theoretic terms, these properties mean that the largest subgraph of an EO transducer that can reach a non-identity transition is a directed acyclic graph.

(36) **Definition – EO functions:** EO functions are those describable with an FST for which only a finite number of strings arrive in a state with a non-identity transition exiting the state.

Requiring that only a finite number of input sequences can be followed by a change ensures that EO functions can only make alterations to their input in a fixed window at the word edge. Remembering the Latin transducer in Chapter 3, for example, $L\#, H\#, LL\#, LH\#$

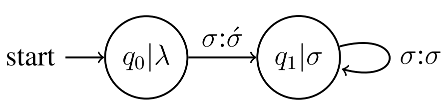
are the only input sequences that can reach a state with a non-identity transition – q_1 and q_2. No transitions return to earlier states, as this would permit an in principle unlimited number of changes to applied to an input, eliminating a necessary property of EO functions that ensures their restrictiveness. In the Latin transducer, arriving in q_4 indicates that the necessary span to determine the behavior of the function has been read. The function does not "care" about what symbols might come after – they are always left unchanged[1]. In the EO transducers throughout this thesis, such states are labeled with the maximal span of input symbols that lead to it, making explicit the size of the window that determines the behavior of the function. For the Latin transducer, $q4$ is labeled with three syllables, as this three-syllable window at the right word edge is the maximal span needed to determine the correct output of the Latin stress pattern.

4.2.2 Motivations and relation to existing classes

EO functions are ISL, as they are local to the word edge in the input, and some are OSL. Not every OSL or ISL function is EO. The difference emerges partially from that fact that OSL and ISL are defined in terms of output and input substrings, while EO is defined in terms of finite distance to the word edge. In other words, they are calculated based on distinct intensional requirements, even when the extensional results of the functions are the same.

The initial stress pattern from Chapter 3, repeated here, is EO – the only substring leading to a non-identity transition is the beginning of the word i.e. the empty string λ:

(37) a. L-R



Once in q_1, no further changes are made. In fact, as the output of all bounded non-iterative patterns is determined by some local span of syllables near a word boundary, all bounded non-iterative patterns can be described with an EO function.

[1]This is reminiscent of the *definite* class of formal languages (Salomaa 1969).

EO functions also encode the atomic properties of iterative stress such as non-finality, clash, and lapse in a straightforward and restrictive manner, as they are always tethered to an edge in attested iterative patterns. In a non-finality function, for example, the only string leading to a non-identity transition is the right word boundary, #. The following syllable is output as unstressed, and the rest of the word is left unchanged. The behavior of the function is determined by a small, local window at the word edge. EO functions provide a direct characterization of this property of stress atoms. EO thus provides a testable hypothesis for the atoms of iterative stress – a substantive claim couched in computational terms that goes beyond categorization of surface patterns to explain *why* iterative patterns appear the way that they do.

In the typology of patterns that are formally non-local, the role of EO functions are naturally diminished, as long-distance patterns cannot be tied to a word boundary. For example, EO functions do not appear in the composed analyses of DTS or Pulaar-like patterns, as shown in Chapter 5. For DTO patterns, however, an EO function does serve to place the default stress. As such, it is noteworthy that, to some extent, these simple EO functions that target a word edge are present in both local and non-local stress typology.

EO is motivated by the fact that ISL alone does not provide a sufficiently restrictive characterization of bounded non-iterative or iterative stress atoms. For example, the cleanup functions necessary for iterative stress maps are both ISL *and* OSL[2]. A non-finality requirement could be computed input-locally by searching for the substring "$\acute{\sigma}\#$" i.e. final stress in the input. It could also be computed output-locally if the function instead enforces "$\sigma\#$" in the output at the end of the word, whether the input was stressed or not. Thus, a limitation to OSL plus ISL fails to achieve the correct restrictiveness, because the particular cleanup functions needed are both ISL and OSL. With no further restrictions, this is then an implicit hypothesis that cleanup functions may be OSL in general. In §3 I demonstrate that OSL plus EO compositions provide better typological predictions for iterative stress

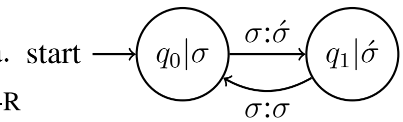
[2] I thank Jane Chandlee for pointing this out.

than when OSL is composed with OSL or ISL.

It is not difficult to determine that the class of EO functions does not overlap completely with the OSL or ISL class. An OSL function handling iteration of stress is not EO, as it applies stress throughout the entire word, as shown in the transducer from Chapter 3:

(38)

a.



L-R

As the transducer makes obvious, there is *no* point for the OSL iteration function at which changes cannot be applied to the string. When the transduction "resets" by entering q_0 every other syllable, stress is applied upon reading the next syllable. Similarly, an ISL function deleting a syllable-final consonant may apply this change at any point in the word where the triggering environment occurs. In both cases, the string is altered for its entire length.

An EO function, however, can only stipulate the specific location for stress in a bounded window at the word edge i.e. stress n, stress $n + 2$, stress $n + 4$ etc. for all cases of stress. Not only is this insufficient to handle words of all lengths in a finite manner, it is not really iteration at all — it is an enumeration of all possible stress locations. Note that this is the same reason that an ISL function cannot describe iterative stress. Similarly, any attempt to use EO to describe an ISL function that may apply anywhere in the word results in an enumeration of the cases as the size of the word grows. In other words, while all EO functions are ISL and some are OSL, their restriction to the word edge means they are not general enough to describe all ISL or OSL patterns. For the cleanup functions in iterative stress maps discussed in this paper, this is a desirable property that increases the restrictiveness of the theory while still capturing the target patterns.

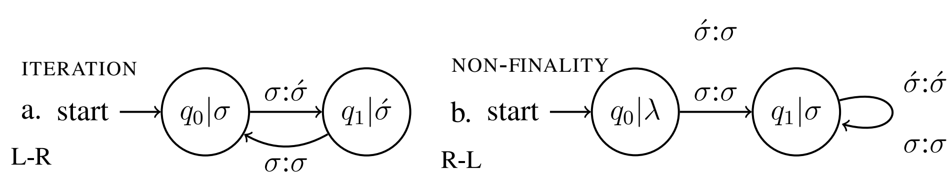
4.3 Function composition

When two functions are composed, the output of the first function becomes the input for the second function. Application of successive rules in a derivational phonological analysis, such as in SPE, is a kind of function composition (Johnson 1972; Kaplan and Kay 1994). I argue that the stress patterns analyzed here are best expressed as the composition of functions instantiating the individual stress atoms that comprise the pattern.

4.3.1 Compositional analyses of stress

For iterative stress, one OSL function and one EO "cleanup" function that acts in place of lookahead compose to derive the correct output. For long-distance patterns, an SP function composes with another function of a certain type, depending on the particular pattern. Thus, the phonological grammar is comprised of these individual stress atoms, and the grammar knows which order they must apply in. To demonstrate, recall the analysis of Pintupi from Chapter 3. Rather than referring to full subsequential power to describe the Pintupi stress function, here it is analyzed as a breakdown into an OSL function dubbed ITERATION that achieves the basic iteration of stress and an EO function dubbed NON-FINALITY that removes a final stress when one is present. Consider the following pair of transducers. Here and for the remainder of the dissertation, where there are multiple transducers, they apply in the order of their alphabetical label:

(39)



The function ITERATION here is the L-OSL transducer in (39a). It is identical to that of Murinbata – it applies stress to every odd syllable. The EO function NON-FINALITY in (39b) takes the output of the first function as its input. If the first syllable it encounters is stressed, it removes the stress. Otherwise, it outputs syllables faithfully. Since the first symbol it reads is really the last syllable, this amounts to deletion of stress on a final syllable, encoding the non-finality property of Pintupi. The only input substring that leads to a non-identity transition is the empty string λ, and so NON-FINALITY is EO. Reaching q_1 indicates that the relevant span of input symbols (one) to calculate the pattern has been seen, and so the only transitions leaving q_1 are identity-mapping loops back to q_1.

Examples of how the two functions interact to produce the correct outputs are given here, using a five and six-syllable form. For the compositional analyses in this paper, these diagrams should be read top to bottom – note how the output of ITERATION lines up vertically with and matches the input for NON-FINALITY:

(40) 5σ — 6σ

ITERATION:

	5σ	6σ
input:	$\sigma \quad \sigma \quad \sigma \quad \sigma \quad \sigma$	$\sigma \quad \sigma \quad \sigma \quad \sigma \quad \sigma \quad \sigma$
states:	$q_0 \rightarrow q_1 \rightarrow q_0 \rightarrow q_1 \rightarrow q_0 \rightarrow q_1$	$q_0 \rightarrow q_1 \rightarrow q_0 \rightarrow q_1 \rightarrow q_0 \rightarrow q_1 \rightarrow q_0$
output:	$\acute{\sigma} \quad \sigma \quad \acute{\sigma} \quad \sigma \quad \acute{\sigma}$	$\acute{\sigma} \quad \sigma \quad \acute{\sigma} \quad \sigma \quad \acute{\sigma} \quad \sigma$

NON-FINALITY:

	5σ	6σ
input:	$\acute{\sigma} \quad \sigma \quad \acute{\sigma} \quad \sigma \quad \acute{\sigma}$	$\acute{\sigma} \quad \sigma \quad \acute{\sigma} \quad \sigma \quad \acute{\sigma} \quad \sigma$
states:	$q_1 \leftarrow q_1 \leftarrow q_1 \leftarrow q_1 \leftarrow q_1 \leftarrow q_0$	$q_1 \leftarrow q_1 \leftarrow q_1 \leftarrow q_1 \leftarrow q_1 \leftarrow q_1 \leftarrow q_0$
output:	$\acute{\sigma} \quad \sigma \quad \acute{\sigma} \quad \sigma \quad \sigma$	$\acute{\sigma} \quad \sigma \quad \acute{\sigma} \quad \sigma \quad \acute{\sigma} \quad \sigma$

ITERATION and NON-FINALITY interact to produce the correct outputs for Pintupi. It is merely the composition of local stress atoms, and requires no reference to a properly subsequential function. This is because, at a fundamental computational level, it is not a long distance pattern.

4.3.2 Composition and restrictiveness

For iterative patterns, the restriction to EO cleanup functions in the composition is a crucial requirement. Composition in general is powerful in that it can result in a mapping that is more expressive than any of its constituent functions. For example, Heinz and Lai (2013) show that sour grapes harmony (Padgett 1995; Wilson 2003) can be derived from the composition of two subsequential functions working in opposite directions, even though sour grapes itself is not subsequential. McCollum et al. (2020) further demonstrate that restrictions on interaction are necessary to avoid overgeneration in the interaction of subsequential functions.

These effects are in part because of how directionality affects function composition. When composing functions, the directionality has a direct impact on the expressive power of the generalizations the composition can express. This is because the first function can essentially provide unbounded lookahead by marking the string in some way that informs the behavior of the second function.

Thus, adherence to unidirectionality in the functions constrains the complexity – it is known that the composition of two unidirectional subsequential functions cannot produce a combined map that is more expressive than one that a single subsequential function could produce (Elgot and Mezei 1965). Thus, we say that subsequential functions are closed under composition, as long as the functions in the composition operate in the same direction. Though there is currently no proof, it is possible that the same may hold for OSL functions as well.

However, this is *not* true of compositions of subsequential or OSL functions operating in opposite directions. Heterodirectional subsequential functions can describe the stress map for default-to-same stress patterns, which are not subsequential (Hao and Anderson 2019; Koser and Jardine 2020b). The composition of a left and right OSL function can also describe non-subsequential functions. Take for example the hypothetical sour grapes stress pattern described in §2, repeated here:

(41) $\acute{\sigma}\sigma$

$\sigma\sigma\sigma$

$\acute{\sigma}\sigma\acute{\sigma}\sigma$

$\sigma\sigma\sigma\sigma\sigma$

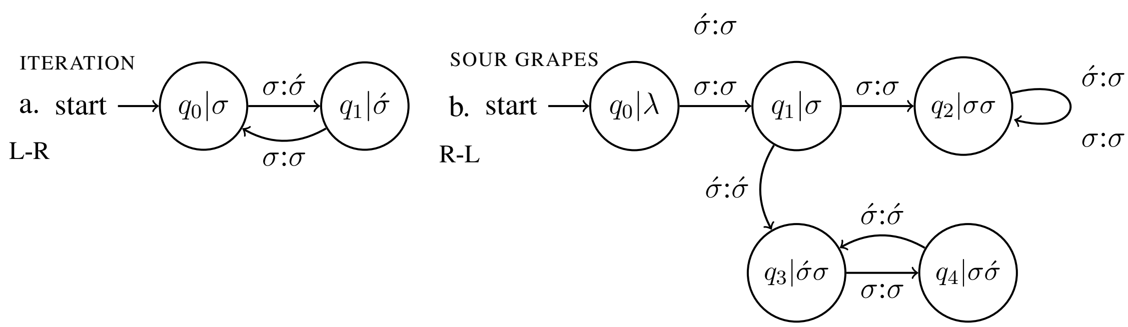
$\acute{\sigma}\sigma\acute{\sigma}\sigma\acute{\sigma}\sigma$

$\sigma\sigma\sigma\sigma\sigma\sigma\sigma$

$\acute{\sigma}\sigma\acute{\sigma}\sigma\acute{\sigma}\sigma\acute{\sigma}\sigma$...

Stress iterates through the word only in the case that it is of even length. Though parity counting is in general properly regular for stringsets (Heinz 2007b; Rogers et al. 2013; Graf 2017), such a stress map can be described as the composition of two heterodirectional OSL functions. The functions are represented by the following transducers:

(42)



(109a) is ITERATION, just as seen in Murinbata and Pintupi. It reads the string left to right, placing stress on every other syllable throughout the word starting with the first syllable. (109b) is the SOUR GRAPES function, producing the pathological aspect of the combined map. Application of ITERATION means that words of even length will end in an unstressed syllable, while words of odd length will end in a syllable bearing stress. SOUR GRAPES makes explicit use of this markup information. Reading the string right to left, the final syllable is always output as unstressed in the transition to q_1. The next syllable – the penult – gives the second function the crucial information it needs. A stressed penult indicates a string of even length, and so the transduction moves to q_3 which, alternating with q_4, faithfully outputs the input unchanged, preserving the stresses. An unstressed penult instead indicates an odd parity word, and so the transduction moves to q_2, where all stresses are removed. The composition of the two OSL functions correctly derives the pathological stress map, shown here for a five and six-syllable form:

(43)

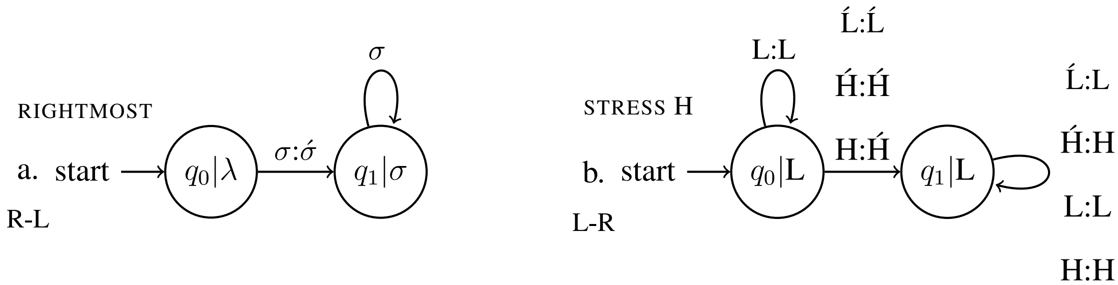
	5σ	6σ
ITERATION:		
input:	$\sigma \quad \sigma \quad \sigma \quad \sigma \quad \sigma$	$\sigma \quad \sigma \quad \sigma \quad \sigma \quad \sigma \quad \sigma$
states:	$q_0 \rightarrow q_1 \rightarrow q_0 \rightarrow q_1 \rightarrow q_0 \rightarrow q_1$	$q_0 \rightarrow q_1 \rightarrow q_0 \rightarrow q_1 \rightarrow q_0 \rightarrow q_1 \rightarrow q_0$
output:	$\acute{\sigma} \quad \sigma \quad \acute{\sigma} \quad \sigma \quad \acute{\sigma}$	$\acute{\sigma} \quad \sigma \quad \acute{\sigma} \quad \sigma \quad \acute{\sigma} \quad \sigma$
SOUR GRAPES:		
input:	$\acute{\sigma} \quad \sigma \quad \acute{\sigma} \quad \sigma \quad \acute{\sigma}$	$\acute{\sigma} \quad \sigma \quad \acute{\sigma} \quad \sigma \quad \acute{\sigma} \quad \sigma$
states:	$q_2 \leftarrow q_2 \leftarrow q_2 \leftarrow q_2 \leftarrow q_1 \leftarrow q_0$	$q_3 \leftarrow q_4 \leftarrow q_3 \leftarrow q_4 \leftarrow q_3 \leftarrow q_1 \leftarrow q_0$
output:	$\sigma \quad \sigma \quad \sigma \quad \sigma \quad \sigma$	$\acute{\sigma} \quad \sigma \quad \acute{\sigma} \quad \sigma \quad \acute{\sigma} \quad \sigma$

Though this is a logically possible stress map, it is clearly pathological and is disallowed by the theory of iterative stress adopted here, which only permits OSL plus EO compositions. Only one OSL function is needed to achieve the basic iteration of stress, while the EO cleanup functions encode edge-adjacent stress generalizations that complete each individual pattern. SOUR GRAPES is clearly not EO, as it may delete stresses that occur at any point in the word. Thus, allowing multiple OSL functions is not only unnecessary for unidirectional iterative stress, it allows the generation of pathological patterns such as sour grapes stress. Adherence to OSL plus EO compositions thus places important constraints on the predicted typology of iterative patterns.

4.3.3 Long-distance compositions

Similarly, though long-distance patterns such as LHOR require subsequential power as a single function map, as shown in Chapter 3, it can also be expressed as the composition of an EO function and an SP function. The EO function places the default “rightmost” stress, and the SP function stresses the leftmost heavy and ensures that all other syllables are output with no stress. Consider the following transducers:

(44)



The first, EO function is RIGHTMOST[3], which stresses the final syllable in the word, representing the default rightmost stress. The only substring leading to a non-identity transition is λ. The second, SP function is STRESS H, but modified for a compositional analysis. It stresses the first heavy syllable it encounters in the transition from q_0 to q_1. This transition also accounts for all-light forms and forms where the only heavy is the final syllable, as the final stress created by the first function should be preserved in these cases. Thus, q_1 corresponds to having output a stress at some previous point in the word, and all further inputs are output as stressless, including removal of the default stress created by RIGHTMOST. In other words, outputting of a stress has a consistent effect on how all other input symbols further along in the word surface, and the function is SP. The following examples demonstrate the combined map:

(45)

		(44a)		(44b)
a. LLLLLL	$\mapsto$	LLLLLĹ	$\mapsto$	LLLLLĹ
b. LHLLHL	$\mapsto$	LHLLHĹ	$\mapsto$	LH́LLHL
b. LLLLLH	$\mapsto$	LLLLLH́	$\mapsto$	ĹLLLLL

Thus, long-distance patterns may also be treated as the composition of simpler individual atoms, even when a single-function analysis requires properly subsequential power. Just as with iterative patterns, this eliminates unwanted typological predictions while still allowing attested patterns such as LHOR to be described. This is because SP functions are less expressive than subsequential functions – though SP functions describe long-distance processes, they cannot derive a pathological pattern such as "stress every odd heavy syl-

[3]Note this is identical to FINALITY, which appears later in the thesis.

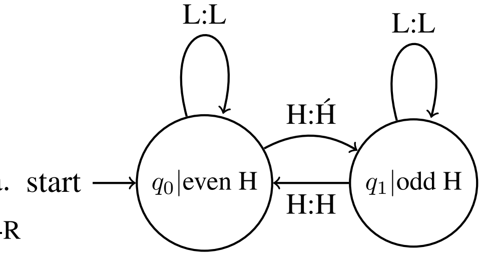
lable". This is precisely because, in such a pattern, the effect of a given input symbol on further inputs is *inconsistent*. To demonstrate, I repeat the pattern here:

(46)

HHLLLHH	$\mapsto$	H́HLLLH́H
HHHHHH	$\mapsto$	H́HH́HH́H
LLLLHHH	$\mapsto$	LLLLH́HH́
LH́HLLLLH́L	$\mapsto$	LH́HLLLLH́L

Consider the behavior of a heavy syllable, for example. In LHOR, outputting of a stressed heavy ensures that every syllable thereafter is unstressed. In the hypothetical pattern in (129), however, this is not the case – any further heavy following a stressed heavy may surface either with or without stress, depending on the even/odd parity of that heavy syllable. In other words, both [H́...H] and [H́...H́] are licit configurations because the pattern depends on an overt parity count. In an SP function, given two words comprised of the same set of preceding symbols, any further input symbol must surface in the same way in both words. This is clearly not the case in "stress every odd heavy" – given two words LH́HLL and LH́HH́L, for example, an additional input H surfaces with stress in the former word and without stress in the latter, even though the set of preceding symbols is the same. Though the pattern is not SP, it *is* subsequential:

(47)

a. 

L-R

The first heavy syllable is stressed in the transition from q_0 to q_1. When another heavy is seen, the transduction returns to q_0 and the cycle resets. Light syllables never receive stress, but may intervene unboundedly between heavy syllables, making the pattern non-local. It is properly subsequential, and not SP for the reasons described above.

Just as OSL plus EO compositions provide a tighter characterization of iterative patterns than the subsequential class, an EO plus SP composition avoids the overgeneration possible with subsequential functions for DTO patterns as well. This is analogous to the result of Rogers et al. (2013), who found that stress patterns as formal *languages* i.e. stringsets could be factored into combinations of local and piecewise constraints. A full picture of the atomic properties of long-distance patterns is provided in Chapter 5.

4.3.4 Relation to Idsari (1992)

It should be acknowledged that, conceptually, the analyses in the current work bear similarity to – and are inspired by – the theoretical stress work of Idsardi (1992). Idsardi demonstrates how a modest set of directionalized parameters can account for a wide range of attested stress patterns. These parameters function like the atoms of this work, combining to derive the surface generalizations via composition. For example, the "iterative constituent construction" parameter (ICC; p. 18) of Idsardi progressively iterates markers from which stress can later be projected throughout the word. The OSL iteration functions appearing in this work perform a similar task, and ICC is indeed an OSL function. While Idsardi is concerned mostly with how stress primitives derive the surface patterns, the work here contributes an exact computational characterization of the primitives and demonstrates what this computational characterization tells us about stress typology and restrictiveness.

Idsardi also introduces a small array of other mechanisms that assist in the derivation of the pattern when plain iteration fails. Though the devices he proposes typically block iteration from occurring in a given location, rather than delete inappropriately-placed stresses as here, the goals of the proposed mechanisms are the same. Idsardi's "Edge" parameter, for example, places boundaries in a way that accounts for non-finality effects, and "Avoid" accounts for lapses in certain languages, just as the EO cleanup functions here make the needed adjustments when iteration alone fails to capture the complete pattern. However, going beyond a surface description based on what patterns should be derived, proposing the EO class offers an explanation as to why those patterns should exist in the first place – they are comprised of atoms that adhere to the computational restrictions outlined in this work i.e. being OSL, EO, or SP.

For QS systems, Idsardi also assumes a basic L or H syllable inventory, abstracting away from details regarding their internal structure or properties that led to their formation. Here as well, this allows for a focus on how the stress system interacts on a computational level with a QS inventory without overcommitting to one or another analysis regarding syllable formation. This approach also highlights the computational similarity between QS and QI systems, which reduces to a difference in alphabet size under this theory. This demonstrates that, despite surface differences, the underlying computations are largely the same.

While this thesis studies string-to-string mappings and Idsardi's system achieves its goals through manipulation of constituents on a metrical grid, the spirit is the same – a theory of how a small set of simple parametric computations derive a great many stress patterns. Idsardi even represents a special rule for Diyari stress with a finite state transducer (p. 72). The current work contributes an investigation of the exact computational complexity properties of the necessary parameters (atoms), what the proper notion of restrictiveness may be, and the implications for the predicted typology of such a theory of stress.

4.4 Stress atoms

In this section I examine the typological consequences of stress atoms from each complexity class for this theory of stress – both individually and as part of a compositional analysis. Recall the following table from Chapter 1:

(48)

Local:	**single stress**	**iterative stress**	**bidirectional stress**
	EO	OSL + EO	OSL + OSL
Non-local:	**DTO**	**DTS**	**suprabinary**
	EO + SP	SP + SP	SP + SP

This table catalogues the ways in which combinations of stress atoms from the EO, OSL, and SP classes describe a wide range of stress patterns. In the following chapters, I give detailed analyses of these stress compositions. The theory correctly predicts the existence of all attested patterns that were surveyed. An important question is – what else does it predict? Adopting an entire function class as a hypothesis for a set of patterns tacitly allows for any function from the class to potentially serve as a stress atom unless further restrictions are imposed. As such, it is important to consider where the theory makes predictions that are incorrect. Where does the theory overgenerate, what can we learn from the nature of the overgeneration, and how might it be constrained? The remainder of this chapter addresses this question directly.

4.4.1 EO atoms

Single atom

EO atoms correspond to numerous phonological generalizations. Any generalization that is tied to a finite window at a word edge can be calculated with an EO function. For stress, this means that any of the attested "single stress" patterns can be described this way, as they all occur a fixed distance from the word edge (never further than three syllables). Adopting EO functions thus correctly predicts the existence of these stress patterns. Initial stress, for example, is EO, as the function need only see the left word boundary # before applying a change and leaving the rest of the string unchanged:

(49) $\#\sigma\sigma\sigma\sigma \mapsto \acute{\sigma}\sigma\sigma\sigma$

$\#\sigma\sigma\sigma\sigma\sigma \mapsto \acute{\sigma}\sigma\sigma\sigma\sigma$

$\#\sigma\sigma\sigma\sigma\sigma\sigma \mapsto \acute{\sigma}\sigma\sigma\sigma\sigma\sigma$

$\#\sigma\sigma\sigma\sigma\sigma\sigma\sigma \mapsto \acute{\sigma}\sigma\sigma\sigma\sigma\sigma\sigma$

There are areas where EO functions overgenerate, in part due to the very nature of their definition – while EO functions are limited to occur in a certain location, they are *not* limited in what *kind* of changes can be made within that location. An EO function could, for example, stress every syllable that occurs in the edge window, as in this hypothetical "stress final four" pattern:

(50) $\sigma\sigma\sigma\sigma\# \mapsto \acute{\sigma}\acute{\sigma}\acute{\sigma}\acute{\sigma}$

$\sigma\sigma\sigma\sigma\sigma\# \mapsto \sigma\acute{\sigma}\acute{\sigma}\acute{\sigma}\acute{\sigma}$

$\sigma\sigma\sigma\sigma\sigma\sigma\# \mapsto \sigma\sigma\acute{\sigma}\acute{\sigma}\acute{\sigma}\acute{\sigma}$

$\sigma\sigma\sigma\sigma\sigma\sigma\sigma\# \mapsto \sigma\sigma\sigma\acute{\sigma}\acute{\sigma}\acute{\sigma}\acute{\sigma}$

As EO allows such functions by definition, no appeal to computation alone can account for the overgeneration. Such patterns can be excluded on substantive grounds instead. Such a pattern partially invalidates what is arguably the "point" of stress i.e. to create contrast between syllables and demarcate morphological boundaries – if an entire block of syllables is stressed, than there is no contrast within that block. Additionally, known occurrences of clash are typically limited to two clashing syllables, and so a longer span would be unusual from what is known from attested stress typology. In other words, "stress final four" does not have the appearance of a stress pattern at all. This could be explained by incorporating foot structure into the theory, which typically avoids such configurations by preferring maximal binary parsing of syllables in the word. Application of foot structure is an OSL process (Dolatian et al. 2021), and so this step of the grammar could apply before location of stress without increasing the complexity of the atoms permitted by the theory. In this conception of the grammar, an OSL process of foot creation first creates foot boundaries in the bare, unstressed string of syllables. Then, location of stress occurs by whatever combination of maximally two EO, OSL, or SP atoms derives the target pattern.

EO plus EO

Compositions of EO atoms can describe "dual stress" patterns (see Gordon (2002)), where a single, fixed stress is located at each edge. An example of such a pattern comes from Udihe (Kormushin 1998), which places stress on both the initial and final syllable. For dual stress patterns, one EO function handles stress on one side of the word, and a second handles the other:

(51)

		INITIAL		FINAL
$\sigma\sigma\sigma\sigma$	$\mapsto$	$\acute{\sigma}\sigma\sigma\sigma$	$\mapsto$	$\acute{\sigma}\sigma\sigma\acute{\sigma}$
$\sigma\sigma\sigma\sigma\sigma$	$\mapsto$	$\acute{\sigma}\sigma\sigma\sigma\sigma$	$\mapsto$	$\acute{\sigma}\sigma\sigma\sigma\acute{\sigma}$
$\sigma\sigma\sigma\sigma\sigma\sigma$	$\mapsto$	$\acute{\sigma}\sigma\sigma\sigma\sigma\sigma$	$\mapsto$	$\acute{\sigma}\sigma\sigma\sigma\sigma\acute{\sigma}$
$\sigma\sigma\sigma\sigma\sigma\sigma\sigma$	$\mapsto$	$\acute{\sigma}\sigma\sigma\sigma\sigma\sigma\sigma$	$\mapsto$	$\acute{\sigma}\sigma\sigma\sigma\sigma\sigma\acute{\sigma}$

Thus, a compositional theory of stress that includes EO atoms in the set of stress primitives accurately predicts the existence of such dual stress patterns. In addition to attested patterns, EO plus EO predicts all logically possible – but unattested – combinations of edge stresses. The absence of certain patterns could be explained as the result of historical factors or random chance. For example, the fact that the attested initial-final pattern of Udihe and the similar-but-unattested peninitial-final pattern have the same computational power suggests that this may be the case, and encourages further research into the stress patterns of undocumented or understudied languages. In addition to typological gaps, EO plus EO compositions predict the same patterns – good and bad – as single EO functions, but occurring at both edges. A hypothetical pattern that stresses the first two and last three is possible under an EO plus EO composition, for example:

(52) INITIAL 2 FINAL 3

$$\begin{array}{lclcl}
\sigma\sigma\sigma\sigma & \mapsto & \acute{\sigma}\acute{\sigma}\sigma\sigma & \mapsto & \acute{\sigma}\acute{\sigma}\acute{\sigma}\acute{\sigma} \\
\sigma\sigma\sigma\sigma\sigma & \mapsto & \acute{\sigma}\acute{\sigma}\sigma\sigma\sigma & \mapsto & \acute{\sigma}\acute{\sigma}\acute{\sigma}\acute{\sigma}\acute{\sigma} \\
\sigma\sigma\sigma\sigma\sigma\sigma & \mapsto & \acute{\sigma}\acute{\sigma}\sigma\sigma\sigma\sigma & \mapsto & \acute{\sigma}\acute{\sigma}\sigma\acute{\sigma}\acute{\sigma}\acute{\sigma} \\
\sigma\sigma\sigma\sigma\sigma\sigma\sigma & \mapsto & \acute{\sigma}\acute{\sigma}\sigma\sigma\sigma\sigma\sigma & \mapsto & \acute{\sigma}\acute{\sigma}\sigma\sigma\acute{\sigma}\acute{\sigma}\acute{\sigma}
\end{array}$$

Note that such patterns arise more as a result of the compositional theory presented here, rather than the nature of EO itself. Just as with single EO functions, these predictions of EO compositions lack certain properties of familiar stress patterns, such as rhythmic distribution of stress and the creation of contrast. This again hints that an appeal to some substantive notion of phonological theory can be made. As above, reference to foot structure and bans on gratuitous unary parsing could alleviate some of the issues. These bans could exist as axiomatic constraints, restricting the pool of atoms that may be selected by the compositional portion of the grammar to only include atomic functions that perform maximally binary foot parsing.

4.4.2 OSL atoms

Single atom

OSL atoms correspond to output-oriented phonological generalizations. Iterative patterns in general are OSL (Dolatian et al. 2021), as the trigger for the next application of the process is located in the output string. Iteration of stress falls under this umbrella, as in the pattern of Murinbata (Street and Mollinjin 1981):

(53)
$$\begin{array}{lcl}
\sigma\sigma\sigma\sigma & \mapsto & \acute{\sigma}\sigma\acute{\sigma}\sigma \\
\sigma\sigma\sigma\sigma\sigma & \mapsto & \acute{\sigma}\sigma\acute{\sigma}\sigma\acute{\sigma} \\
\sigma\sigma\sigma\sigma\sigma\sigma & \mapsto & \acute{\sigma}\sigma\acute{\sigma}\sigma\acute{\sigma}\sigma \\
\sigma\sigma\sigma\sigma\sigma\sigma\sigma & \mapsto & \acute{\sigma}\sigma\acute{\sigma}\sigma\acute{\sigma}\sigma\acute{\sigma}
\end{array}$$

Incorporation of OSL atoms correctly predicts the existence of stress patterns where iteration of stress through the entire word is the only factor. OSL predicts some unattested patterns unless some upper bound on the interval over which iteration occurs is established. For example, a potential "stress every fourth syllable" pattern is OSL:

$$
\begin{array}{lll}
(54) \quad \sigma & \mapsto & \acute{\sigma} \\
 & \vdots & \\
\sigma\sigma\sigma\sigma\sigma & \mapsto & \acute{\sigma}\sigma\sigma\sigma\acute{\sigma} \\
 & \vdots & \\
\sigma\sigma\sigma\sigma\sigma\sigma\sigma\sigma\sigma & \mapsto & \acute{\sigma}\sigma\sigma\sigma\acute{\sigma}\sigma\sigma\sigma\acute{\sigma}
\end{array}
$$

Under OSL, n-ary iteration for any n is permitted. Appealing to metrical structure is again one possible solution – the binary and ternary rhythmic alteration found in attested patterns is often derived by invoking a particular metrical structure. Most researchers propose that feet are maximally binary or ternary. If this is the case, then the possible intervals of iteration are constrained by the inventory of possible feet, which would eliminate patterns that stress any arbitrary n^{th} syllable. This again would require a limitation on the kinds of atoms that can be proposed – only functions that perform binary or ternary foot parsing may be draw from the pool of possible OSL atoms. Alternatively, word length may provide a simple explanation – as n grows, an increasing number of words become smaller than n, removing evidence of the iterating pattern from the lexicon.

OSL plus OSL

Compositions of OSL atoms can be used to describe patterns where stress iterates through the word twice. Such "bidirectional" patterns are rare, but do exist. Cahuilla (Seiler 1977) is an example. In Cahuilla, stress iterates left-to-right and right-to-left, starting at the root boundary on each pass:

$$
\begin{array}{lllll}
(55) & & \text{L-R} & & \text{R-L} \\
\sigma\sigma+\sigma\sigma & \mapsto & \sigma\sigma+\acute{\sigma}\sigma & \mapsto & \acute{\sigma}\sigma+\acute{\sigma}\sigma \\
\sigma\sigma\sigma+\sigma\sigma\sigma & \mapsto & \sigma\sigma\sigma+\acute{\sigma}\sigma\acute{\sigma} & \mapsto & \sigma\acute{\sigma}\sigma+\acute{\sigma}\sigma\acute{\sigma} \\
\sigma\sigma\sigma\sigma+\sigma\sigma\sigma\sigma & \mapsto & \sigma\sigma\sigma\sigma+\acute{\sigma}\sigma\acute{\sigma}\sigma & \mapsto & \acute{\sigma}\sigma\acute{\sigma}\sigma+\acute{\sigma}\sigma\acute{\sigma}\sigma
\end{array}
$$

Thus, OSL plus OSL compositions accurately predict the existence of bidirectional patterns like Cahuilla. As described above in §3, OSL plus OSL also allows pathological patterns such as sour grapes stress to be described. It is more difficult to argue that this is a typological accident – attested stress patterns are mostly consistent between words of various lengths in the language, and no known patterns rely on the parity of each word to determine placement of stress. In such cases, a closer examination of the functions in the composition can be undertaken. Sour grapes, for example, makes extensive use of "markup" – the alteration of elements in the string that were also changed by the first function. Attested patterns such as Cahuilla, on the other hand, do not display this behavior – the domains in which each function applies stress are disjoint, and so there can be no interaction. A statement targeting the interaction and markup-using properties of functions in OSL plus OSL compositions could thus provide the appropriate restriction, along the lines of research into restricted interactions of subsequential functions (Heinz and Lai 2013; Meinhardt et al. submitted).

4.4.3 SP atoms

Single atom

SP atoms enact long-distance generalizations that occur in stress typology. For example, in DTO or DTS patterns, a function that stresses the first heavy syllable encountered and no others encodes the "left/right-most" aspect of those patterns. Intuitively, this is long-distance because the function must remember that stress was previously applied to a heavy syllable for the entire duration of the word. This function was referred to as "stress H" in Chapter 3:

(56) LLLLHL $\mapsto$ LLLLH́L
HLLLHH $\mapsto$ H́LLLHH
LLHHLL $\mapsto$ LLH́HLL
HHHHHH $\mapsto$ H́HHHHH

Such atoms are an integral part of the compositional analyses proposed under this theory of stress, allowing for the description of long-distance patterns that is not possible with EO or OSL atoms alone. They are central to the analysis of DTO, DTS, and suprabinary patterns. SP also predicts some unattested patterns. For example, instead of enforcing no further stress after the first heavy, an SP function could require *all* heavies after the first to be stressed, in a kind of "stress harmony" generalization:

(57) LLLLHL $\mapsto$ LLLLH́L

HLLLHH $\mapsto$ H́LLLH́H́

LLHHLL $\mapsto$ LLH́H́LL

HHHHHH $\mapsto$ H́H́H́H́H́H́

All heavy syllables "harmonize" to the stressed status of the first one. This is simply the descriptive inverse of "stress H", having the same computational power. Some analyses of DTO and DTS patterns do assume that all heavies bear at least secondary stress (Baković (2004) for example). However, if secondary stress is absent, arguments from phonological substance can be mustered. A generalization that stresses all heavies clearly resembles a harmony generalization, and so the theoretical mechanisms that separate harmony patterns from stress patterns could be used to differentiate good and bad typological predictions when it comes to SP atoms in stress compositions. With only a single stress, a constraint banning atoms that create more than one foot in the word could be employed, for example.

SP plus SP

Compositions of SP atoms can describe DTS stress patterns. The first function stresses the first syllable and first heavy syllable, while the second function preserves the appropriate stress:

(58)		STRESS		DESTRESS
LLLLHL	$\mapsto$	ĹLLLH́L	$\mapsto$	LLLLH́L
HLLLHH	$\mapsto$	H́LLLHH	$\mapsto$	H́LLLHH
LLHHLL	$\mapsto$	ĹLH́HLL	$\mapsto$	LLH́HLL
HHHHHH	$\mapsto$	H́HHHHH	$\mapsto$	H́HHHHH

SP plus SP thus accurately predicts the existence of DTS stress patterns. Any typological issues that may be present with single SP atoms described above are duplicated in SP compositions. The first function could stress all heavies, and the second function stress all light syllables, for example:

(59)		ALL H		ALL L
LLLLHL	$\mapsto$	LLLLH́L	$\mapsto$	ĹĹĹĹH́Ĺ
HLLLHH	$\mapsto$	H́LLLH́H́	$\mapsto$	H́ĹĹĹH́H́
LLHHLL	$\mapsto$	LLH́H́LL	$\mapsto$	ĹĹH́H́ĹĹ
HHHHHH	$\mapsto$	H́H́H́H́H́H́	$\mapsto$	H́H́H́H́H́H́

Each function enforces stress "harmony" such that all heavy or light syllables follow the stress configuration of the first heavy or light. This leads to forms where the word is fully "harmonized" – all syllables bear stress. Just as with single SP functions, arguments from substantive phonological principles could be invoked to remove patterns such as this from the predicted typology. By stressing every syllable, there is no rhythmicity or contrast, and a look at the metrical structure would suggest that each syllable occupies an individual, unary foot. Arguments centered around these unusual aspects of the pattern could be used to avoid it when explanations purely from computation fail. In other words, incorporation of restrictions on the possible pool of atoms that can be fruitfully imposed on single SP functions will aid in the typological predictions of SP plus SP compositions as well.

4.4.4 OSL plus EO

As outlined above in §3, adopting OSL plus EO atoms allows for the description of patterns such as Pintupi without necessitating reference to properly subsequential expressivity. By eschewing subsequential atoms entirely, the theory presented here is able to avoid some issues with regards to overgeneration that they would introduce, such as the ability to explicitly track syllable parity. OSL plus EO compositions do inherit typological claims inherent to the EO class, such as the ability to stress all syllables in the edge window. An EO function could also invert the stress configuration in the EO window. For example, a pattern that combines left-to-right iteration with changing the stress of the final three syllables is describable with OSL plus EO:

(60)

		ITERATE		FINAL 3
$\sigma\sigma\sigma\sigma$	$\mapsto$	$\acute{\sigma}\sigma\acute{\sigma}\sigma$	$\mapsto$	$\acute{\sigma}\acute{\sigma}\sigma\acute{\sigma}$
$\sigma\sigma\sigma\sigma\sigma$	$\mapsto$	$\acute{\sigma}\sigma\acute{\sigma}\sigma\acute{\sigma}$	$\mapsto$	$\acute{\sigma}\sigma\sigma\acute{\sigma}\sigma$
$\sigma\sigma\sigma\sigma\sigma\sigma$	$\mapsto$	$\acute{\sigma}\sigma\acute{\sigma}\sigma\acute{\sigma}\sigma$	$\mapsto$	$\acute{\sigma}\sigma\acute{\sigma}\acute{\sigma}\sigma\acute{\sigma}$
$\sigma\sigma\sigma\sigma\sigma\sigma\sigma$	$\mapsto$	$\acute{\sigma}\sigma\acute{\sigma}\sigma\acute{\sigma}\sigma\acute{\sigma}$	$\mapsto$	$\acute{\sigma}\sigma\acute{\sigma}\sigma\sigma\acute{\sigma}\sigma$

While inversion of the stress pattern in any arbitrary edge window allows for overgeneration of unattested patterns, such deviations from the otherwise expected iteration pattern do occur in "switch languages" (Houghton 2013), where the final stress retracts from a word edge. The existence of such patterns is explained in part by the presence of foot structure, demonstrating that once again, substantive principles can be postulated to reign in the behavior of EO functions, whether they occur alone or are embedded in a compositional analysis. Incorporating a step of foot creation into the grammar would allow us to differentiate between a principled retraction of a single stress as part of a non-finality requirement and the ability to invert stress in an edge window of any size.

4.4.5 EO plus SP

Compositions of EO and SP atoms can accurately describe DTO stress patterns. This is achieved by the EO function marking the default stress position, while the SP function either stresses the first heavy syllable and removes the default or – preserves the default it is when appropriate:

(61)

	DEFAULT		STRESS	
LLLLHL	$\mapsto$	ĹLLLHL	$\mapsto$	LLLLH́L
HLLLHH	$\mapsto$	H́LLLHH	$\mapsto$	H́LLLHH
LLHHLL	$\mapsto$	ĹLHHLL	$\mapsto$	LLH́HLL
HHHHHH	$\mapsto$	H́HHHHH	$\mapsto$	H́HHHHH

Thus, inclusion of EO and SP atoms in the theory accurately predicts the existence of DTO stress patterns, without reference to full subsequential power. Just as with OSL plus EO, issues endemic to both the EO and SP classes present areas where the theory overgenerates with regards to EO plus SP. A pattern that marked the default but then "harmonized" all heavy syllables stress would be EO plus SP, for example. So would a pattern that marked the first four syllables as stressed, then applied the SP function. The same substantive requirements that may be proposed to restrict EO and SP individually, such as metrical structure, will similarly reign in the behavior of EO plus SP compositions.

4.4.6 SP plus OSL

Though no patterns of which I am aware exhibit a generalization matching an SP plus OSL composition, inclusion of both types of atoms in the theory is a tacit hypothesis that such patterns could exist, and so I examine some possible patterns here. One possibility is a pattern where the initial SP function first locates a heavy syllable, and the following OSL function iterates stress from this point:

(62)

	STRESS H		ITERATE	
LLLLHL	↦	LLLLH́L	↦	ĹLĹLH́L
HLLLHH	↦	HLLLHH́	↦	HĹLĹHH́
LLHHLL	↦	LLHH́LL	↦	LĹHH́LL
HHHHHH	↦	HHHHHH́	↦	HH́HH́HH́

This hypothetical pattern bears some resemblance to that of Iñapari (Parker 1999), where stress iterates in both directions away from a quantity-sensitive stress. However, the basis of iteration in Iñapari occurs maximally four syllables from the right word edge, and so it need not employ a long distance function. Nevertheless, the pattern in (62) does not appear pathological in the sense that sour grapes or “stress odd heavy” does. It also resembles known stress patterns, unlike many other unattested examples discussed in this section, and so I leave its presence or absence in the typology as a topic for future research. Other patterns permitted by SP plus OSL compositions inherit issues with SP generally, and so are less likely to appear as attested members of stress typology. For example, the SP function in a composition could stress all heavy syllables, while the OSL function iterates stress normally from right to left, creating a tangled web of stresses:

(63)

	ALL H		ITERATE	
LLLLHL	↦	LLLLH́L	↦	LĹLĹH́Ĺ
HLLLHH	↦	H́LLLH́H́	↦	H́ĹLĹH́H́
LLHHLL	↦	LLH́H́LL	↦	LĹH́H́LĹ
HHHHHH	↦	H́H́H́H́H́H́	↦	H́H́H́H́H́H́

Once again, whatever principles can be invoked to prevent individual SP functions from creating these harmony-like generalizations, such as foot structure and constraints against certain atoms, will also have the effect of stopping SP functions from contributing them to stress compositions as well.

4.4.7 Summary

The previous section provided a summary of the atomic functions permitted by this theory of stress and their consequences for stress typology, both individually and as part of a compositional analysis. Note that this overview was not meant to be exhaustive – there are other ways in which these atoms and their compositions make both good and bad predictions. The goal instead was to demonstrate the kinds of generalizations that are possible given the theory of stress that this dissertation proposes. Additionally, while all atoms and their combinations overgenerate to some extent, I have argued that this overgeneration may be constrained by substantive aspects of phonological grammar such as foot structure, culminativity, and other mechanisms. While these theoretical devices add great value to our overall understanding of what a possible stress pattern is, I instead focus on the computational complexity of the atomic primitives of stress and the direct effect of this complexity on the predicted typology. It is also worth noting that other theories of stress, whether parametric or OT based, will suffer from overgeneration-related issues for precisely the same reasons as this theory. Without stating further restrictions, there is no *a priori* reason that any logically possible rule or constraint cannot be proposed. The computational perspective makes some progress here by delineating the space from which stress primitives may be drawn, which – while still open to overgeneration – does carve out a well-defined, more restrictive space than a theory that does not enforce such requirements.

Chapter 5: Local Stress Patterns

5.1 Introduction

This chapter provides a restrictive theory of unidirectional iterative stress from a computational perspective. I argue that the best characterization of iterative stress is one where the "atoms" of the stress generalization are composed in sequence. The atoms are defined as functions that implement basic aspects of stress such as basic iteration or non-finality. The paper thus joins a vast body of previous work treating surface stress patterns as the sum of individual stress generalizations (Chomsky and Halle 1968; Booij 1983; Halle and Vergnaud 1987; Idsardi 1992; Hayes 1995; Bailey 1995; Hyde 2002; Gordon 2002; Kager 2005; Buckley 2009; Kager 2012; Rogers et al. 2013; Heinz 2014), but addresses the question of what a possible stress pattern can be through the lens of computation, analyzing these individual stress functions with the tools of formal language theory (FLT). The iteration of stress is shown to be an output strictly local (OSL) function (Chandlee et al. 2015; Chandlee and Heinz 2018), while phonological requirements such as non-finality, clash, and lapse are encoded with "edge-oriented" (EO) functions as defined in Chapter 4. EO functions are limited to apply only at or near a word edge, thus representing a novel characterization of a typologically real property of stress patterns using computational methods. The proposal then is that the observed output of an iterative stress map is the composition of the local functions implementing the relevant stress primitives. It is demonstrated that the restriction to OSL plus EO compositions constraints the hypothesized typology for iterative stress in a way that more permissive hypotheses do not, while still accounting for a wide range of attested iterative patterns.

This proposal provides a better hypothesis for the typology of iterative stress than using a single function in part because, for some patterns, this single function is properly subsequential. However, I demonstrate that these iterative stress patterns *are* local, containing just a small set of atoms that each express a local generalization. Though compositions of local functions can in general describe non-local patterns, the restrictions invoked here ensure that a well-defined notion of locality is preserved. Full subsequential power also overgenerates by predicting pathological parity-counting stress patterns that are non-phonological. Thus, this chapter connects the FLT complexity classes with substantive elements of phonological theory – though one subsequential function does suffice, it masks the true computational nature of the individual stress atoms such as clash, lapse, and the basic iteration of stress. The result is a restrictive theory that makes explicit the computational requirements for iterative stress while embracing its individual substantive elements that have been noted in the literature for decades. It also suggests the separation of stress typology into local and non-local halves based on the computational properties that attested patterns have, as the different properties lead to different typological predictions.

By examining both *quantity insensitive* (QI) and *quantity sensitive* (QS) stress, I show that – despite surface differences – they share fundamental computational properties, including that of iteration of stress by an OSL function. While minor differences appear in the analyses to account for surface patterns, all iterative QI and QS patterns can be captured with this decomposed OSL plus EO analysis.

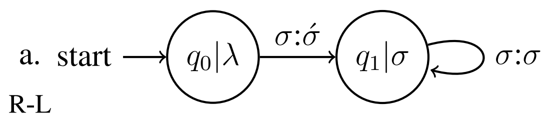
The content of the chapter also directly bears on issues of function composition. The OSL function iterates stress through a string. Where this fails to capture the correct input-output map with regard to the specific language, a following EO function provides "cleanup" for the OSL function. In intuitive terms, EO functions are limited to apply only within a fixed distance from a word edge, and so make a limited number of alterations to the string

– the cleanup provided is always local. Limitation to EO prevents the abuse of markup, where the first function leaves information that the second function can exploit. Avoiding this constrains overgeneration and, I argue, is a property of bounded stress systems in general.

5.2 Analyses

To begin, consider that any bounded non-iterative pattern can be described with a single EO function. For example, a final stress function can be represented with the following transducer:

(64)



This function is EO, as the only substring leading to a non-identity transition is λ - after one syllable is read, no further changes are made to the string, as indicated by the label on q_1. All bounded non-iterative patterns, QI or QS, are EO because they are subject to similar computational requirements. Peninitial stress is placed after the substring $\#\sigma$ is read, for example. Latin stress is placed correctly after a maximum of three input symbols are read – $L\#, H\#, LL\#,$ or $LH\#.$ That any single stress pattern is determined entirely by some small window of syllables at a word edge is precisely the property that makes them describable with an EO function.

Iterative patterns rely on different classes of functions. As described in Chapter 3, the OSL functions are functions that rely on information in the output string. This makes the OSL class a good hypothesis for iterative functions in general, including iterative stress, as application of successive stresses depends on the presence of another stress somewhere in the output. However, plain iteration of stress is not the only factor at play in iterative stress assignment. The phenomena of clash, lapse, or non-finality disrupt iteration of stress. In QS systems, a heavy syllable may disrupt the count for application of stress. Below I show that, despite surface differences, these types of patterns share fundamental computational properties, including that of OSL iteration of stress. This is true of both QI patterns and QS patterns, suggesting that despite a small difference in alphabet size resulting in surface differences, the core computational properties are the same.

5.2.1 Iteration of stress

Binary iteration

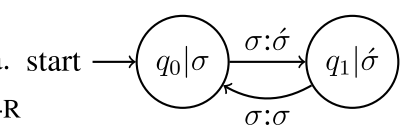
In some languages, binary iteration is the only feature of stress assignment. This means the stress function in these languages is a simple left or right OSL function. Murinbata (Street and Mollinjin 1981), repeated here, was given as a QI example in the previous chapter. Murinbata stresses every odd syllable in the string:

(65) $\acute{\sigma}\sigma, \acute{\sigma}\sigma\acute{\sigma}, \acute{\sigma}\sigma\acute{\sigma}\sigma, \acute{\sigma}\sigma\acute{\sigma}\sigma\acute{\sigma}, \acute{\sigma}\sigma\acute{\sigma}\sigma\acute{\sigma}\sigma, ...$

As described above, a function describing this mapping needs only keep track of what the previous output symbol was. If the previous output was a stressed syllable, then the current output will be unstressed, and vice versa. The Murinbata transducer is repeated here:

(66)

a. L-R



Reading left to right, the first input syllable is output as stressed, moving to q_1. The next syllable is output with no stress and we return to q_0, where the process repeats until the end of the word. The information encoded by each state makes it clear that the process is output-oriented. q_1 encodes a stressed syllable having been written in the output, while q_0 does the same for an unstressed syllable in the output. The input string contains no such information that could be used to iteratively place stress. The function is OSL. The following derivations show how stress is applied to the string in five and six-syllable words:

(67)

5σ						
ITERATION:						
input:	σ	σ	σ	σ	σ	
states:	$q_0 \rightarrow q_1 \rightarrow q_0 \rightarrow q_1 \rightarrow q_0 \rightarrow q_1$					
output:	$\acute{\sigma}$	σ	$\acute{\sigma}$	σ	$\acute{\sigma}$	

6σ						
input:	σ	σ	σ	σ	σ	σ
states:	$q_0 \rightarrow q_1 \rightarrow q_0 \rightarrow q_1 \rightarrow q_0 \rightarrow q_1 \rightarrow q_0$					
output:	$\acute{\sigma}$	σ	$\acute{\sigma}$	σ	$\acute{\sigma}$	σ

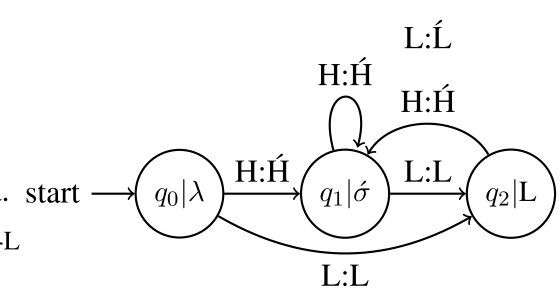
The same generalization can be seen in QS languages as well. Fijian stress (Hayes 1995) displays binary iteration with some extra caveats related to the light/heavy syllable distinction. Despite these additional requirements, the function is still OSL, calculated based on previous outputs, just as in Murinbata.

Main stress falls on a heavy final syllable, or if absent, the penult. Then, additional stress is placed on all heavy syllables and on every other light syllable counting leftward from the previous stress. This means that a heavy resets the count for iteration of stress, as seen in (g) and (h):

(68)
a. ĹL b. LĹL
c. LH́ d. H́H́
e. LĹLĹL f. ĹLH́
g. LH́ĹL h. ĹLH́ĹL

If placement of stress were not sensitive to syllable weight, we would expect an initial stress in (g), and a peninitial stress in (h). Despite the sensitivity to weight, the stress function in Fijian is still OSL, as placement of stress once again depends on previous stressed outputs:

(69)

a. 

R-L

Reading right-to-left, whenever an H is seen, it is output as stressed, landing in q_1. Whenever an L is seen, an unstressed L is output, landing in q_2. From here, the next syllable will bear stress, regardless of whether it is heavy or light, bringing us back to q_1. Thus, q_1 represents having just written a stress in the output, resetting the binary count. An example derivation for LLLLL and LLHLL are given here:

(70)

input:	L	L	L	L	L		input:	L	L	H	L	L
states:	$q_2 \leftarrow$	$q_1 \leftarrow$	$q_2 \leftarrow$	$q_1 \leftarrow$	$q_2 \leftarrow$	q_0	states:	$q_1 \leftarrow$	$q_2 \leftarrow$	$q_1 \leftarrow$	$q_1 \leftarrow$	$q_2 \leftarrow q_0$
output:	L	Ĺ	L	Ĺ	L		output:	Ĺ	L	H́	Ĺ	L

Note the similarity to Murinbata – q_1 encodes the same information in both (66) and (69), despite the fact that Murinbata is QI and Fijian is QS. q_2 of Fijian is analogous to q_0 of Murinbata, the difference being that only light syllables can be output with no stress in Fijian. The R-OSL function in (69) correctly describes the stress mapping in Fijian.

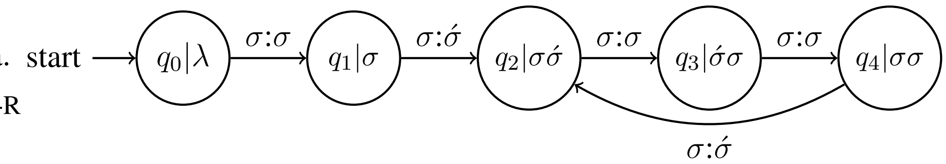
Ternary iteration

Some languages employ a ternary count for iteration of stress. When this is the only factor, stress assignment is OSL. A QI example is Ioway-Oto (Whitman 1947; Gordon 2002), which stresses every third syllable from the peninitial counting to the right:

(71) $\sigma\acute{\sigma}, \sigma\acute{\sigma}\sigma, \sigma\acute{\sigma}\sigma\sigma, \sigma\acute{\sigma}\sigma\sigma\acute{\sigma}, \sigma\acute{\sigma}\sigma\sigma\acute{\sigma}\sigma, \sigma\acute{\sigma}\sigma\sigma\acute{\sigma}\sigma\sigma, \sigma\acute{\sigma}\sigma\sigma\acute{\sigma}\sigma\sigma\acute{\sigma}...$

The stress function for Ioway-Oto is like Murinbata, but with a slight delay. It outputs a stressed syllable and two unstressed syllable before the iteration of stress resumes. The following transducer represents the Ioway-Oto stress function. Note that we encounter a case of the "short word problem" here – accounting for the monosyllabic form would require a waiting transition with the empty string λ, which by definition would make the function non-OSL. This is true of every QI pattern where main stress is not anchored directly on the left or right word edge. A solution to this issue is beyond the scope of this thesis, and so I set it aside for now:

(72) a.



L-R

The transducer represents a TERNARY ITERATION function. Reading left to right, the peninitial receives stress in the transition from q_1 to q_2. The next two syllables are output as unstressed, moving to q_3 and q_4. From here, a stressed syllable is output in the return to q_2 from q_4. This ternary cycle continues for the length of the word. This captures the ternary iteration of Ioway-Oto and indicates that, just like the binary iteration of Murinbata, it is OSL:

(73) a. $\sigma\sigma\sigma\sigma\sigma\sigma\sigma \mapsto \sigma\acute{\sigma}\sigma\sigma\acute{\sigma}\sigma\sigma$

b. $\sigma\sigma\sigma\sigma\sigma\sigma\sigma\sigma \mapsto \sigma\acute{\sigma}\sigma\sigma\acute{\sigma}\sigma\sigma\acute{\sigma}$

c. $\sigma\sigma\sigma\sigma\sigma\sigma\sigma\sigma\sigma \mapsto \sigma\acute{\sigma}\sigma\sigma\acute{\sigma}\sigma\sigma\acute{\sigma}\sigma$

An example derivation for an eight syllable input is given here:

(74) 8σ

TERNARY ITERATION:

input:	σ	σ	σ	σ	σ	σ	σ	σ
states:	$q_0 \rightarrow q_1$	$\rightarrow q_2$	$\rightarrow q_3$	$\rightarrow q_4$	$\rightarrow q_2$	$\rightarrow q_3$	$\rightarrow q_4$	$\rightarrow q_2$
output:	σ	$\acute{\sigma}$	σ	σ	$\acute{\sigma}$	σ	σ	$\acute{\sigma}$

Ternary iteration of stress occurs in QS systems as well. As with binary iteration, despite descriptive differences, the input-output map remains an OSL function. One example is Sentani (Cowan 1965; Hayes 1995). In Sentani, the final syllable is stressed if it is heavy. Otherwise, the penult is stressed instead (a-b). From here, ternary iteration proceeds to the left (c-d). Iteration is interrupted by a heavy syllable (e-f), but not in the case that this would create a stress clash (g-h):

(75) a. LH́ b. H́L

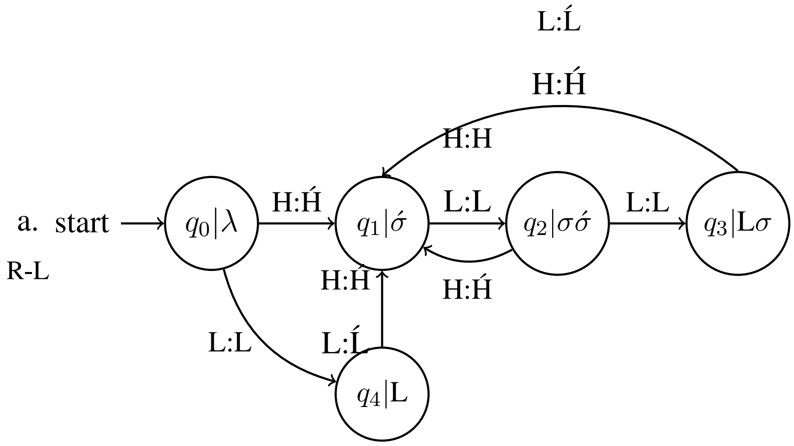
c. ĹLLĹL d. LLĹLLH́

e. LH́LĹL f. LH́LH́L

g. HĹL h. LH́LHH́L

As with Fijian, the stress function for Sentani is OSL, despite the additional requirements related to weight and clash. Iteration of stress is output-oriented, seen in the transducer as a return to q_1 whenever stress is applied. No such generalization about the input string is possible:

(76) a.



Reading right to left, a final heavy syllable is stressed in the transition from q_0 to q_1, or the penult is stressed in the path from q_4 to q_1. From here, ternary iteration of stress occurs in the transitions between states q_1, q_2 and q_3, with q_1 representing having just seen a stress in the *output* string. The transition from q_2 to q_1 represents the reset of iteration by a heavy syllable. Clash is disallowed, as the transition from q_1 (having just output a stress) to q_2 only produces unstressed syllables. Reaching q_3 indicates that an unstressed syllable and an L were just output, and so the following symbol is stressed regardless of weight, returning again to q_1. The stress map for Sentani is R-OSL:

(77) a. LH $\mapsto$ LH́

b. LLLLL $\mapsto$ ĹLLĹL

c. LHLLL $\mapsto$ LH́LĹL

d. LHLHHL $\mapsto$ LH́LHH́L

An example derivation for LLLLL and LHLHHL are given here:

(78) QS TERNARY ITERATION:

input:	L	L	L	L	L	input:	L	H	L	H	H	L
states:	$q_1 \leftarrow q_3 \leftarrow q_2 \leftarrow q_1 \leftarrow q_4 \leftarrow q_0$					states:	$q_2 \leftarrow q_1 \leftarrow q_3 \leftarrow q_2 \leftarrow q_1 \leftarrow q_4 \leftarrow q_0$					
output:	Ĺ	L	L	Ĺ	L	output:	L	H́	L	H	H́	L

The analysis undertaken in this section indicates that, in general, iteration of stress is an OSL function, whether it is binary or ternary, or whether the specific case is QI or QS. Indeed, all patterns analyzed below require some version of (66) or (69) to model iterative application of stress. While this alone is sufficient for languages like Murinbata and Fijian, other types of patterns will require additional machinery – additional application of a function corresponding to the atomic elements of the stress pattern such as non-finality, lapse, and clash.

5.2.2 Non-finality

In non-finality languages, iteration of stress is disrupted by a requirement that the final syllable surface as unstressed. A QI example comes from Pintupi (Hansen and Hansen 1969), as discussed in Chapter 3:

(79) $\acute{\sigma}\sigma, \acute{\sigma}\sigma\sigma, \acute{\sigma}\sigma\acute{\sigma}\sigma, \acute{\sigma}\sigma\acute{\sigma}\sigma\sigma, \acute{\sigma}\sigma\acute{\sigma}\sigma\acute{\sigma}\sigma, \ldots$

The ITERATION function alone does not capture the Pintupi pattern. Unlike Murinbata, the pattern requires a small amount of lookahead that is not available to OSL functions. Rather, a transducer for Pintupi as a single function is properly subsequential, as information other than the most recent output or input is required to compute the function.

Though this suggests that iteration of stress in some languages is more complex than in other languages, it was demonstrated that separating the limited lookahead from the iteration of stress by dividing the map into two different functions reveals underlying similarities among iterative patterns and provides a more restrictive theory of stress than just assuming stress is any subsequential function. It preserves the formal notion of locality present in OSL and EO functions but absent in properly subsequential functions. Both Pintupi and Murinbata share the OSL ITERATION function, while Pintupi requires an additional cleanup step that acts in place of lookahead to derive the correct output. This comes in the form of the EO function that removes a final stress if present, NON-FINALITY, as seen in §4 of the previous chapter.

A parallel, QS version of non-finality appears in Wergaia (Hercus 1986; Hyde 2011b). In this language, stress iterates from left to right starting with the first syllable (a-d), avoiding the final syllable only in the case that the final syllable is light (e-f):

(80) a. ĹL

b. H́L

c. ĹLĹL

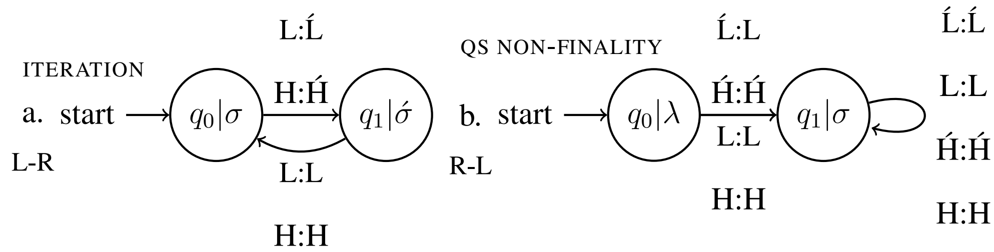
d. ĹLH́

e. H́LL

f. ĹHL

As seen in (e) and (f), iteration of stress avoids a final syllable if it is light. Conversely, if the final syllable is heavy as in (d), iteration proceeds as normal. Though this binary weight distinction for non-finality sets Wergaia apart from Pintupi in a descriptive sense, the computation of the stress map is of equal complexity. This is because the decision on whether to output an odd-numbered *light* syllable as stressed depends on if the current position is the end of the word. Like the single-function map for Pintupi, this is not OSL – it is properly subsequential as a single-function map because of the waiting. It is also amenable to the same compositional analysis as Pintupi, separating the iteration of stress from the non-finality requirement:

(81)



The L-OSL transducer in (a) is an ITERATION function, placing stress on every odd-numbered syllable regardless of weight. This means that, despite the different input and output symbols, the application of the function is identical to the ITERATION function of the QI patterns above. The transducer in (b) is QS NON-FINALITY. Just like the NON-FINALITY function of Pintupi, the only input substring leading to a non-identity transition is λ, and so it is EO. Working right to left, taking the output of the first function as its input, it removes stress from final light syllables, otherwise outputting the word faithfully and modeling the weight-specific non-finality requirement of Wergaia. This highlights the fact that, despite the different alphabets, the information encoded in the states is identical to that of Pintupi, reinforcing the fact that the computation of both patterns is the same.

(82)

		(81a)		(81b)
a. LLLL	↦	ĹLĹL	↦	ĹLĹL
b. LLH	↦	ĹLH́	↦	ĹLH́
c. HLL	↦	H́LĹ	↦	H́LL

A derivation for inputs LLLLL and LLLLH are given here:

(83) ITERATION:

input:	L	L	L	L	L	
states:	$q_0 \rightarrow$	$q_1 \rightarrow$	$q_0 \rightarrow$	$q_1 \rightarrow$	$q_0 \rightarrow$	q_1
output:	Ĺ	L	Ĺ	L	Ĺ	

input:	L	L	L	L	H	
states:	$q_0 \rightarrow$	$q_1 \rightarrow$	$q_0 \rightarrow$	$q_1 \rightarrow$	$q_0 \rightarrow$	q_1
output:	Ĺ	H	Ĺ	L	H́	

QS NON-FINALITY:

input:	Ĺ	L	Ĺ	L	Ĺ	input:	Ĺ	H	Ĺ	L	H́
states:	$q_1 \leftarrow q_1 \leftarrow q_1 \leftarrow q_1 \leftarrow q_1 \leftarrow q_0$					states:	$q_1 \leftarrow q_1 \leftarrow q_1 \leftarrow q_1 \leftarrow q_1 \leftarrow q_0$				
output:	Ĺ	L	Ĺ	L	L	output:	Ĺ	H	Ĺ	L	H́

Despite the differences in the surface generalizations of Pintupi and Wergaia, the fundamental properties underlying the calculation of stress in both languages are identical. This kind of compositional analysis makes the computational properties of different stress phenomena explicit — iteration of stress is OSL and non-finality is EO, regardless of weight-sensitivity.

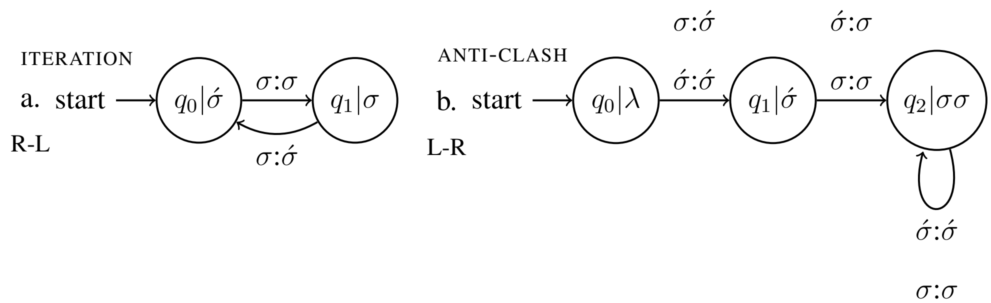
5.2.3 Internal lapse

Languages that display an internal lapse can also be analyzed compositionally in this way. Quantity insensitive examples include Garawa (Furby 1974) and Piro (Matteson 1965). To my knowledge, there are no iterative QS internal lapse counterparts. The pattern of Garawa is as follows:

(84) $\acute{\sigma}\sigma, \acute{\sigma}\sigma\sigma, \acute{\sigma}\sigma\acute{\sigma}\sigma, \acute{\sigma}\sigma\sigma\acute{\sigma}\sigma, \acute{\sigma}\sigma\acute{\sigma}\sigma\acute{\sigma}\sigma, \acute{\sigma}\sigma\sigma\acute{\sigma}\sigma\acute{\sigma}\sigma...$

The initial syllable is always stressed. Stress iterates right to left from the penultimate, but avoids the peninitial when placing a stress there would create a clash. Since ITERATION in this case must apply R-L, the OSL function will fail to stress the initial syllable and erroneously stress the peninitial syllable in odd-parity forms. This is again due to the lack of lookahead in the OSL function – a fully subsequential function could provide the necessary lookahead, but an OSL function cannot. The EO function then must place an initial stress where absent and resolve clashes that this would produce. The functions are represented via the following transducers:

(85)



The R-OSL ITERATION transducer in (a) iterates stress right to left starting with the penult. Taking the output of the first function as its input and reading the string left to right, ANTI-CLASH adds an initial stress if it is absent but otherwise leaves the string unchanged. It then resolves any potential clashes this would create in the transition from q_1 to q_2. For Garawa, this clash resolution results in the characteristic internal lapse in odd parity forms. The rest of the string is left unchanged. ANTI-CLASH is EO, as $\#$ and $\#\acute{\sigma}$ are the only subsequences leading to a state where a change can be applied. The following examples demonstrate the effect of the composed mapping:

		(85a)		(85b)	
(86)	a. $\sigma\sigma\sigma\sigma\sigma\sigma$	$\mapsto$	$\acute{\sigma}\sigma\acute{\sigma}\sigma\acute{\sigma}\sigma$	$\mapsto$	$\acute{\sigma}\sigma\acute{\sigma}\sigma\acute{\sigma}\sigma$
	b. $\sigma\sigma\sigma\sigma\sigma\sigma\sigma$	$\mapsto$	$\sigma\acute{\sigma}\sigma\acute{\sigma}\sigma\acute{\sigma}\sigma$	$\mapsto$	$\acute{\sigma}\sigma\sigma\acute{\sigma}\sigma\acute{\sigma}\sigma$

Even-parity forms (86a) are unchanged by the EO function—the action of the OSL function alone provides the correct output. Odd-parity forms (86b) have the first two syllables altered by the EO function, creating an internal lapse. The generalization in Garawa matches that of other iterative patterns – it adheres to the single-OSL restriction with EO cleanup function acting in place of the lookahead seen in the subsequential transducers. This indicates that along with non-finality, the forces that create internal lapse are also EO when viewed as atomic stress elements. A derivation for a five and six syllable word are given here.

(87) 5σ / 6σ

ITERATION:

input:		σ		σ		σ		σ		σ		input:		σ		σ		σ		σ		σ		σ		
states:	q_1	$\leftarrow$	q_0	$\leftarrow$	q_1	$\leftarrow$	q_0	$\leftarrow$	q_1	$\leftarrow$	q_0	states:	q_0	$\leftarrow$	q_1	$\leftarrow$	q_0	$\leftarrow$	q_1	$\leftarrow$	q_0	$\leftarrow$	q_1	$\leftarrow$	q_0	
output:		σ		$\acute{\sigma}$		σ		$\acute{\sigma}$		σ		output:		$\acute{\sigma}$		σ		$\acute{\sigma}$		σ		$\acute{\sigma}$		σ		

ANTI-CLASH:

input:		σ		$\acute{\sigma}$		σ		$\acute{\sigma}$		σ		input:		$\acute{\sigma}$		σ		$\acute{\sigma}$		σ		$\acute{\sigma}$		σ		
states:	q_0	$\rightarrow$	q_1	$\rightarrow$	q_2	$\rightarrow$	q_2	$\rightarrow$	q_2	$\rightarrow$	q_2	states:	q_0	$\rightarrow$	q_1	$\rightarrow$	q_2	$\rightarrow$	q_2	$\rightarrow$	q_2	$\rightarrow$	q_2	$\rightarrow$	q_2	
output:		$\acute{\sigma}$		σ		σ		$\acute{\sigma}$		σ		output:		$\acute{\sigma}$		σ		$\acute{\sigma}$		σ		$\acute{\sigma}$		σ		

5.2.4 Clash

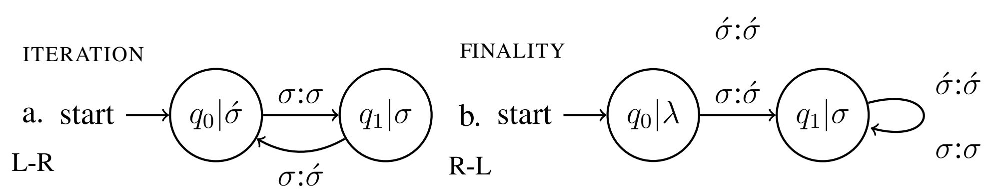
Thus far, the patterns examined have explicitly avoided placing two stresses next to each other. Another category of iterative pattern does the opposite, allowing adjacent stresses. A QI example comes from Ojibwe (Kaye 1973), shown here:

(88) $\sigma\acute{\sigma}, \sigma\acute{\sigma}\acute{\sigma}, \sigma\acute{\sigma}\sigma\acute{\sigma}, \sigma\acute{\sigma}\sigma\acute{\sigma}\acute{\sigma}, \sigma\acute{\sigma}\sigma\acute{\sigma}\sigma\acute{\sigma}, \ldots$

Stress iterates L-R from the peninitial syllable and the final syllable is always stressed, even if this would create a clash. In odd-parity forms, binary iteration will not add a final stress, and so an EO function does the job instead.

Like Pintupi, this pattern requires bounded lookahead. The decision to stress the third syllable, for example, cannot be made until the next symbol is read. Whereas Pintupi must avoid stressing the final syllable, Ojibwe is required to do so. Transducers describing the functions are given here:

(89)



The L-OSL function ITERATION in (89a) iterates stress starting from the peninitial. This is identical to the Garawa transducer, but operating in the opposite direction. The function (89b), dubbed FINALITY adds a final stress if one is absent, but otherwise leaves the string unchanged. This is the inverse of the function NON-FINALITY function seen in Pintupi, which deletes a final stress. Just like NON-FINALITY, the only input substring leading to a non-identity transition is the word boundary #, and so FINALITY is also EO. Examples are given here:

(90)

		(89a)		(89b)
a. $\sigma\sigma\sigma\sigma\sigma\sigma$	$\mapsto$	$\sigma\acute{\sigma}\sigma\acute{\sigma}\sigma\acute{\sigma}$	$\mapsto$	$\sigma\acute{\sigma}\sigma\acute{\sigma}\sigma\acute{\sigma}$
b. $\sigma\sigma\sigma\sigma\sigma\sigma\sigma$	$\mapsto$	$\sigma\acute{\sigma}\sigma\acute{\sigma}\sigma\acute{\sigma}\sigma$	$\mapsto$	$\sigma\acute{\sigma}\sigma\acute{\sigma}\sigma\acute{\sigma}\acute{\sigma}$

Once again, the EO function demonstrates that the lookahead required to describe the pattern as a single subsequential function is unnecessary in a compositional analysis. Above it was shown that non-finality and lapse creation are EO. Now clash can be added to this group of EO stress functions. An example derivation for a five and six syllable word are given here:

(91) 5σ

ITERATION:

input: $\sigma \quad \sigma \quad \sigma \quad \sigma \quad \sigma$

states: $q_0 \rightarrow q_1 \rightarrow q_0 \rightarrow q_1 \rightarrow q_0 \rightarrow q_1$

output: $\sigma \quad \acute{\sigma} \quad \sigma \quad \acute{\sigma} \quad \sigma$

FINALITY:

input: $\sigma \quad \acute{\sigma} \quad \sigma \quad \acute{\sigma} \quad \sigma$

states: $q_1 \leftarrow q_1 \leftarrow q_1 \leftarrow q_1 \leftarrow q_1 \leftarrow q_0$

output: $\sigma \quad \acute{\sigma} \quad \sigma \quad \acute{\sigma} \quad \acute{\sigma}$

6σ

ITERATION:

input: $\sigma \quad \sigma \quad \sigma \quad \sigma \quad \sigma \quad \sigma$

states: $q_0 \rightarrow q_1 \rightarrow q_0 \rightarrow q_1 \rightarrow q_0 \rightarrow q_1 \rightarrow q_0$

output: $\sigma \quad \acute{\sigma} \quad \sigma \quad \acute{\sigma} \quad \sigma \quad \acute{\sigma}$

FINALITY:

input: $\sigma \quad \acute{\sigma} \quad \sigma \quad \acute{\sigma} \quad \sigma \quad \acute{\sigma}$

states: $q_1 \leftarrow q_1 \leftarrow q_1 \leftarrow q_1 \leftarrow q_1 \leftarrow q_1 \leftarrow q_0$

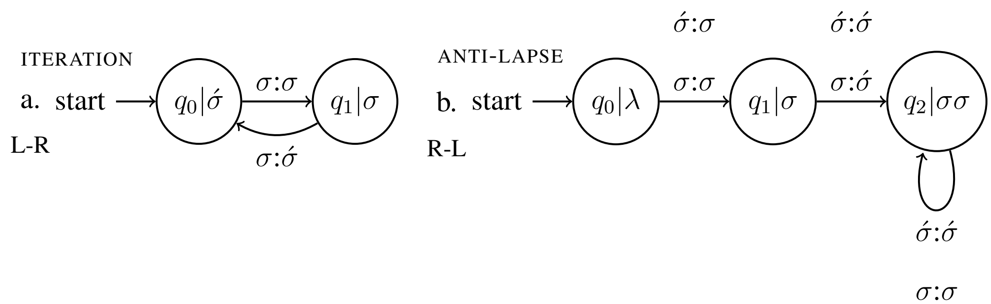
output: $\sigma \quad \acute{\sigma} \quad \sigma \quad \acute{\sigma} \quad \sigma \quad \acute{\sigma}$

Another type of pattern displays what could be called internal clash, mirroring the internal lapse languages. An example comes from Southern Paiute (Harms 1966):

(92) $\acute{\sigma}\sigma, \sigma\acute{\sigma}\sigma, \sigma\acute{\sigma}\acute{\sigma}\sigma, \sigma\acute{\sigma}\sigma\acute{\sigma}\sigma, \sigma\acute{\sigma}\sigma\acute{\sigma}\acute{\sigma}\sigma, \sigma\acute{\sigma}\sigma\acute{\sigma}\sigma\acute{\sigma}\sigma...$

The penult is always stressed. Stress iterates from the peninitial syllable, but unlike Garawa, any resulting clashes are allowed to surface. In even parity forms, the L-OSL function ITERATION skips the penult and stresses the final syllable. Knowing when to stress a given odd syllable or not would require a bounded amount of lookahead that the OSL function cannot provide. Instead, an EO function reverses these errors, providing cleanup and deriving the correct surface form. Transducers for the functions are as follows:

(93)



The OSL function in (a) is identical to Ojibwe, stressing every even syllable starting from the left. The function in (b) is an ANTI-LAPSE function that removes final stress and stresses the penult if it is unstressed, leaving the rest of the string unchanged. This is an EO function, as the only input substring leading to a possible changes are # and #σ. ANTI-LAPSE introduces a clash in even parity forms. In this sense, patterns like Paiute combine a non-finality generalization with a clash generalization. This compositional analysis makes the individual atomic stress properties and their complexity explicit. Paiute is the inverse of Garawa, which instead adds an initial stress and avoids clash. Examples are given here:

(94)

	(93a)		(93b)	
a. $\sigma\sigma\sigma\sigma\sigma\sigma$	$\mapsto$	$\sigma\acute{\sigma}\sigma\acute{\sigma}\sigma\acute{\sigma}$	$\mapsto$	$\sigma\acute{\sigma}\sigma\acute{\sigma}\acute{\sigma}\sigma$
b. $\sigma\sigma\sigma\sigma\sigma\sigma\sigma$	$\mapsto$	$\sigma\acute{\sigma}\sigma\acute{\sigma}\sigma\acute{\sigma}\sigma$	$\mapsto$	$\sigma\acute{\sigma}\sigma\acute{\sigma}\sigma\acute{\sigma}\sigma$

A derivation for a five and six syllable word are given here.

(95) 5σ $\qquad$ 6σ

ITERATION:

	5σ		6σ
input:	$\sigma \quad \sigma \quad \sigma \quad \sigma \quad \sigma$	input:	$\sigma \quad \sigma \quad \sigma \quad \sigma \quad \sigma \quad \sigma$
states:	$q_0 \rightarrow q_1 \rightarrow q_0 \rightarrow q_1 \rightarrow q_0 \rightarrow q_1$	states:	$q_0 \rightarrow q_1 \rightarrow q_0 \rightarrow q_1 \rightarrow q_0 \rightarrow q_1 \rightarrow q_0$
output:	$\sigma \quad \acute{\sigma} \quad \sigma \quad \acute{\sigma} \quad \sigma$	output:	$\sigma \quad \acute{\sigma} \quad \sigma \quad \acute{\sigma} \quad \sigma \quad \acute{\sigma}$

ANTI-LAPSE:

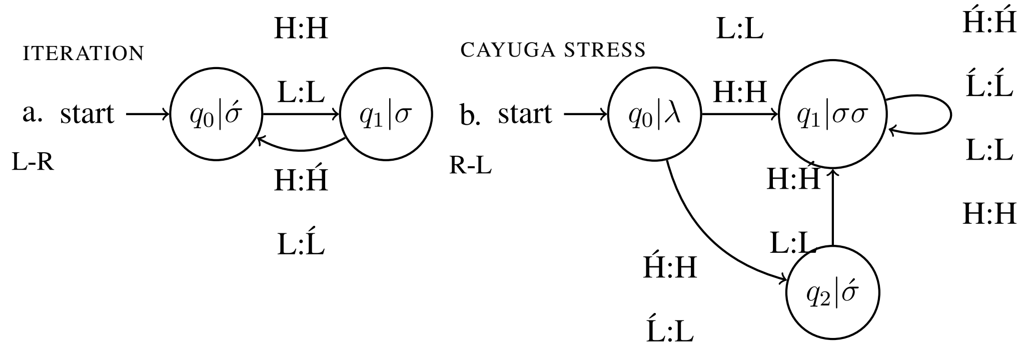
input:	$\sigma \quad \acute{\sigma} \quad \sigma \quad \acute{\sigma} \quad \sigma$	input:	$\sigma \quad \acute{\sigma} \quad \sigma \quad \acute{\sigma} \quad \sigma \quad \acute{\sigma}$
states:	$q_2 \leftarrow q_2 \leftarrow q_2 \leftarrow q_2 \leftarrow q_1 \leftarrow q_0$	states:	$q_2 \leftarrow q_2 \leftarrow q_2 \leftarrow q_2 \leftarrow q_2 \leftarrow q_1 \leftarrow q_0$
output:	$\sigma \quad \acute{\sigma} \quad \sigma \quad \acute{\sigma} \quad \sigma$	output:	$\sigma \quad \acute{\sigma} \quad \sigma \quad \acute{\sigma} \quad \acute{\sigma} \quad \sigma$

For QS languages, a clash may occur incidentally when heavy syllables fall next to another stress that resulted from binary iteration, assuming there is a requirement to stress all heavies in that language. This can be seen for Fijian in (68) above. Another interesting case is that of Cayuga (Foster 1982). In Cayuga, stress iterates onto even-numbered syllables counting from the left, with some further requirements. When the penult is even, it receives main stress, regardless of weight (a-b). If the penult is odd and heavy it receives main stress (c). If the penult is odd and light, however, the antepenultimate syllable is stressed instead (d):

(96)
a. LĹLĹL
b. HĹLH́L
c. LĹLĹH́L
d. LĹLĹLL

Relying not only on weight but on an apparent even/odd parity distinction, the pattern of Cayuga looks on the surface to be more complex than the others discussed to this point. However, it too breaks down into an OSL iteration function and an EO cleanup function. This is partly because the iterative placement of stress provides a pseudo-parity count on the surface in a way that is local, without explicitly tracking the parity of the entire word or a specific syllable:

(97)



The L-OSL function in (a) is another example of ITERATION as seen in Wergaia. It applies stress in a binary iterative manner starting from the peninitial, regardless of weight. The function CAYUGA STRESS in (b) corrects overapplications of stress, much like in Garawa or Paiute. When it encounters an unstressed final syllable, this indicates that the penult is even and stressed, and so the transduction moves to q_1 where everything is output faithfully. If instead it encounters a stressed final syllable, it removes the stress and moves to q_2. From here, a heavy syllable is stressed while a light syllable is left unchanged, moving to q_1.

Despite some conceptual similarities to the SOUR GRAPES function discussed in earlier chapters, (97b) is EO – the maximal input substrings leading to a non-identity transition is $\#\sigma$ – while the second function in the sour grapes pattern is not. Thus, the composed map adheres to the stated single-OSL plus EO restriction and correctly describes the Cayuga stress function.

(98)		(97a)		(97b)
a. LLLLL	$\mapsto$	LĹLĹL	$\mapsto$	LĹLĹL
b. HLLHL	$\mapsto$	HĹLH́L	$\mapsto$	HĹLH́L
c. LLLLHL	$\mapsto$	LĹLĹHĹ	$\mapsto$	LĹLĹH́L
d. LLLLLL	$\mapsto$	LĹLĹLĹ	$\mapsto$	LĹLĹLL

A derivation for inputs LLLHL and LLLLLL are given here:

(99) ITERATION:

input:	L	L	L	H	L		
states:	$q_0 \rightarrow q_1 \rightarrow q_0 \rightarrow q_1 \rightarrow q_0 \rightarrow q_1$						
output:	L	Ĺ	L	H́	L		

input:	L	L	L	L	L	L
states:	$q_0 \rightarrow q_1 \rightarrow q_0 \rightarrow q_1 \rightarrow q_0 \rightarrow q_1 \rightarrow q_0$					
output:	L	Ĺ	L	Ĺ	L	Ĺ

CAYUGA STRESS:

input:	L	Ĺ	L	H́	L
states:	$q_1 \leftarrow q_1 \leftarrow q_1 \leftarrow q_1 \leftarrow q_1 \leftarrow q_0$				
output:	L	Ĺ	L	H́	L

input:	L	Ĺ	L	Ĺ	L	Ĺ
states:	$q_1 \leftarrow q_1 \leftarrow q_1 \leftarrow q_1 \leftarrow q_1 \leftarrow q_2 \leftarrow q_0$					
output:	L	Ĺ	L	Ĺ	L	L

So far it has been shown that bounded lookahead is a property shared by a wide range of unidirectional iterative patterns in in both QI and QS stress. The inability of the OSL stress ITERATION function to provide any lookahead is made up for by EO cleanup functions that instantiate the phonological properties of clash, lapse, and non-finality in way that is restrictive and captures their limitation to the word edge. Despite the descriptive differences and apparent surface complexity of some of the patterns, the explicit compositional analyses confirm that they all share fundamental OSL plus EO computational properties, described here as the atomic properties of stress. While surface differences that appear quite substantial suggest a more disparate typology for stress when taken at face value, they are actually of equal complexity.

5.3 Parity-counting patterns

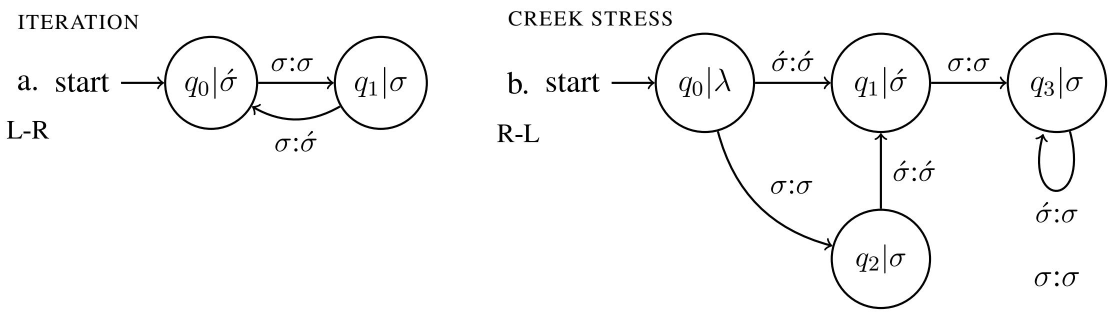
The proposal that iterative stress is the composition of an OSL function with EO functions is a testable hypothesis about the typology of iterative stress patterns. It makes explicit claims about what a possible iterative pattern can be. This is exemplified by the pattern of Creek (Haas 1977; Halle and Vergnaud 1987; Hayes 1995). In words of all light syllables, stress falls on whichever of the penult or ultima is even, counting from the left (this ignores syllable weight and its effects, which are orthogonal to the point made here):

(100) a. $\sigma\sigma\sigma\acute{\sigma}$

b. $\sigma\sigma\sigma\acute{\sigma}\sigma$

Parity-counting in general is properly regular when considered as a formal language i.e. stringset (Heinz 2007b; Rogers et al. 2013; Graf 2017). This suggests that the stress *function* for Creek may be of a higher complexity than what is needed for iterative stress. This is demonstrated by the fact that an OSL plus OSL composition could derive the correct stress map for Creek by using stress in the intermediate form as a pseudo-parity count. The transducers are as follows:

(101)



The L-OSL function in (a) stresses every other syllable starting with the peninitial, just like other iterative functions seen above. The second function in (b) reads right to left, keeping the first stress it encounters but removing all others. This approach is analogous to that of Halle and Vergnaud (1987), which also employs a stress-removal function dubbed "conflation". The compositional analysis here generates the attested surface pattern:

(102)

		(101a)		(101b)	
a.	$\sigma\sigma\sigma\sigma\sigma$	$\mapsto$	$\sigma\acute{\sigma}\sigma\acute{\sigma}\sigma$	$\mapsto$	$\sigma\sigma\sigma\acute{\sigma}\sigma$
b.	$\sigma\sigma\sigma\sigma\sigma\sigma$	$\mapsto$	$\sigma\acute{\sigma}\sigma\acute{\sigma}\sigma\acute{\sigma}$	$\mapsto$	$\sigma\sigma\sigma\sigma\sigma\acute{\sigma}$

The potentially unlimited alteration to the string in the form of deleting all stresses except the rightmost one is not definable with an EO function, and so such a pattern is excluded from the typology of iterative stress under the theory adopted here. If Creek is truly an unbounded parity-counting pattern, then this is a desirable outcome – the proposal outlined in this chapter is a restrictive characterization of iterative patterns, and long-distance stress patterns are expected to be subject to a different analysis given their different computational properties.

However, there is reason to believe that Creek may not be a truly unbounded pattern. Consider again the pattern of Cayuga, analyzed above. It follows a similar generalization to that of Creek, with the important difference that additional stresses are *not* removed by the second function. As such, Cayuga is an iterative pattern that *does* adhere to the EO restriction. This indicates that, should further study of Creek reveal surface secondary stress, then it would fall in line with other, more widely attested iterative patterns. There is evidence that this is the case (Martin and Johnson 2002; Martin 2011). The same can be said of the similar pattern of Cairene Arabic, which has been argued to feature secondary stress (Harrell 1957; Kentsowicz 1980).

Notably, the difference in complexity between the two shows that, separate from any other factors contributing to its existence, secondary stress is a computational aid that is directly responsible for the different level of complexity of iterative patterns when compared to unbounded patterns. It also shows that, as a restriction, EO makes substantive, real distinctions when it comes to stress typology.

CHAPTER 6: BIDIRECTIONAL AND LONG-DISTANCE STRESS PATTERNS

6.1 Introduction

This chapter provides a computational analysis of bidirectional iterative patterns and long-distance patterns. Unlike the iterative patterns of Chapter 3, where stress iterates in just one direction, here I present some cases where stress iterates in *both* directions. This means that an OSL plus EO decomposition is insufficient – bidirectional iteration of stress requires at least an OSL plus OSL decomposition.

Long-distance patterns differ from the iterative patterns in that they are fundamentally non-local – no decomposition into OSL and EO atoms is possible. As a holistic map, these patterns appear quite complex – default to opposite (DTO) stress can be computed with a single subsequential function, while default to same (DTS) stress patterns require an unbounded search of the string in both directions. Thus, while DTO patterns are subsequential, DTS patterns are weakly deterministic (Hao and Anderson 2019; Koser and Jardine 2020b) per Meinhardt et al. (submitted)'s definition of WD based on interaction (see also Heinz and Lai (2013)). Even more complex, properly regular cases come from languages such as Pulaar (Niang 1997) and Nanti (Crowhurst and Michael 2005), which require interaction in the composition to compute the pattern (Koser and McCollum to appear). I demonstrate that the higher complexity is a direct result of the suprabinary weight scale that these languages are sensitive to. These facts mean that DTS and Pulaar-like patterns cannot be computed with a single deterministic function, indicating that they are properly regular when defined as a single step from input to output.

However, all long-distance patterns can be expressed as the composition of *strictly piecewise* functions (SP; Burness and McMullin 2020), the functional analogue of strictly piecewise languages (Heinz 2009; Rogers et al. 2010). This provides a more restrictive hypothesis for the atomic elements of long-distance stress than full subsequential power, which I demonstrate leads to pathological predictions that are eliminated under an SP restriction. The relationship between the scale and the input alphabet i.e. syllable inventory of long-distance patterns is also discussed, showing that more complex generalizations that can be calculated with at least two input alphabet symbols cannot be calculated with an alphabet of just one input symbol.

6.2 Bidirectional iterative patterns

Bidirectional iterative patterns differ from the iterative patterns discussed in Chapter 3 in that iteration of stress occurs in both directions through the string. As iteration occurs in two directions, the decomposition of any bidirectional iterative pattern must contain two OSL functions that iterate stress in opposite directions.

6.2.1 Cahuilla

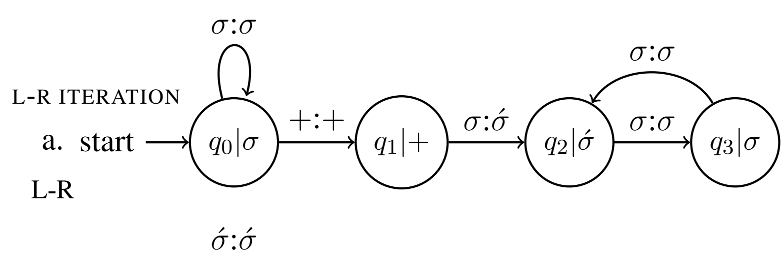
One example comes from Cahuilla (Seiler 1977; Levin 1988a; Idsardi 1992). In all-light syllable forms in the language, the first syllable of the root is stressed, while further stresses iterate to alternating syllables away from the root-initial stress in *both* directions:

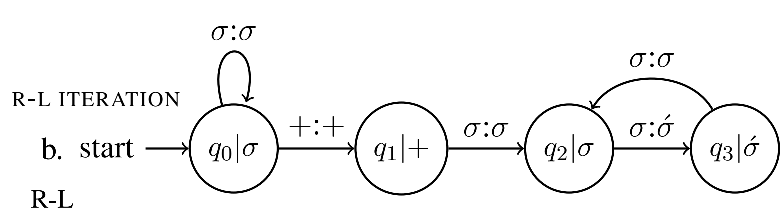
(103)

$$\acute{\sigma}\sigma + \acute{\sigma}\sigma\acute{\sigma}$$
$$\sigma\acute{\sigma}\sigma + \acute{\sigma}\sigma\acute{\sigma}\sigma$$
$$\acute{\sigma}\sigma\acute{\sigma}\sigma + \acute{\sigma}\sigma\acute{\sigma}\sigma\acute{\sigma}$$
$$\sigma\acute{\sigma}\sigma\acute{\sigma}\sigma + \acute{\sigma}\sigma\acute{\sigma}\sigma\acute{\sigma}\sigma$$
$$\ldots$$

While an OSL plus EO composition for iterative stress was proposed in the previous chapter to provide a tighter computational characterization than a single properly subsequential function, here a compositional analysis is *necessary* – there is no way to deterministically compute the output with a single monodirectional function. This is because, since the count for iteration is determined by the position of the root-initial syllable, a left to right iteration function alone would not know when to begin iteration of stress in the prefix. Similarly, a right to left function would not know when to begin iteration of stress in the root. Thus, no deterministic function operating in a single direction can compute the correct map for a bidirectional pattern such as Cahuilla. Instead, the pattern can be treated as the composition of two OSL functions iterating stress in opposite directions after the root boundary is encountered. Note that the transducers below are minimized for clarity:

(104)





L-R ITERATION makes no changes until it encounters the root boundary, moving to q_1. From here, the first syllable is stressed in the transition to q_2. The next syllable is left unchanged, and then iterative stress is applied in the transitions between q_2 and q_3. R-L IT-

ERATION, working in the other direction, preserves the stresses created by the first function and, after reading the root boundary, applies iterative stress in the transitions between q_2 and q_3. This OSL plus OSL decomposition correctly describes the Cahuilla stress map. An example derivation is given here:

(105) L-R ITERATION:

input:	σ	σ	$+$	σ	σ	σ	σ
states:	$0 \rightarrow 0 \rightarrow 0 \rightarrow 1 \rightarrow 2 \rightarrow 3 \rightarrow 2 \rightarrow 3$						
output:	σ	σ	$+$	$\acute{\sigma}$	σ	$\acute{\sigma}$	σ

R-L ITERATION:

input:	σ	σ	$+$	$\acute{\sigma}$	σ	$\acute{\sigma}$	σ
states:	$3 \leftarrow 2 \leftarrow 1 \leftarrow 0 \leftarrow 0 \leftarrow 0 \leftarrow 0 \leftarrow 0$						
output:	$\acute{\sigma}$	σ	$+$	$\acute{\sigma}$	σ	$\acute{\sigma}$	σ

Note that the two functions in the decomposition do not interact – the same correct Cahuilla forms are derived even if the order of the functions is reversed. Intuitively this is because each function only makes changes on one side of a root boundary – they will never alter the same syllable in a string. Thus, the stress map for Cahuilla is WD. As the WD class refers to compositions of non-interacting subsequential functions, it is possible that compositions of non-interacting OSL functions represent a yet undescribed proper subclass of the WD functions – one that is relevant for bidirectional patterns where the functions in the decomposition are output local, such as bidirectional iterative stress and harmony patterns.

6.2.2 Auca

While stress in Cahuilla iterates outward, the opposite pattern is found in Auca (Pike 1964), which features "edge-in" application of iterative stress in both directions. Starting with the initial syllable, stress iterates to alternating syllables until the end of the root. Iterative stresses also occur right to left, starting with the penult, stopping at or one syllable after the root boundary:

(106) a. $\acute{\sigma}$ + $\acute{\sigma}\sigma$

b. $\acute{\sigma}\sigma\acute{\sigma}$ + $\acute{\sigma}\sigma\acute{\sigma}\sigma$

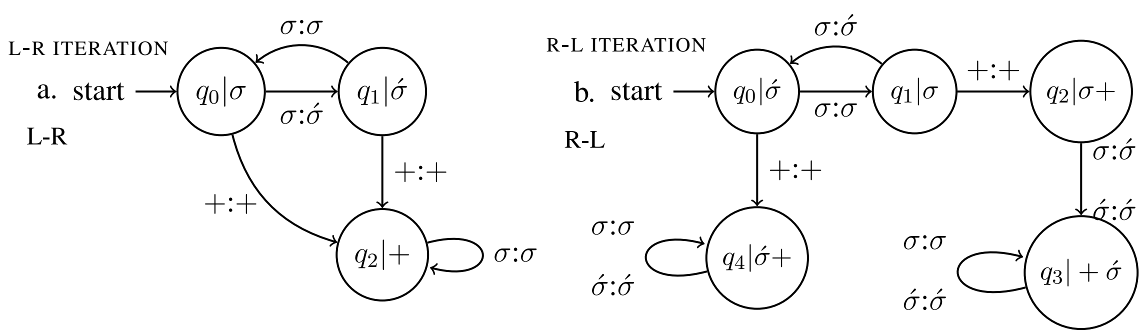
c. $\acute{\sigma}\sigma\acute{\sigma}\sigma$ + $\acute{\sigma}\sigma$

d. $\acute{\sigma}\acute{\sigma}$ + $\sigma\acute{\sigma}\sigma$

e. $\acute{\sigma}\sigma\acute{\sigma}\acute{\sigma}$ + σ

Stress is iterates rightward from the initial syllable, and leftwards from the penult (106a-c). When the final syllable of the root is unstressed, right to left iteration from the suffix continues over the root boundary (106d), but does not iterate further (106e). Like Cahuilla, a deterministic analysis of Auca requires a compositional analysis. No single subsequential function could place stress correctly in both the root and suffix. The Auca stress map can be described with an OSL plus OSL decomposition, where each OSL function handles iteration in one direction and uses the root boundary to determine when iteration should halt:

(107)



L-R ITERATION iterates stress to alternating syllables starting with the initial syllable in the transitions between q_0 and q_1. When a root boundary is encountered, no further stresses are output. R-L ITERATION follows a similar pattern, iterating stress starting from the penult as the transduction moves between q_0 and q_1. R-L ITERATION differs from its counterpart in its behavior upon seeing a root boundary. The output sequence $\sigma+$ brings the transduction to q_2. From here, iteration terminates on the root-final syllable in the case that it is unstressed. An output sequence $\acute{\sigma}+$ instead lands in q_4, where no further changes are made – the root final syllable is not in the iterative chain of R-L ITERATION in this case. Example derivations for (106c) and (106d) are given here:

(108) L-R ITERATION:

input:	$\sigma\ \sigma\ \sigma\ \sigma\ +\ \sigma\ \sigma$	input:	$\sigma\ \sigma\ +\ \sigma\ \sigma\ \sigma$
states:	$0 \rightarrow 1 \rightarrow 0 \rightarrow 1 \rightarrow 0 \rightarrow 2 \rightarrow 2 \rightarrow 2$	states:	$0 \rightarrow 1 \rightarrow 0 \rightarrow 2 \rightarrow 2 \rightarrow 2 \rightarrow 2$
output:	$\acute{\sigma}\ \sigma\ \acute{\sigma}\ \sigma\ +\ \sigma\ \sigma$	output:	$\acute{\sigma}\ \sigma\ +\ \sigma\ \sigma\ \sigma$

R-L ITERATION:

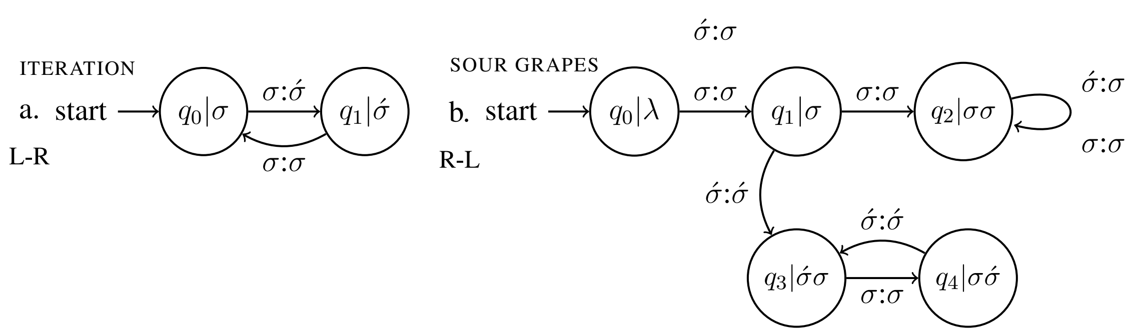
input:	$\acute{\sigma}\ \sigma\ \acute{\sigma}\ \sigma\ +\ \sigma\ \sigma$	input:	$\acute{\sigma}\ \sigma\ +\ \sigma\ \sigma\ \sigma$
states:	$4 \leftarrow 4 \leftarrow 4 \leftarrow 4 \leftarrow 4 \leftarrow 0 \leftarrow 1 \leftarrow 0$	states:	$3 \leftarrow 3 \leftarrow 2 \leftarrow 1 \leftarrow 0 \leftarrow 1 \leftarrow 0$
output:	$\acute{\sigma}\ \sigma\ \acute{\sigma}\ \sigma\ +\ \acute{\sigma}\ \sigma$	output:	$\acute{\sigma}\ \acute{\sigma}\ +\ \sigma\ \acute{\sigma}\ \sigma$

The OSL plus OSL decomposition correctly describes Auca stress. The interaction-freeness of the decomposition hinges on the presence of the root boundary. If there were no root boundary, the only signal to stop iterating stress would be stresses placed by the function operating in the opposite direction. So, while the root boundary in Cahuilla indicates the starting point of stress, in Auca, it instead indicates when stress should halt.

6.2.3 OSL plus OSL decompositions

In the previous chapter, it was demonstrated that a restriction to one OSL function in the composition of an iterative stress pattern excludes patterns such as sour grapes stress from the predicted typology. However, two OSL functions working in opposite directions can describe the sour grapes stress map. The transducers and example derivation are repeated here:

(109)



(110) 5σ 6σ

ITERATION:												
input:	σ	σ	σ	σ	σ	input:	σ	σ	σ	σ	σ	σ
states:	$0 \rightarrow 1 \rightarrow 0 \rightarrow 1 \rightarrow 0 \rightarrow 1$					states:	$0 \rightarrow 1 \rightarrow 0 \rightarrow 1 \rightarrow 0 \rightarrow 1 \rightarrow 0$					
output:	$\acute{\sigma}$	σ	$\acute{\sigma}$	σ	$\acute{\sigma}$	output:	$\acute{\sigma}$	σ	$\acute{\sigma}$	σ	$\acute{\sigma}$	σ
SOUR GRAPES:												
input:	$\acute{\sigma}$	σ	$\acute{\sigma}$	σ	$\acute{\sigma}$	input:	$\acute{\sigma}$	σ	$\acute{\sigma}$	σ	$\acute{\sigma}$	σ
states:	$2 \leftarrow 2 \leftarrow 2 \leftarrow 2 \leftarrow 1 \leftarrow 0$					states:	$3 \leftarrow 4 \leftarrow 3 \leftarrow 4 \leftarrow 3 \leftarrow 1 \leftarrow 0$					
output:	σ	σ	σ	σ	σ	output:	$\acute{\sigma}$	σ	$\acute{\sigma}$	σ	$\acute{\sigma}$	σ

Though sour grapes stress is a logically possible stress map, it is clearly pathological, and a sufficiently restrictive theory of stress should exclude it from the predicted typology. However, while the OSL plus EO characterization of monodirectional iterative stress does acheive this, bidirectional iterative stress patterns require an OSL plus OSL analysis, and so sour grapes cannot be excluded in these cases based on the computational power of OSL versus EO alone. Some other restriction is needed.

A close examination of the transducers in (109) is insightful as to where exactly the pathological flavor of the sour grapes stress pattern is produced. (109a) is a standard iteration function of the type seen in the previous and current chapter. In fact, it is a subgraph of the L-R ITERATION and R-L ITERATION functions in both Cahuilla and Auca and identical to the iteration transducer for Pintupi (among others) in the previous chapter. As such, (109a) presents nothing obvious as a target for a potential restriction to OSL plus OSL compositions. Considering (109b), however, there are some clear differences from (109a) in how the function behaves. In state q_2, it deletes a potentially unbounded number of input stresses. To do this, it uses markup – the transduction will only arrive in q_2 if it sees a $\sigma\sigma$ sequence in the output, and instead move to q_3 upon reading a $\acute{\sigma}\sigma$ sequence. Thus, while (109a) does nothing but iterate stress blindly, (109b) relies on the application of the first function to determine its pathological behavior.

It is clear that, for patterns like Cahuilla and Auca, the functions in the decomposition are more like (109a) than (109b) – both OSL functions merely iterate stress, with some added sensitivity to morphological boundaries. Neither of the functions interact, while sour grapes stress relies on the fact that they do. This, then, is the target of the restriction – the OSL functions in a composition describing a bidirectional stress pattern may not interact – they must produce the same output form no matter what order they are applied in. This is intuitively in line with what Meinhardt et al. (submitted) propose for weakly deterministic compositions of subsequential functions. The loop over state q_2 in SOUR GRAPES also suggests that the "edge-oriented" class of Chapter 3 offers a relevant notion of restrictiveness – this loop does not map identity, and so may make an unbounded number of changes to an input string. Though iterative stress functions must be allowed to iterate stress through the entire word, and so cannot be EO exactly as described in the previous chapter, disallowing this use of markup results in some restricted subclass of OSL functions that may be related to or overlap with the EO class in some way.

Another possible restriction is to require that the image of the functions in an OSL plus OSL composition will be a strictly local (SL) formal language i.e. stringset, SL languages being those that are describable with conjunctions of negative literals (CNLs) (McNaughton and Papert 1971). For example, a simple iterative stress language might be described with the CNL $\neg\ \#\sigma \wedge \neg\ \sigma\sigma \wedge \neg\ \acute{\sigma}\acute{\sigma}$. The first conjunct forbids an unstressed initial syllable; the second conjunct forbids lapses and thus enforces iteration; the third conjunct forbids clashes. Taken together, these literals describe an iterative stress language as a well-formed stringset by determining what strings are not permitted in the language. Early computational work conceived of stress in this way (Heinz 2007a; Rogers et al. 2013; Heinz 2014). A pattern described by the CNL above is found in Murinbata (Street and Mollinjin 1981), which iterates stress to alternating syllables starting with the first.

However, no statement in CNL logic could describe the sour grapes stress language corresponding to (109b). For example, $\acute{\sigma}\sigma$ is forbidden in strings of odd parity, but must be allowed in strings of even parity. Sensitivity to word parity requires knowledge of the entire word, and so the only way to state the grammar in CNLs would be to enumerate each case i.e. $\neg\ \#\sigma\sigma\# \wedge \neg\ \#\acute{\sigma}\sigma\acute{\sigma}\# \wedge \neg\ \#\sigma\sigma\sigma\sigma\#$ ..., and so on. Such a grammar is not finite or local, and so a requirement that OSL functions in a composition have SL images would also ban sour grapes stress.

Interestingly, this indicates that while the FST in (109b) is OSL, the corresponding language is *not* SL. This is because each state in an OSL FST encodes local output information, which allows both an output $\sigma\sigma$ sequence and $\acute{\sigma}\sigma$ sequence to be "licit" in terms of the function. But the CNLs describing an SL language must hold for every string in the language at once. For example, the negative literal $\neg\ \sigma\sigma$ is necessary for even-parity forms, and cannot be "turned off" when evaluating membership of an odd-parity form in the language.

In sum, bidirectional iterative patterns are describable as the composition of two OSL functions that iterate stress in either direction through the string. Adding a requirement that the functions in the composition may not interact results in a more restrictive theory of bidirectional stress patterns that excludes pathological patterns such as sour-grapes stress.

6.3 Long-distance patterns

To this point, only patterns that are fundamentally local have been discussed in detail. I now turn to patterns that are long-distance or "unbounded". Whereas bounded patterns fix stress in a certain location – be it input or output-based – stress in unbounded systems falls potentially anywhere in the word. This is exactly why they are non-local, as stress placement is not bound to a word edge or to a previous output stress.

Based on their surface characteristics, long-distance patterns can be divided into the default to opposite (DTO), default to same (DTS) and Pulaar-like categories. If no restriction beyond subsequentiality is proposed, these patterns all appear to have quite different computational properties. DTO patterns can be described with a single subsequential function, while DTS patterns require a more powerful weakly deterministic composition (Hao and Anderson 2019; Koser and Jardine 2020b). Pulaar-like patterns are yet more complex, as their suprabinary weight scale necessitates interaction, making them non-deterministic and thus properly regular.

However, all long-distance patterns can be described as the composition of atoms that correspond to EO or *strictly piecewise* functions (SP; Burness and McMullin 2020). SP functions model long-distance processes because they are calculated based on the non-contiguous sub*sequences* of a word, unlike SL functions which are defined in terms of contiguous sub*strings*. Just like the SL functions, SP functions are divided into input- and output-local counterparts (ISP and OSP) based on where the relevant information is located. As demonstrated below, the functions that are pertinent to stress exist in the overlapping space between the two – they are both OSP and ISP. A hypothesis that long-distance patterns are composed of SP atoms avoids overgeneration that is possible with unconstrained subsequential power. Intuitively, in an SP function, each input symbol has a *consistent effect* on the behavior of other inputs that appear further along in the string. An example comes from sibilant harmony in Aari (Hayward 1990), where an underlying /s/ surfaces as [ʃ] if another [ʃ] appears at any earlier point in the word – the subsequence ʃ...s is forbidden. This means that [ʃ] determines the behavior of all proceeding underlying /s/ in a consistent, long-distance manner – only [ʃ] will ever surface.

More formally, the SP class is defined based on the *contribution* that each symbol in the alphabet makes to strings of the pattern for a given function, written $\mathtt{cont}_f(a, w)$ for some $a \in \Sigma$ and $w \in \Sigma^*$. Using the example of Aari harmony, given a string $w =$ ʃa, the contribution of input /s/ to the string is [ʃ], or $\mathtt{cont}_f(\text{s}, \text{ʃa}) = \text{ʃ}$. Then, a crucial requirement for SP functions is that this behavior with regards to the contribution propagates consistently to strings whose subsequences are a superset of those found in the original string. The requirement is stated in Burness and McMullin (2020) as follows:

(111) A function f is SP_k iff for each $\sigma \in \Sigma$, either (1) or (2) holds when $\mathtt{cont}_f(\sigma, w_1) \neq$

$\texttt{cont}_f(\sigma, w_2)$:

1. $\texttt{cont}_f(\sigma, w_1) = \texttt{cont}_f(\sigma, w_3)$ for all w_3 such that:

$$\texttt{sub}_{\leq k-1}(w_3) \supseteq [\texttt{sub}_{\leq k-1}(w_1) \cup \texttt{sub}_{\leq k-1}(w_2)]$$

2. $\texttt{cont}_f(\sigma, w_2) = \texttt{cont}_f(\sigma, w_3)$ for all w_3 such that:

$$\texttt{sub}_{\leq k-1}(w_3) \supseteq [\texttt{sub}_{\leq k-1}(w_1) \cup \texttt{sub}_{\leq k-1}(w_2)]$$

In words, this requirement means that when an input alphabet symbol makes a different contribution to two strings w_1 and w_2, given a third string w_3 whose subsequences are a superset of the union of the subsequences of w_1 and w_2, the contribution of that input alphabet symbol to w_3 must match either its contribution to w_1 *or* w_2 – but not both. This ensures that the effect of the previous subsequences in a word on further inputs is consistent, which is a hallmark of SP functions. Note that the definition (111) describes the ISP functions. OSP functions are instead described over the output subsequences of strings i.e. $\texttt{sub}_{\leq k-1}(f(w_1))$.

Returning to the example of Aari, the contribution of /s/ to a string ‘sa’ is [s], while its contribution to the string /ʃa/ is [ʃ]. The union of the subsequences in this case is the set ʃ, a, s. As any w_3 containing these same (and possibly more) subsequences necessarily contains an ʃ, the contribution of /s/ will always be [ʃ], matching its contribution to ‘ʃa’. Thus, the contribution of /s/ to strings in the pattern remains consistent from a set of input subsequences to supersets of that set. If this holds for all alphabet symbols, then the function is SP. I now show that long-distance stress patterns are composed of SP atoms that adhere to this restriction.

6.3.1 Default to opposite

Default to opposite (DTO) patterns are sensitive to a binary weight distinction – heavy and light. DTO places stress on a heavy syllable closest to a specified edge, or – if no heavies are present – a light syllable at the opposite edge. Both logical possibilities, leftmost-heavy or right (LHOR) and rightmost-heavy or left (RHOL) are attested. An example of an LHOR pattern is found in Kwakw'ala (Bach 1975; Hayes 1995):

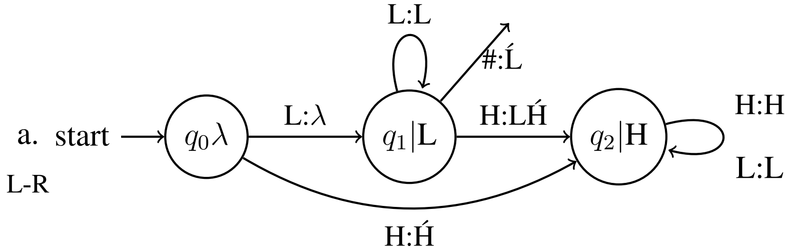
(112)

LLLL	$\mapsto$	LLLĹ
HHHH	$\mapsto$	H́HHH
LHLLLL	$\mapsto$	LH́LLLL
LHLLHL	$\mapsto$	LH́LLHL

Reading the string left to right, the first heavy encountered is stressed. If no heavies are encountered, the final light syllable is stressed instead. As a single function, DTO patterns are properly subsequential (Hao and Anderson 2019; Koser and Jardine 2020b) – this type of pattern loses the property of locality that is characteristic of EO, ISL, or OSL classes. This is because, to determine if any given H or a final L should be stressed, the function must keep track of the lack or presence of heavy syllables for the entire length of the word up to that point, as shown in the following diagram:

(113) $\overleftarrow{\text{LL}}\boxed{\text{H́}}\text{LHL} \quad \overleftarrow{\text{LLLLLL}}\boxed{\text{Ĺ}}$

A transducer is as follows:

(114) 

The transduction proceeds left to right. The transition from q_0 to q_1 takes an input L and "waits" with λ – no output can be written until more input symbols are seen. If the word is light syllables only, the transduction loops in q_1, outputting Ĺ when the word ends, as seen in the exit transition on input #. If at any point a heavy is seen, it is output with stress and the transduction moves to q_2, where no further changes are made. In intuitive terms, being in q_1 means that only light syllables have been read, whereas being in q_2 means at least one heavy has been read. Note that this is not local information – LHOR is a long-distance pattern. That the transducer encodes this information about an arbitrary previous number of symbols it has seen makes the function properly subsequential. The following derivations for LLLL and LLHL demonstrate the LHOR mapping:

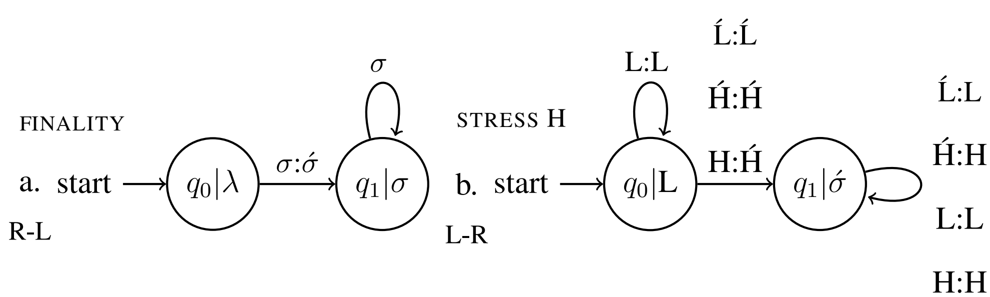
(115)

input:		L		L		L		L		input:		L		L		H		L
states:	q_0	$\rightarrow$	q_1	$\rightarrow$	q_1	$\rightarrow$	q_1	$\rightarrow$	q_1	states:	q_0	$\rightarrow$	q_1	$\rightarrow$	q_1	$\rightarrow$	q_2	$\rightarrow$	q_2
output:		λ		L		L		L Ĺ		output:		λ		L		LH́		L

When evaluating whether any position in the string, light or heavy, should surface with stress, the function depends on information about the – in principle – unbounded sequence of symbols preceding the current one. The loop in q_1 allows for an arbitrarily large sequence of L inputs before stress is ultimately applied. q_1 corresponds to only having read input L for the duration of the transduction, while q_2 corresponds to having seen at least one H somewhere in the string. As such, the pattern cannot be determined locally, and so is not OSL, ISL, or EO.

Though a DTO stress pattern such as LHOR requires subsequential power as a single function map, it can be expressed as the composition of an EO function and an SP function. The EO function places the default "rightmost" stress, and the SP function stresses the leftmost heavy and ensures that all other syllables are output with no stress. Thus, a stressed heavy has a consistent, long-distance effect on the rest of the word that can be described with an SP function. Consider the following transducers:

(116)



The first, EO function is FINALITY, as seen above in Ojibwe in Chapter 3. It stresses the final syllable in the word, representing the default rightmost stress. The only substring leading to a non-identity transition is the empty string λ, and so the function is EO. The second, SP function STRESS H stresses the first heavy syllable it encounters in the transition from q_0 to q_1. This transition also accounts for all-light forms and forms where the only heavy is the final syllable, as the final stress created by the first function should be preserved in these cases. Thus, q_1 corresponds to having output a stress at some previous point in the word, and all further inputs are output as stressless, including removal of the default stress created by FINALITY. In other words, outputting of a stress has a consistent effect on how all other input symbols further along in the word surface, and the function is SP. The following examples demonstrate the combined map for input LLLLL and LHHLL:

(117) FINALITY:

input:	L	L	L	L	L	input:	L	H	H	L	L
states:	$1 \leftarrow 1 \leftarrow 1 \leftarrow 1 \leftarrow 1 \leftarrow 0$					states:	$1 \leftarrow 1 \leftarrow 1 \leftarrow 1 \leftarrow 1 \leftarrow 0$				
output:	L	L	L	L	Ĺ	output:	L	H	H	L	Ĺ

STRESS H:

input:	L	L	L	L	Ĺ	input:	L	H	H	L	Ĺ
states:	$0 \rightarrow 0 \rightarrow 0 \rightarrow 0 \rightarrow 0 \rightarrow 1$					states:	$0 \rightarrow 0 \rightarrow 1 \rightarrow 1 \rightarrow 1 \rightarrow 1$				
output:	L	L	L	L	Ĺ	output:	L	Ḣ	H	L	L

To see that STRESS H is SP, remember the contribution-based definition in (111). The only output for an input L is L, so the condition holds vacuously. An input H may be output as either H or Ḣ. The stressed variant appears only as the contribution to strings whose subsequences are L – words of all light syllables. When at least one H is already present i.e. the set of subsequences contains at least H or Ḣ, then the contribution is always H. So, for any other string comprised of a superset of the subsequences just described – L, H or L, Ḣ – input H will always contribute unstressed H. The contribution of input H is conferred in a consistent manner from set to superset of subsequences. A similar argument can be made for the remaining input symbols, Ḣ and Ĺ. They contribute Ḣ and Ĺ, respectively, to an all-light string, and contribute H and L instead to a string whose set of subsequences minimally contains H or Ḣ. Thus, the superset of subsequences is at least L, H or L, Ḣ, meaning Ḣ and Ĺ always contribute their unstressed variant. The condition in (111) holds for all input alphabet symbols, making STRESS H an SP function.

Specifically, the transducer in (116b) shows that STRESS H is both input- and output-strictly piecewise. For all possible input symbols, for any two strings such that one lands in q_0 and the other lands in q_1, any string containing a superset of the subsequences of those strings also arrives in q_1, precisely because any non-L input brings the transduction to q_1. This makes the function ISP. The same logic applies to the output symbols of the function, making it OSL. For any two strings such that one lands in q_0 and one lands in q_1, a string containing a superset of the output subsequences lands in q_1. This is because q_0 indicates having output only unstressed syllables, while being in q_1 entails having output at least one stressed syllable. Once q_1 is reached, it cannot be exited, which reflects the consistent effect of inputs (or outputs) on ISP (or OSP) functions. STRESS H is thus both ISP and OSP.

All DTO patterns are subject to this EO plus SP compositional analysis, which provides a better typological fit than a hypothesis that long-distance stress atoms are fully subsequential. This is demonstrated in detail in §3.4.

6.3.2 Default to same

Default to same (DTS) patterns are sensitive to the same binary weight distinction as DTO patterns. In DTS systems, an edge-most heavy is stressed if a heavy is present, or a light syllable at the same edge is stressed. Both leftmost-heavy or left (LHOL) and rightmost-heavy or right (RHOR) are attested and assign stress as shown here (LHOL – Lushootseed; RHOR – Klamath (Hayes 1995)):

(118)

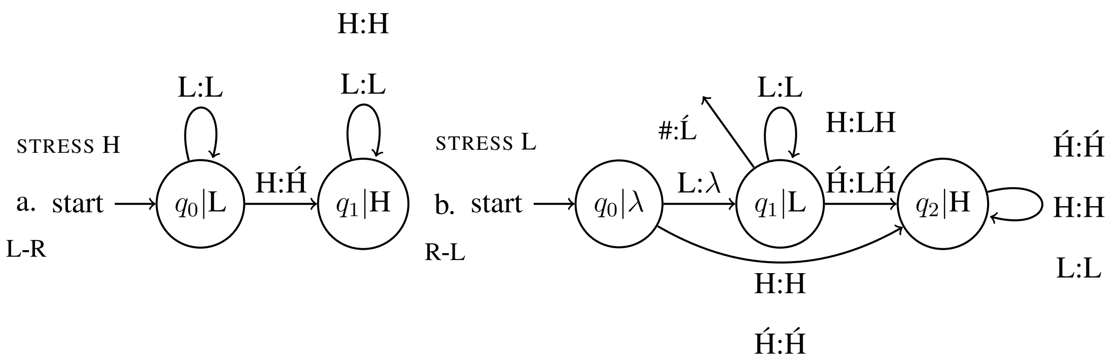
LHOL:	RHOR:
ĹLLL	LLLĹ
ḢHHH	HHHḢ
LḢLLLL	LḢLLLL
LḢLLHL	LHLLḢL

Hao and Anderson (2019) and Koser and Jardine (2020b) demonstrate the that DTS patterns are *not* subsequential. The jump in complexity between DTO and DTS is counterintuitive, given that on a descriptive level, DTO may sound "more long-distance". However, this is not the case – while DTO is monodirectional, DTS is a true bidirectional long-distance pattern. Take LHOL, for example. Locating which heavy syllable is the leftmost requires a leftward search of the string, as no other heavies will precede the leftmost one. On the other hand, verifying that an initial light syllable should be stressed requires a search of all symbols that *follow* it. Determining stress in LHOL (or RHOR) thus requires a long-distance search of the string in both directions:

(119) LHOL: $\overleftarrow{\text{LL}}\boxed{\text{Ĥ}}$LHL $\boxed{\text{Ĺ}}\overrightarrow{\text{LLLLLL}}$

As the pattern is bidirectional, it requires the composition of a pair of contradirectional subsequential functions to describe. The transducers are as follows:

(120)



The first function, STRESS H, reads the string left to right, outputting a stressed H for the first input H it encounters in the transition from q_0 to q_1. This expresses the "leftmost heavy" portion of LHOL stress. The second function, STRESS L reads the string right to left, stressing an L in the case that the word is comprised of only light syllables. It waits with λ in the transition from q_0 to q_1, outputting Ĺ if the word boundary is read from here. Otherwise, if an H is read, the transduction moves to q_2 without making any alterations to the string. A derivation for LLLLL and LHHLL follows:

(121) STRESS H:

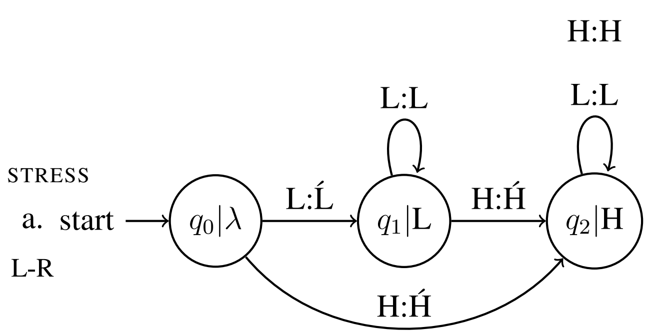
input:	L	L	L	L	L		input:	L	H	H	L	L
states:	0 → 0 → 0 → 0 → 0 → 0						states:	0 → 0 → 1 → 1 → 1 → 0				
output:	L	L	L	L	L		output:	L	H́	H	L	L

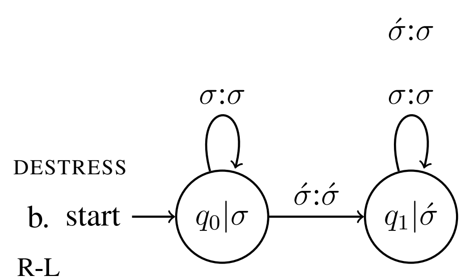
STRESS L:

input:	#	L	L	L	L	L		input:	L	H́	H	L	L
states:	1 ← 1 ← 1 ← 1 ← 1 ← 0							states:	2 ← 2 ← 2 ← 1 ← 1 ← 0				
output:	Ĺ	L	L	L	L	λ		output:	L	H́	HL	L	λ

The composition correctly describes the LHOL stress map. Intuitively, since the first function only ever changes an input H, and the second function only ever changes an input L, the ordering of the functions does not affect the output. While Hao and Anderson (2019) and Koser and Jardine (2020b) describe the pattern as weakly deterministic i.e. the composition of two non-interacting properly subsequential functions, the admission of properly subsequential power is not necessary if interaction is allowed in the composition. Instead, DTS patterns can be described as the composition of two SP functions that operate in opposite directions – one to identify up to two possible locations for stress and a second to preserve whichever stress created by the first function is appropriate. Transducers representing the two functions are as follows:

(122)





The STRESS function in (122a) reads the string left to right, stressing the first syllable it encounters and the first heavy syllable it encounters. It thus accounts for both the default stress and the leftmost heavy stress. In (122b), DESTRESS reads right to left, preserving the first input stress it encounters in the transition to q_1, removing all others. Note that for DESTRESS, weight is irrelevant. A derivation for LLLLL and LHHLL are given here:

(123) STRESS:

input:	L	L	L	L	L		input:	L	H	H	L	L
states:	$0 \rightarrow 1 \rightarrow 1 \rightarrow 1 \rightarrow 1 \rightarrow 1$						states:	$0 \rightarrow 1 \rightarrow 2 \rightarrow 2 \rightarrow 2 \rightarrow 2$				
output:	Ĺ	L	L	L	L		output:	Ĺ	H́	H	L	L

DESTRESS:

input:	Ĺ	L	L	L	L		input:	Ĺ	H́	H	L	L
states:	$1 \leftarrow 0 \leftarrow 0 \leftarrow 0 \leftarrow 0 \leftarrow 0$						states:	$1 \leftarrow 1 \leftarrow 0 \leftarrow 0 \leftarrow 0 \leftarrow 0$				
output:	Ĺ	L	L	L	L		output:	L	H́	H	L	L

The default syllable – the first, in LHOL – is always stressed by the STRESS function. STRESS also stresses the first heavy syllable it encounters, leaving other syllables unchanged. The function thus stresses two potential targets – the default and a leftmost heavy. The second function DESTRESS reads the string in the opposite direction, preserving the first stress it encounters and deleting any others. In an all light form, the default stress is preserved. In forms with at least one heavy syllable, the input stress on the leftmost heavy is preserved and all others are removed.

Both functions in the composition are SP. STRESS has two inputs, L and H. L contributes Ĺ only in the case that it is the first syllable – when the subsequences of the string are $\{\lambda\}$. For any other set of subsequences, L contributes unstressed L instead. The union of a set containing only the empty string with another non-empty set will return exactly that non-empty set. As a result, the contribution of L always matches its contribution to the non-empty set for any other string composed of subsequences of that set. Input H contributes unstressed H to strings with subsequences that contain at least one H, {H} or {L, H}. If the set of subsequences is only {L}, the contribution is H́ instead. A string composed of a superset of these subsequences must contain an H, and so input H always contributes unstressed H to such strings. This exhausts the set of input alphabet symbols for STRESS –

the function is SP.

The transducer shows explicitly that the function is both ISP and OSP. Any string that lands in q_1 has only L for its input subsequences. Any string that lands in q_2 contains at least one input H. For any other string containing a superset of these input subsequences, that string arrives in q_2, making STRESS ISP. It is also OSP – the set of output subsequences for strings landing in q_1 is maximally L, Ĺ. For strings landing in q_2 the set of out output subsequences contains at least H́. Any string containing a superset of these subsequences lands in q_2, from which the transduction cannot return. STRESS is both ISL and OSL.

DESTRESS has two inputs, σ and $\acute{\sigma}$. The only output for σ is itself, σ, and so the condition on SP functions in (111) holds vacuously. For $\acute{\sigma}$, the potential outputs are σ and $\acute{\sigma}$. $\acute{\sigma}$ is contributed to strings whose subsequences are unstressed only. σ is contributed instead when the set of subsequences contains a stressed syllable, $\acute{\sigma}$. A superset of these sets of subsequences contains $\acute{\sigma}$, and so the contribution of $\acute{\sigma}$ to such strings is always unstressed σ. DESTRESS is also both ISP and OSP. For both input symbols, for any two strings such that one arrives in q_0 and one arrives in q_1, any other string containing a superset of the input subsequences of those strings arrives in q_1, because it necessarily contains a stressed input. The same holds for the output subsequences – for both the input and output, q_0 represents only unstressed subsequences, while q_1 represents having input and output at least one stress. Once q_1 is reached, is cannot be exited. The function is both ISP and OSP.

Whereas an EO plus SP composition is sufficient for DTO, it fails for DTS. Consider for example an EO function that marks the initial syllable in an LHOL pattern, encoding the "default" stress. Now, an SP function working right to left will not know when a given heavy syllable is the leftmost one. Attempting a right to left pass instead leaves the function unable to discern when the default stress should be removed, as it cannot "see" ahead to check if a heavy syllable is coming. Thus, the same composition strategy that worked for DTO patterns cannot capture DTS patterns. Even separating the default stress aspect of STRESS in (122a) into a third, EO function merely results in another step, not a change in the computational nature of stress assignment – it still requires a composition of two SP functions, whether the default stress is encoded by STRESS or not.

By the same token, the strategy applied in (122) for DTS cannot be applied to DTO patterns such as LHOR. If a single right to left SP function marks both the final default and first heavy syllable it encounters, then stress is placed improperly in forms with multiple heavy syllables, as a right to left function cannot know when a given heavy is the leftmost one. If the first SP function made greater use of markup by stressing *every* heavy syllable, then the second, left to right function could behave as DESTRESS in (122), preserving the first stress and removing others. However, this strategy for DTO fails for DTS patterns because left to right marking of every heavy in the string in LHOL, for example, leaves the following right to left function with no way to know if a particular heavy it encounters is the last one i.e. the leftmost. In other words, an attempt to unify the analysis of DTO and DTS patterns fails for one of the categories of patterns.

There a several takeaways here. First, though DTO and DTS are both long-distance and receive similar treatment in much of the stress literature (see Hayes (1995), for example), computational analyses suggest a different conception of stress typology based on their computational properties. The differences between the two types of patterns – initially described in the computational literature as a difference between subsequentiality and weak determinism by Hao and Anderson (2019) and Koser and Jardine (2020b) – persist here, and are not reconciled by the more granular analysis as the composition of maximally restrictive atoms. This is precisely because the two patterns do differ at a fundamental computational level. Thus, any particular formalism that treats the patterns in the same way is able to do so because of the properties of that formalism, not the patterns themselves. As mentioned in an earlier chapter, abstract computational characterizations of phonological phenomena such as stress are useful in part because they allow us to study how the computational properties of such phenomena change when they are inserted into a particular model of phonology. It allows us to appraise the computational properties of the models themselves.

The difference in the computation between DTS and DTO is also not a factor of the size of the input alphabet – both are sensitive to a binary heavy versus light distinction. Instead, it comes from the way in which the patterns interact with the alphabet, specifically in terms of directionality. DTO need only consider a long-distance portion of the string in a single direction, while DTS must track long distance information in both directions.

While DTO and DTS patterns cannot be completely unified in terms of the computational properties of their individual atoms, the compositional analysis presented above still provides an explicit hypothesis for DTO and DTS that is more restrictive than allowing properly subsequential atoms that may interact, while also allowing DTO and DTS patterns to be described. However, it is shown below in the case of Pulaar that SP plus SP maps can describe at least some properly regular patterns, which the *non*-interacting subsequential functions of weak determinism (Meinhardt et al. submitted) cannot. Further work is thus needed to reign in the power of SP compositions and ensure that only those properly regular patterns that correspond to possible stress generalizations may be described with SP plus SP compositions. Like OSL plus OSL and sour grapes, whatever properties of SP functions allow for regular patterns can be targeted for further restrictiveness. I now turn to cases of patterns with a suprabinary weight scale.

6.3.3 Suprabinary scales

DTS and DTO patterns provide examples of the value of analyzing long-distance stress patterns as a series of atomic stress functions. Though DTO and DTS as holistic input-output maps appear properly subsequential and weakly deterministic, respectively, an analysis of the individual functions comprising the patterns indicates that they combine generalizations that are maximally EO and SP.

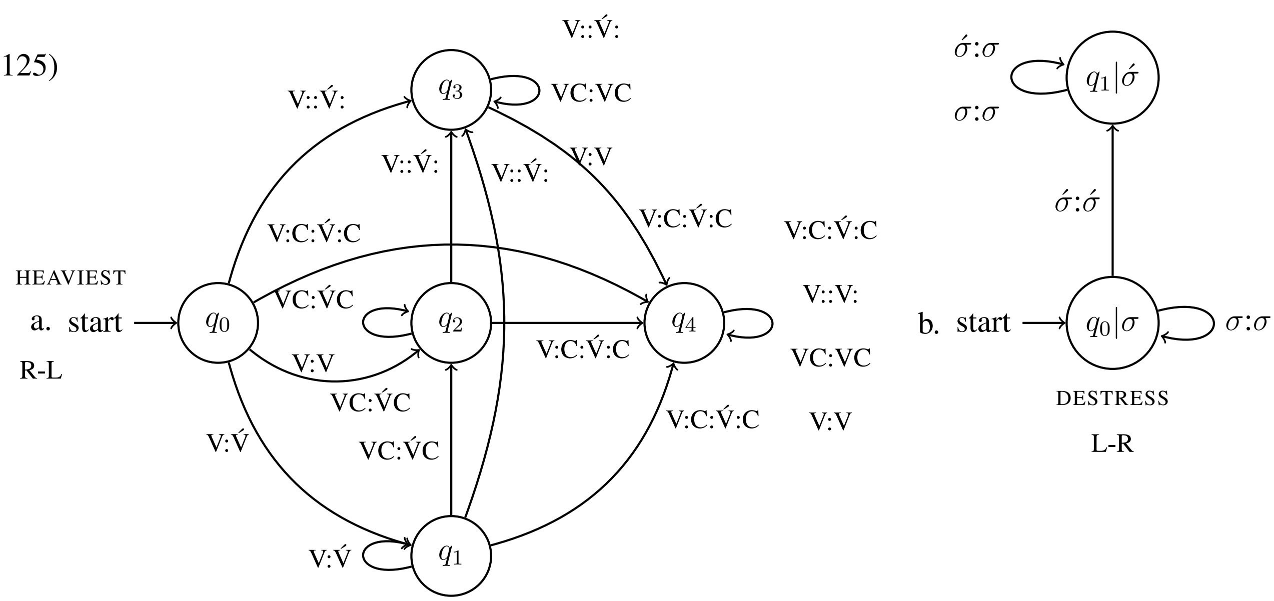
The binarity of the scale in DTO and DTS is exactly what allows for a subregular analysis of the patterns as a holistic map. When a heavy syllable is encountered, it can safely be stressed, as there can be no better potential target for stress elsewhere in the string. However, there are patterns that appear even more complex than DTO or DTS from a non-atomic perspective. In languages with a suprabinary scale, a heavy (or any other non-maximal syllable) cannot be stressed right away, as a better target may occur later in the word, and a single function cannot "see" the rest of the string. Consider the case of Mauritanian Pulaar (Niang 1997), which has the following quaternary weight scale: V:C > V: > VC > V. Main stress is placed on the leftmost heaviest syllable in the word:

(124)	a. á.du.na	'world'	e. bá:.wa:.do	'weak person'
	b. bá.la.be	'shoulders'	f. jol.ti.nó:.wo	'person who removes out of'
	c. da.dór.de	'waist'	g. ha:l.pu.lá:r.ʔen	'speakers of Pulaar'
	d. tál.lor.de	'place for rolling over'	h. ka:.sa.má:s.na:.jo	'person from Casamance'

In words of all light syllables, stress falls on the initial syllable (124a-b). In words where a VC syllable is the heaviest, stress falls on the leftmost of these (124c-d). In words where a V: syllable is heaviest, stress falls on the leftmost of these (124e-f). In words with one or more V:C syllables, stress falls on the leftmost of these (124g-h). Pulaar is not subsequential because a parse of the string in one direction cannot locate the heaviest syllable alone. In (124d), for example, a left to right function cannot safely stress the first syllable because it has no way of knowing if something heavier is coming later in the string. An attempt at a right to left function fails as well, as – for any syllable – it can never know if it has found the leftmost heaviest or if something heavier remains to be seen.

Pulaar is also not WD – an analysis relying on interaction-free subsequential functions fails because of the increased number of levels in the weight scale. For example, when the STRESS function of DTS locates a heavy syllable, it knows it has found the correct target for stress because heavy is the top level of its binary weight scale – no better target will appear elsewhere. This fails in Pulaar because it needs information about syllables further along in the word. To properly assign stress, the first function must proceed from the right to the left and mark every heaviest syllable it has encountered thus far in the computation. This is because, reading right to left, if the first function encounters a CVC, it cannot “see” the rest of the string and so has no way of knowing if something heavier is present elsewhere. Any CV syllable is a potential target until CVC, CV:, or CV:C is read, and any CVC syllable is a potential target until CV: or CV:C is read, and so on. The first function must stress the last syllable in the string, as this is the heaviest (only) syllable it has encountered. Unless this is the best target for stress, the second function will always need to remove this stress. Reading from left to right, the second function preserves stress on the leftmost stressed syllable and removes it from other (i.e., lighter or more rightward) syllables. This decomposed analysis is possible because the first function creates a string where the leftmost heaviest syllable is the first stressed syllable the second function encounters. Despite the increased level of complexity as a holistic map, Pulaar can be described as the composition of two SP functions in the same manner as DTS stress. The transducers are as follows:

(125)



The function HEAVIEST reads the string from right to left. Each state corresponds to having read at least one syllable of the corresponding weight level for that state. If only V is read in the input, the transduction loops in q_1. If at least one VC and nothing heavier has been read, the transduction loops in q_2. q_3 encodes having seen at least one

V:, and q_4 encodes having seent at least one V:C. Crucially, each state continues to stress syllables of their corresponding weight level, while outputting all lighter types of syllables as unstressed. The transduction does not "go back" – once a certain type of syllable is read, only syllables that are equally or more heavy can receive stress. HEAVIEST thus creates strings marked up in a way that the leftmost heaviest syllable bears stress. The second function is DESTRESS from the analysis of DTS patterns above. Reading left to right, the first stress it encounters is lodged on the leftmost heaviest syllable and so should be preserved in the transition from q_0 to q_1. All other stresses are removed in the loop over q_1. An example derivation for a V.V.V. word (as in (124a)) and a V:.V.V:C.V:.V word (as in (124h)) are given here:

(126) HEAVIEST:

input:	V.	V.	V.		input:	V:.	V.	V:C.	V:.	V		
states:	$1 \leftarrow 1 \leftarrow 1 \leftarrow 0$				states:	$4 \leftarrow 4 \leftarrow 4 \leftarrow 3 \leftarrow 1 \leftarrow 0$						
output:	V́.	V́.	V́.		output:	V:.	V.	V́:C.	V́:.	V́		

DESTRESS:

input:	V́.	V́.	V́.		input:	V:.	V.	V́:C.	V́:.	V́
states:	$0 \rightarrow 1 \rightarrow 1 \rightarrow 1$				states:	$0 \rightarrow 0 \rightarrow 0 \rightarrow 1 \rightarrow 1 \rightarrow 1$				
output:	V́.	V.	V.		output:	V:.	V.	V́:C.	V:.	V

The decomposition correctly describes the Pulaar stress map. By marking up the string, HEAVIEST essentially reduces the quaternary weight scale of Pulaar to a binary one – stressed and unstressed – and DESTRESS picks the leftmost of the stressed syllables. Both functions in the decomposition are SP. HEAVIEST has four possible inputs, one for each level of its weight scale. For the highest level V:C, the only possible output is itself, and so the requirement of SP functions holds vacuously. For the next highest level V:, there are two possible outputs – V: and V́:. Stressless V: is contributed to strings whose set of subsequences contain the heavier V:C. To all other strings i.e. those strings whose set of subsequences is some combination of V, VC, and V: only, the contribution of V: is V́:.

Thus, any string containing a superset of those subsequences necessarily contains the heavier V:C, and so the contribution of V: to such strings will always be the stressless V:. A similar demonstration can be made for the other levels of the Pulaar scale – V and VC contribute their stressless forms to strings with subsequences of a heavier level and contribute their stressed form to strings whose subsequences do not contain a heavier syllable. This means that, just like V:, V and VC always contribute their unstressed forms to any string containing a superset of the relevant subsequences. This exhausts the input alphabet of the HEAVIEST function, showing that it is SP. That DESTRESS is also SP was demonstrated in the composition of LHOL above in (122). Thus, Pulaar stress is described by the composition of two SP functions.

Like the other SP functions in long distance stress compositions, HEAVIEST is also both ISP and OSP. Consider the possible inputs for q_1 and q_2, V and VC. Any string landing in q_1 contains only V input subsequences. For q_2, the set of input subsequences is maximally V, VC. Thus, for any string containing a superset of these input subsequences, the transduction arrives in q_2, from which point it cannot return. The same holds for the output subsequences of q_1 and q_2 – any string arriving in q_1 contains only V́ outputs, while q_2 has V́C in its set of output subsequences. Any string with a superset of these output subsequences thus arrives in q_2, from which the transduction cannot return and V and V́C are the only outputs. This same relationship between states and subsequences holds for every state in the transducer. Whenever an as-of-yet unseen subsequence of either the input or output is seen, the transduction enters a state from which it cannot return to a previous state. The effect of all previous input and output subsequences is consistent from state to state as additional subsequences are seen, and so HEAVIEST is both ISP and OSP.

Patterns like Pulaar necessitate the use of markup when analyzed as the composition of two functions. This indicates that an attempt to restrict the expressivity of stress compositions by avoiding markup is doomed to fail – interaction is a necessity because of patterns such as Pulaar that appear properly regular from a non-atomic perspective. Instead, a restriction on the complexity of the functions in the composition provides a better hypothesis for stress typology than one which allows interaction of more expressive functions. Limiting long-distance stress atoms to SP power at most rules out pathological patterns that are possible with subsequential functions, even with no markup in the composition.

That DTS and Pulaar-like patterns require a pair of contradirectional SP functions to describe, while DTO patterns require only an EO plus SP composition exactly mirrors what has been demonstrated here for atoms of iterative stress patterns. Unidirectional iterative patterns are the composition of one output-local atom and one EO atom, while bidirectional patterns require the composition of two output-local atoms. DTO is long-distance in only one direction, and so requires a single SP atom with an EO atom. DTS and Pulaar, on the other hand, have bidirectional long-distance properties, and so require two SP functions to describe. Thus, unidirectional and bidirectioanl stress patterns share similar computational properties, and the main differences emerge from whether the language employs local stress atoms or long-distance ones.

Any unbounded pattern, stress or otherwise, that employs a suprabinary scale requires an analysis like the one presented here for Pulaar (Koser and McCollum to appear). Sensitivity to an increased scale is what gives Pulaar-like patterns this property, rather than the syllable inventory itself. For example, some DTS languages do contain VC, V:, and V:C syllables, but equate all of them in terms of weight – they are simply "heavy". Pulaar instead treats the heavy syllables gradiently, prioritizing some over others. Thus, the number of distinctions that a phonological pattern is sensitive to affect the nature of its computation, rather than the size of the alphabet i.e. syllable inventory (though a possible exceptional case is discussed below).

It should be noted, however, further increases in the number of levels of a suprabinary scale do not have the effect of increasing the expressivity of the total mapping. Any unbounded pattern with a suprabinary distinction can be described by the composition of two SP functions – nothing more complex. One famously complicated example of unbounded stress assignment comes from Nanti (Crowhurst and Michael 2005). In Nanti, stress placement is sensitive to four weight levels CV:N> CV: > CVN > CV and a three-level sonority distinction in vowels, low > mid > high. This results in a twelve-step scale for weight, with a Ca:N syllable being the best target, and a Ci syllable being the worst. In general, Nanti places main stress (subject to non-finality) on the most stressable target in a word or, if two are tied, the rightmost one:

(127)				
	a. o.ko.ri.kʃi.tá.ka	'she wore nose discs'	e. i.tin.ka.ráa.ʃii.gʒi	'they picked thatch'
	b. já.nɯi.ti	'he walked'	f. nóo.ka.na.kse.ro	'I discarded it'
	c. non.kan.táa.ga.kse	'we will have said'	g. pi.ká.bi.ri.ti	'you get going
	d. noo.gái.ga.ro	'we ate it'	h. non.ka.mán.te	'I will tell'

Placement of stress in the word is variable depending on where the most stressable target is located. This requires a compositional analysis for the same reasons as Pulaar – a single function cannot determine which syllable is the most stressable alone. In fact, a derivation for Nanti will look much like that of Pulaar, but in reverse – the first function HEAVIEST reads left to right, stressing members of the most stressable level in the scale it has encountered. Then, proceeding right to left, DESTRESS preserves the first stress it encounters and deletes all others:

(128)			L-R		R-L	
	a.	o.ko.ri.kʃi.ta.ka	↦	ó.kó.ri.kʃi.tá.ka	↦	o.ko.ri.kʃi.tá.ka
	b.	non.kan.taa.ga.kse	↦	nón.kán.táa.ga.kse	↦	non.kan.táa.ga.kse
	c.	ja.nɯi.ti	↦	já.nɯi.ti	↦	já.nɯi.ti
			HEAVIEST		DESTRESS	

As mentioned above, a consequence of the suprabinary scale is that the first syllable is always stressed by HEAVIEST, as it is a valid potential stress target. This is necessary for forms like (128c), where the first syllable is the best landing site for stress in the word. In (128a), the second syllable is an equally good target, and so it is also stressed. The next two syllables are of equal weight to the first two, but feature a less sonorous vowel – they are lower in the twelve-level scale of Nanti and so are left unstressed. Finally, the penult -ta- is stressed, as it is yet more sonorous than any syllable seen thus far. Non-finality prevents stressing of ultima. The second function encounters stress on the penult and preserves it, deleting all others. The same generalization can be seen in (128b) – progressively stressable targets are stressed left to right by Nanti's HEAVIEST, and all stresses save the last are removed right to left by Nanti's DESTRESS.

As in Pulaar, both functions in the composition are SP. The twelve input symbols have two possible outputs – stressed and unstressed, other than the heaviest Ca:N, which is always output with stress. The stressed variant is contributed to a string only in the case that the subsequences of the string do not contain any syllables of a higher weight level. If it does, an unstressed syllable is contributed instead. As such, the contribution to any string with a superset of the two categories of subsequences is unstressed, as the superset necessarily contains a heavier syllable than the one in question. Just as in Pulaar, HEAVIEST and DESTRESS are SP.

Like Pulaar, the decomposition describes the Nanti stress pattern because the first function marks up the string in a way that allows the second function to identify the rightmost stressable syllable. Thus, despite the increased number of distinctions in the Nanti scale, the input-output map operates in exactly the same manner as Pulaar. As transducers, the difference amounts to an increase in the number of labels on state transitions, as well as an increase in number of states – one for each level of the scale as was seen in Pulaar. This is essentially notational, and does not increase the expressivity of the map.

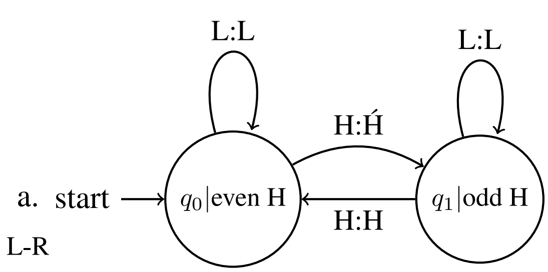
6.3.4 Unattested long-distance patterns

DTO patterns can be described as the composition of EO and SP atoms, providing a more restrictive hypothesis than fully subsequential atoms for long-distance stress. For example, as discussed in Chapter 2, a pattern that stresses every odd heavy syllable in the word, "stress odd H" is not SP because it depends on not only subsequences but an overt parity count of the heavy syllables in the word. To demonstrate, I repeat the pattern here:

(129)	HHLLLHH	$\mapsto$	H́HLLLH́H
	HHHHHH	$\mapsto$	H́HH́HH́H
	LLLLHHH	$\mapsto$	LLLLH́HH́
	LH́HLLLH́L	$\mapsto$	LH́HLLLH́L

The patterning of the heavy syllable demonstrates clearly why the pattern is not SP. When an input H is an odd-numbered H, it will contribute H́ to any string. When it is an even-numbered H, it will contribute unstressed H instead. Consider two strings w_1 = LHLL and w_2 = LHLLH. Stressed H́ is contributed to the former, and stressless H is contributed to the latter. The subsequences of both strings are {L, H}. Now consider two other strings, w_3 = LHL and w_4 = LHHL. Both are composed of {L, H} subsequences and so contain a superset of the subsequences of the w_1 and w_2. As such, we should have an equal contribution from H to both w_3 and w_4, but this is not the case – extending w_3 with H results in unstressed H, while extending w_4 with H results in stressed H́. The pattern is not SP because it relies on overt parity counting. However, it is subsequential – a one-way deterministic function can derive a "stress odd H" generalization. The following transducer is repeated from Chapter 2:

(130)



The function stresses the first heavy syllable it encounters, then leaves the following one unstressed. Light syllables do not receive stress. This cycle continues for the duration of the string, deriving the "stress odd H" pattern. It is not SP, but is properly subsequential for the reasons described above. Thus, the SP boundary excludes patterns that are clearly pathological such as “stress odd H” while still allowing attested long-distance patterns to be described.

The ability to carry out an explicit count of input symbols is what separates subsequential functions from SP functions. This ability is directly responsible for the prediction of pathological patterns such as “stress odd H” using just a single properly subsequential atom. That properly subsequential atoms are not necessary in stress generalizations speaks to the nature of phonological generalizations in general. Whether phonological patterns are local or non-local, they are sensitive to material that is present in the string – substrings for local patterns, or subsequences for long-distance ones. They are *not* determined based on “meta-properties” of the string such as parity or other overt modulo counting. For example, even though iteration of stress can be described in prose as “stress every odd syllable”, an actual numerical count of the syllables is unnecessary. Iterative stress is encoded by enforcement of a certain output structure – repetitions of $\acute{\sigma}\sigma$, for example. But requiring that the output have the structure $\acute{\sigma}\sigma^n$ does *not* require any overt counting at all. The same can be said of a final stress function that outputs stress when it encounters the structure $\sigma\#$ in the input – the “count” of one syllable from the word boundary is nothing more than a requirement imposed on a certain input structure. In other words, it is not really counting at all.

That phonological patterns are limited to interacting with the actual material present in the word makes sense on an intuitive level – what else should phonological grammar concern itself with than the actual phonological units that comprise the string? Computational studies of phonology give us a way to distinguish the kind of “counting” that phonology must engage in from the kind of modulo-counting that can lead to pathological predictions such as “stress odd H”. If the atoms that combine to implement a given surface pattern are limited to EO, SL, or SP power, then no single atomic function can produce a pathological modulo counting pattern. For compositions of atomic functions, certain restrictions that eliminate unwanted patterns such as sour-grapes stress can be stated. However, if subsequential atoms are allowed, then even a single atomic function lets unwanted modulo-counting patterns such as “stress odd H” into the predicted typology. As demonstrated above and in Chapter 3, reference to properly subsequential power is unnecessary – the full range of known stress patterns can be analyzed as the composition of EO, SL, or SP functions. This is analogous to the results of Rogers et al. (2013), who found that all stress patterns can be factored into combinations of local and piecewise constraints when analyzed as formal language sets i.e. as phonotactic restrictions.

Some differences do emerge – for example, no SL or SP formal language can *require* a stress or ensure that *only one* primary stress is present (obligatoriness and culminativity, (Hyman 2006)) – reference to *sets* of substrings and subsequences is needed instead, which can be stated as locally and piecewise-testable constraints, respectively. When analyzing stress as functions, however, these conditions are enforced automatically by the way the functions interact with the string. This is seen in the structure of the transducers representing the functions – at least one transition is taken that will write a stress to the output, and no more than one transition that writes a main stress to the output will be taken. That is, transducers that adhere to these restrictions are structurally within the scope of SL and SP functions. Despite these differences, the computational similarities between stress patterns as formal languages factored into individual constraints and stress patterns as compositions of atomic functions suggests that both kinds of analyses engage with the fundamental computational properties of stress generalizations in a meaningful way. Other work on stress can use these results as a benchmark for the computational expressivity of their models of representation and the input-output map for stress generalizations.

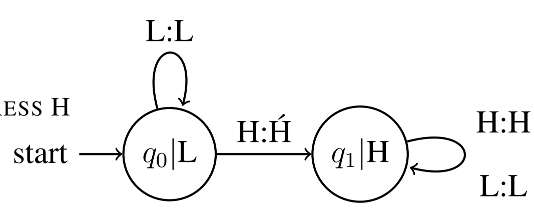
That iterative and long-distance patterns are subject to different computational requirements suggests a conceptual separation of stress typology into different categories with long-distance atoms and categories with only local atoms. I argue that this is a natural choice given the divergent nature of their computational properties – different computational characteristics lead to distinct typological predictions and differing notions of how to constrain overgeneration. If a given formalism treats two patterns with distinct computational properties as the same, then that is a facet of the patterns as they exist in that formalism, not of the abstract properties of the patterns themselves. On the other hand, the complexity of a pattern cannot be examined at all without adopting some representational assumptions, highlighting the fact that studies of computation in linguistics and more substantive proposals with regards to theoretical devices should enjoy a mutually beneficial interchange of insight that improves the quality of work being done in both domains.

6.4 Unary versus higher alphabet size

It was demonstrated above that long-distance input-output maps can be defined whether the input contains two (DTO + DTS), three (Pulaar), or more symbols (Nanti). There is one apparent counterexample to this generalization regarding input alphabet size. Though the difference between two or three symbols is immaterial in terms of what type of functions can be described, the difference between *one* input symbol or more *does* have a direct effect on the kind of input-output maps that can be derived. For functions with an input alphabet of one, true long-distance generalizations cannot be described at all. This is relevant to QI stress patterns, where the input is typically taken to be a string of unstressed syllables, σ^n. To demonstrate why this is the case, consider the STRESS H function from above, repeated here:

(131) STRESS H

a.



This could be considered a canonical example of a long-distance pattern – make a change to the first of a certain kind of input in the string (a heavy syllable), no matter where it occurs. It is a strictly piecewise pattern, as demonstrated above. For such patterns, the presence of multiple input alphabet symbols is precisely what allows such a pattern to be described. This is reflected in the information encoded by each state – if only Ls are seen, the transduction loops indefinitely in q_0. The presence of some H is the signal for stress – the signal to make an alteration to the string, no matter where it occurs, and move to state q_1 where the change is "remembered".

Now consider an attempt to define a similar function with an alphabet size of one, using σ as the example. Such a function is permanently stuck in q_0 – there is no other symbol that can signal the function to make a change whenever it appears, as H does for STRESS H. With an alphabet of at least two, STRESS H can search any string for an H to the exclusion of Ls and place stress on the first H, no matter where it occurs. If the input alphabet is comprised only of σ, however, this type of function cannot be defined:

(132) $\Sigma = $ L,H $\qquad\qquad$ $\Sigma = \sigma$

a. LHLLLL $\mapsto$ LḢLLLL $\qquad$ a. $\sigma\sigma\sigma\sigma\sigma \mapsto \sigma\sigma\sigma\sigma\sigma$

b. LLLLHL $\mapsto$ LLLLḢL $\qquad$ b. $\sigma\sigma\sigma\sigma\sigma \mapsto \sigma\sigma\sigma\sigma\sigma$

c. LLLLLL $\mapsto$ LLLLLL $\qquad$ c. $\sigma\sigma\sigma\sigma\sigma \mapsto \sigma\sigma\sigma\sigma\sigma$

The set of input elements changed by STRESS H is empty in the case that the word is comprised of all Ls or all σ, as in (132c). As soon as an H is introduced, however, the sets are no longer equal – some input H will join the set for the L and H-containing alphabet, but the set of changed elements is *always* empty for the unary σ alphabet.

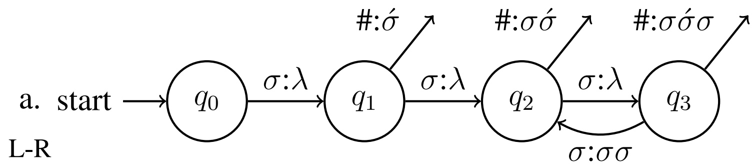
While it is true that long-distance functions such as STRESS H cannot be defined with inputs of just σ, it is not the case that *no* subsequential functions can be defined when the size of the input alphabet is one. An example comes from Creek stress (Haas 1977; Halle and Vergnaud 1987; Hayes 1995). In words of all light syllables, stress falls on whichever of the penult or ultima is even, counting from the left:

(133) a. $\sigma\sigma\sigma\acute{\sigma}$

b. $\sigma\sigma\sigma\acute{\sigma}\sigma$

By waiting and using the right-edge word boundary as a reference point, a single subsequential function can describe the input-output map for Creek stress:

(134)



The function is non-local – the states encode information about parity. Parity is a meta-property of the word, and is unrelated to the input or output substrings of subsequences of the string. Reading left to right, the transducer waits to see the right word boundary. Ending in q_2 indicates an even-parity string, and so $\sigma\acute{\sigma}$ is output when # is encountered. Conversely, ending in q_3 indicates an odd-parity string, and so $\sigma\acute{\sigma}\sigma$ is output instead. A derivation for five- and six-syllable forms is given here:

(135) 5σ

input:	σ	σ	σ	σ	σ	#
states:	$0 \rightarrow 1 \rightarrow$	$2 \rightarrow$	$3 \rightarrow$	$2 \rightarrow$	$3 \rightarrow$	
	λ	λ	λ	$\sigma\sigma$	λ	$\sigma\acute{\sigma}\sigma$
output:	σ	σ	σ	$\acute{\sigma}$	σ	

6σ

input:	σ	σ	σ	σ	σ	σ	#
states:	$0 \rightarrow 1 \rightarrow$	$2 \rightarrow$	$3 \rightarrow$	$2 \rightarrow$	$3 \rightarrow$	$2 \rightarrow$	
	λ	λ	λ	$\sigma\sigma$	λ	$\sigma\sigma$	$\sigma\acute{\sigma}$
output:	σ	σ	σ	σ	σ	$\acute{\sigma}$	

By looping between an "even" and "odd" state and waiting for the end of the word, the left subsequential function derives stress assignment in Creek. Thus, even with an input alphabet size of one, some long-distance functions can be described. However, while subsequential power is required to carry out the pseudo-parity count of the word, it is *not* unbounded in the sense of STRESS H – Creek limits stress to the penult or final syllable, but STRESS H can place stress anywhere in the word.

These factors, taken together, suggest that while a function with an input alphabet of size one can describe input-output mappings that appear long-distance via a pseudo-parity count, they cannot describe truly unbounded patterns such as STRESS H. It may be the case that, for any class of functions, the subset of functions with an alphabet size of one is formally less expressive than the subset of functions with a larger alphabet. This result offers an explanation as to why unbounded stress patterns are apparently always of the quantity sensitive variety – an input alphabet of at least size two, such as H and L, is necessary to describe the kind of long-distance pattern we see in QS stress typology with a single function. It is possible to derive a descriptively long-distance pattern such as "stress the final syllable if the first element in the string is a syllable" with an input of just σ, but such a function will stress the final syllable in every word in the language, and so is extensionally equivalent to a simple "final stress" function, which can be computed locally.

This extends to single stress patterns as well – a generalization such as "stress the penult if heavy, else the antepenult", which is found in Latin, requires an input alphabet of at least heavy and light syllables to define. If the only input symbol is σ, then a heavy penult cannot be distinguished from a heavy antepenult, or from any other position in the word. Thus, the size of the input alphabet can affect the expressivity of patterns that can be defined.

CHAPTER 7: METRICAL STRUCTURE AND COMPUTATION

7.1 Introduction

The metrical foot has a long pedigree as a theoretical device in generative phonology (Liberman and Prince 1977; Halle and Vergnaud 1978; Selkirk 1980; Hammond 1984; Halle and Vergnaud 1987; Idsardi 1992; Hayes 1995). While the motivations for foot structure are typically studied in terms of stress, this chapter provides evidence from the principles of formal language theory (Chomsky 1956; Hopcroft and Ullman 1979) for foot-based analyses of non-stress processes. Though use of foot structure in these analyses is not novel (see González (2018) for an overview) this chapter contributes a precise characterization of what is at stake in terms of the computation of these processes when foot structure is present versus when it is not. This formal computational analysis indicates that feet have measurable implications for the predicted typology of these patterns. Thus, support is provided for a specific substantive phonological proposal based on the well-defined measures of complexity that formal language theory offers.

Specifically, this chapter demonstrates that segmental and morphological alternations which appear to rely on an overt parity count of the syllables in the word are rendered fundamentally local in the presence of foot structure. In Capanahua, for example, coda glottal stops are deleted in even-numbered syllables (Loos 1969). Thinking of such alternations as *functions* that take in an input string and return an output string, the addition of feet ensures that an otherwise long-distance, properly *subsequential* (Mohri 1997) process is computed as the combination of two local functions instead: creation of feet via an *output stricly local* function (OSL) and calculation of the alternation via an *input strictly local* function (ISL) (Chandlee 2014; Chandlee and Heinz 2018). This leads to better typological predictions and highlights that these processes are fundamentally local only if the correct representational assumptions are adopted.

The computational analysis further demonstrates that full subsequential power is too weak of a hypothesis for the class of phonological patterns described here. This is because subsequential functions, which are more expressive than OSL or ISL functions, allow for counting modulo some finite number throughout the word. For example, a function that deletes every even-numbered occurrence of a glottal stop is not a natural phonological generalization and can not be derived input- or output-locally, but it *is* describable with a subsequential function. The hypothesis adopted in this chapter is thus that attested patterns are limited to an OSL plus ISL map, which excludes such pathological generalizations. This is because OSL and ISL functions can only "count" in a way that is formally local. For example, a penultimate stress pattern checks for the structure '$\acute{\sigma}\sigma\#$'. An iterative stress pattern places further stresses based on the location of a previous stress in the output i.e. a local output window. While this kind of non-modulo counting that checks some local span is an inextricable aspect of computation in phonology, the modulo counting allowed by subsequential functions is not. Thus, a non-local theory leads to clearly inaccurate typological predictions.

In sum, the study of computational complexity allows us to identify two formally different types of counting and understand why one is a central part of phonological generalizations and the other leads to a less restrictive typology containing unattested pathological patterns. The explicit analysis of the effect of foot structure in this chapter thus clarifies both what feet tell us about the nature of phonological generalizations and why this is important.

7.2 Counting in phonology

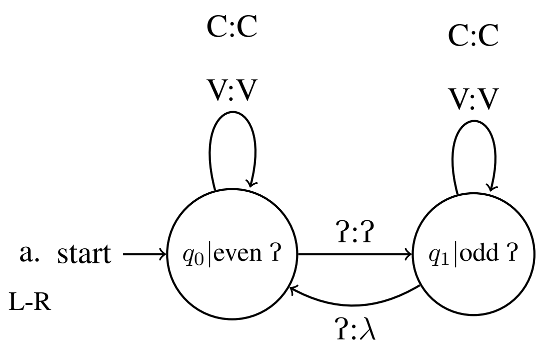
The issue of the extent to which phonological grammar can provide a numerical count has been the subject of much debate in the literature (Kenstowicz 1994; Hayes 1995; Isac and Reiss 2008; Marlo et al. 2015; Paster 2019; Rolle and Lionnet 2020). An intuition shared by many researchers is that the ability to count should be limited in some way. Hayes (1995), for example, offers that, "a reasonable conjecture is that phonological rules can count only to two" (p.307). The study of phonological patterns through the lens of formal language theory offers some crucial insight to this active debate by identifying different *kinds* of counting – one that is necessary and abundant in phonological typology and one that is pathological and should be ruled out. Thus, instead of searching for an arbitrary upper bound on the amount of counting that is allowed, we can state a hypothesis on how phonological counting should be limited based on the formal complexity of the distinct modes of counting. A hypothesis based on this distinction helps eliminate pathological counting while allowing the type of counting that is a necessary aspect of phonology.

Specifically, EO, ISL, and OSL functions all encode a local kind of "counting" by enforcing the presence or absence of certain configurations in the input or output string. A non-finality function, for example, enforces the structure $\sigma\#$ by checking the first symbol preceding the word boundary. A binary iterative stress function requires that stress fall in an alternating pattern by checking the previous symbol in the output – if this symbol is σ, place stress on the following syllable. Note that these are local requirements on the structure of the input or output string. I argue that this sort of counting that is definable with an EO, ISL, or OSL functions and permeates the typology of phonological patterns is not really counting at all. This is because checking a local window of input or output symbols for a certain structure need not rely on a numerical value – just the presence or absence of the structure.

However, as demonstrated in Chapters 4 and 5, moving beyond these local function classes to the more complex subsequential class allows for pathological typological predictions. This is precisely because subsequential functions can carry out an explicit long-distance “modulo” count for the entire length of the string, such as an even/odd parity count. Modulo counting is distinct from the local structural generalizations that are possible with less expressive function classes, as it does allow for overt numerical counting. For example, consider a hypothetical pattern that deletes every even-numbered glottal stop that is found in the word – the pathological counterpart of Capanahua coda-ʔ deletion:

(136)

a.



The function tracks the parity of the glottal stops it encounters, deleting even-numbered instances of the segment. Tracking of parity must span the entire word – deletion could occur in the second syllable and seventh syllable, for example, demonstrating that the pattern cannot be determined by the parity of the syllable. The process is not ISL, because there is no local input structure that can determine the parity of a given glottal stop. The process is also not OSL for analogous reasons – no local output structure can determine how a particular glottal stop should surface, as it may be separated from a previous glottal stop by an, in principle, unbounded number of intervening segments that do not participate in the process. However, the process *is* subsequential – subsequential functions have access to the kind of long distance memory necessary to determine that the current input symbol is an even-numbered glottal stop, no matter where it occurs in the word. It is this ability to count modulo some number that makes the subsequential class a weak hypothesis for the expressive power of phonology.

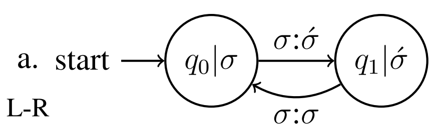
Note that while OSL processes such as iteration of stress could be described as a parity counting pattern i.e. stress every odd syllable, the computational properties of the function tell us that this is an incorrect descriptive characterization of the process. Calculation of iterative stress is carried out by observation of a local window up to the previous stress in the output string, rather than a long-distance count of the parity of each syllable over the duration of the word. The difference between these two kinds of counting matters for theories of phonology – enforcement of local structural restrictions as in ISL or OSL is an integral part of phonological computation, while the long-distance overt parity counting possible with a subsequential function is not. Allowing this kind of explicit counting into phonology leads to a less restrictive typology featuring pathological patterns as in (136), while the limitation to OSL and ISL proposed here excludes such patterns. Thus, the computational analysis here adds clarity to an active debate in the field by formally differentiating pathological and non-pathological modes of counting.

7.3 Feet and computation

The presence of feet affects the formal complexity of phonological processes in a non-trivial way. When foot structure is present in the input string, it provides a local reference point from which processes may be calculated. The first transducer represents an iterative stress function calculated with no feet:

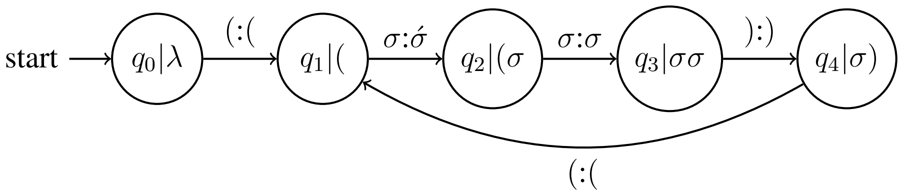
(137)

a.



As demonstrated in earlier chapters, such iterative application of stress is OSL because further application of stress depends on previous, local material in the output. This is seen in the state information – being in q_1 means that an unstressed syllable has just been output, and so the following syllable is output with stress.

However, when foot structure is present in the input, no reference to the output is required. Consider the following transducer:

(138) a.


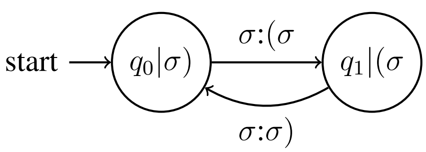
L-R

Stress can be placed correctly whenever a '$(\sigma$' input-sequence is observed, as in the transition from q_1 to q_2. Thus, feet give the input string structure that an ISL function can use to calculate phonological processes locally based on the position of foot boundaries. The following is a derivation for input $(\sigma\sigma)(\sigma\sigma)$:

(139)

input:	(	σ	σ	)	(	σ	σ	)
states:	$q_0 \rightarrow q_1 \rightarrow q_2 \rightarrow q_3 \rightarrow q_4 \rightarrow q_1 \rightarrow q_2 \rightarrow q_3 \rightarrow q_4$							
output:	(	$\acute{\sigma}$	σ	)	(	$\acute{\sigma}$	σ	)

Importantly, creation of feet is itself an iterative OSL process, proceeding in the same way as placement of stress in (138). Consider the following transducer:

(140) a.


L-R

Further foot boundaries are inserted depending on the location of the previous foot boundary in the output. A left boundary is placed when the first syllable is read in the transition from q_0 to q_1. From here, a right boundary is placed after the next syllable in the transition back to q_1. This is just like OSL application of stress in (137), but instead the relevant output feature is the foot boundary. The function in (140) is not ISL – the states correspond to local output information. So, whether a derivation employs iteratively created feet, or a process such as stress placement occurs iteratively by itself, the total map in each case is subject to output locality. The following is a derivation for input $\sigma\sigma\sigma\sigma$:

(141)

input:	σ	σ	σ	σ
states:	$q_0 \rightarrow q_1 \rightarrow q_0 \rightarrow q_1 \rightarrow q_0$			
output:	$(\sigma$	$\sigma)$	$(\sigma$	$\sigma)$

While iterative stress is output local with or without feet, there are cases where the presence or lack of foot structure has measurable implications for locality and the complexity of the function that computes the process. These types of patterns only apply in syllables of a certain parity, such as deletion of coda-ʔ in even numbered syllables in Capanahua (Loos 1969; Safir 1979; González 2009). Unlike iterative stress, which is placed on every syllable in the alternating count of the pattern, processes like in Capanahua apply in any arbitrary syllable of the correct parity (6th, 8th, 10th, etc.) where the structural description of the rule is met. For example, the declarative marker /taʔ/ surfaces as [ta] in the sixth syllable in the following word meaning, 'it's probably not a dog':

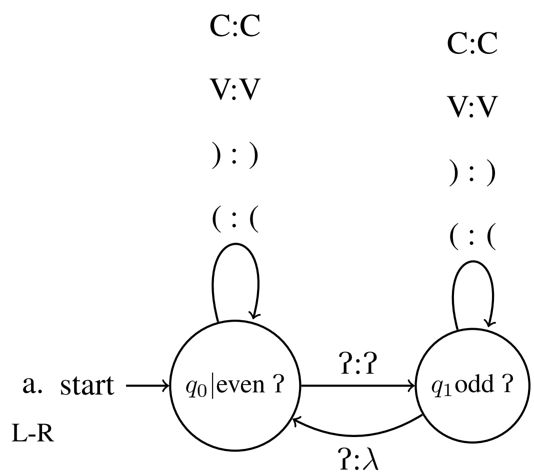
(142) / ʔú.tʃi.ti.ma.raʔ.**taʔ.**ki / $\mapsto$ [ʔú.tʃi.ti.ma.raʔ.**ta.**ki]

This is the first occurrence of coda-ʔ deletion in the word. As a consequence of this, such processes are *not* iterative – there is no local structure in the output (or input) that can determine the alternation alone. Thus, computation of a process like coda-ʔ deletion in Capanahua requires a properly subsequential function to compute.

However, if we assume that processes like coda-ʔ apply to a string where foot structure is present, then the properly subsequential pattern is rendered ISL instead. This is because, like in (138), feet provide an input local reference point that conditions the application of the process without any need for explicit parity counting. Thus, breaking patterns like Capanahua down into an OSL step of foot creation and an ISL step of glottal stop deletion reveals the fundamental locality that the patterns are subject to – a result that is completely obscured when looking at the map as a single holistic step from input to output.

This mirrors the compositional analyses of stress patterns presented in the preceding chapters, follow the same intuition that surface phonological patterns are actually the coalescence of simpler phonological atoms. Here again we see that adopting an atomic approach versus a more complex single-step approach directly affects the predicted typology of phonological patterns. For example, if we do not adopt a stepwise analysis for coda-ʔ deletion in Capanahua, the pattern is properly subsequential. This is an implicit claim that, without any further restrictions, we might expect to find any possible subsequential function in the typology of segmental phenomena. As demonstrated via the pathological pattern in (136), this typological claim is a bad one. Crucially, a pattern that deletes segments of a certain parity cannot be derived by the same breakdown into local functions as proposed here for Capanahua. Consider the following attempt to derive (136) via the application of foot structure:

(143)



The generalization is the same as in the example without feet. The only difference is the addition of foot boundaries to the loops over q_0 and q_1. As the process is no longer tied to the parity of the syllable, foot structure is no longer helpful. The local input structure it provides does not offer the same computational aid as it does in attested patterns like Capanahua. Deleting every odd/even occurrence of a segment does not seem phonological in an intuitive sense, and the measures of formal complexity provided by formal language theory tell us why – it is beyond the expressive power of the hypothesized breakdown of such processes into simpler pieces that are fundamentally local. While it is also ultimately necessary to state a restriction on combinations of local processes such as OSL and ISL, I argue that, in terms of typological predictions and learnability, they provide a better starting hypothesis for these processes that can be tested and refined as more data is observed and research in this vein progresses.

7.4 Analyses

In this section, I provide analyses of patterns from four languages that demonstrate how foot structure serves to render a long-distance, parity counting pattern formally local. These patterns that support the presence of foot structure are notable in that they are segmental and morphological alternations, rather than stress, where the use of feet is typically motivated. For an excellent overview of such patterns, see González (2018).

7.4.1 Huariapano

In the now-extinct Panoan language Huariapano (Parker 1994, 1998), a process of coda /h/ epenthesis occurs in an odd-numbered syllable when the following syllable has a voiceless obstruent onset i.e. when the structure 'V.T' occurs. Consider the following pairs of underlying and surface forms:

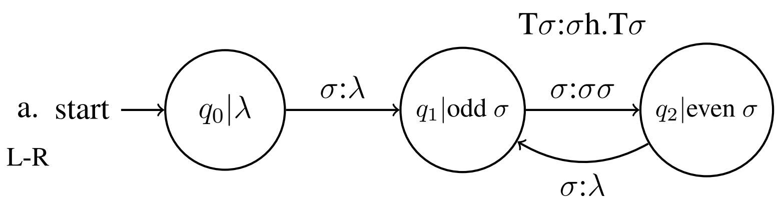
(144)	UR	SR	gloss
	a. / ja.na.pa.kwin /	[jà.na.pah.kwín]	'I will help'
	b./ βo.no.si.kæj /	[βò.no.síh.kæj]	'they will take, carry'
	c./ jo.mɯ.ra.ka.ti.kæj /	[jò.mɯ.ràh.ka.tíh.kæj]	'they hunted'
	d./ paj.ri.rah.kaj /	[pàj.ri.ráh.kaj]	'still'

From the data, we observe that epenthesis only occurs when the following syllable has a voiceless obstruent onset. In (144a) for example, the third syllable surfaces with a coda [h] because of the following [k], but no epenthesis occurs on the first syllable, which is followed by a nasal onset. Epenthesis is also restricted to syllables of odd-parity – in (144c), the fourth syllable is followed by the voiceless obstruent [t], but no epenthesis occurs. Note that the process is also independent of stress – it can occur in syllables with main (144c,d), secondary (144c), or no stress (144a).

Without feet

In the absence of foot structure, h-epenthesis in Huariapano must know the parity of each syllable, starting from the left edge and maintain the count for the duration of the word. This is because, with no other structure to refer to, an explicit parity count is the only method to track where h-epenthesis may or may not apply. Consider the following transducer:

(145)



Starting from the left edge, the function takes in a syllable and outputs λ. This is because application of h-epenthesis relies on further information. If the following syllable is Tσ – a syllable with a voiceless obsruent onset – then σh.Tσ is output, making up for the waiting and outputting an epenthesized h-coda. If a syllable with no obstruent onset is read instead, the output is $\sigma\sigma$ i.e. an identity mapping that makes up for the previous waiting. The next syllable is of odd parity, and so the process resets, returning to q_1. Intuitively, this is the information encoded in each state – q_1 is an "odd syllable" state, and q_2 is an "even syllable" state. This is not local input or output information – parity counting must start at the left edge and be maintained as the word grows, as syllables in the first, third, fifth – and so on – positions are potential targets for h-epenthesis. Thus, the behavior of further input syllables is affected by information all the way at the opposite edge. Crucially, it is also not OSL because it is not iterative – h-epenthesis may apply for the first time in any arbitrary odd syllable where the structural description of the rule is met. If it applied consistently to *every* odd syllable, then this would provide local output information for further propagation of the process, but this is not the case. Instead, a footless analysis of Huariapano h-epenthesis requires a properly subsequential function, which can track long-distance information such as parity. The following is a derivation for / ja.na.pa.kwin / 'I will help', from (144a):

(146)

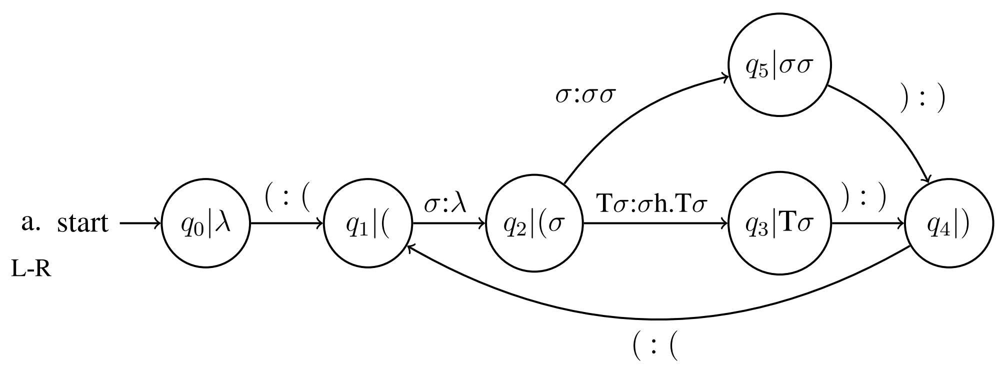
input:		ja.		na.		pa.		kwin.		
states:	q_0	$\rightarrow$	q_1	$\rightarrow$	q_2	$\rightarrow$	q_1	$\rightarrow$	q_2	
output:		λ		ja.na.		λ		pah.kwin		

The function checks for a voiceless obstruent following every odd syllable. This occurs in the third syllable, leading to surface [pah] for underlying /pa/. Though the first syllable is always odd, the following onset is a nasal, and so h-epenthesis does not apply. It also does not apply in even syllables, such as the second syllable here, even though it is followed by a voiceless obstruent.

With feet

For patterns like Huariapano that are subsequential as a mapping between bare strings of segments, the inclusion of feet serves to make the patterns ISL. This is because feet provide a reference point that is local in the input string which a phonological process can use to check if it should apply. Consider the transducer here in (147), which crucially assumes that the word has been parsed into feet by an OSL foot placement process, as described in §2.4:

(147)



Reading the string left to right, a left foot boundary informs the function that the following syllable is odd, *without* relying on an explicit count from the beginning of the word. The next syllable is output as λ. From here, if an onset voiceless obstruent is seen, a coda-h is epenthesized in the transition from q_2 to q_3. If no voiceless obstruent is present, $\sigma\sigma$ is output in the transition to q_5 instead. Reading of a right foot boundary takes the transduction to q_4, from where a left foot boundary returns to q_1. Thus, instead of relying on the parity of the syllables, the function relies on the input-local span of two syllables following a left foot boundary. The following is an example derivation for input (ja.na.)(pa.kwin):

(148)

input:		(		ja.		na.		)		(		pa.		kwin.				)
states:	q_0	$\rightarrow$	q_1	$\rightarrow$	q_2	$\rightarrow$	q_5	$\rightarrow$	q_4	$\rightarrow$	q_1	$\rightarrow$	q_2	$\rightarrow$	q_3	$\rightarrow$	q_4	
output:		(		λ		ja.na		)		(		λ		pah.kwin.				)

Huariapano h-epenthesis modulo foot structure is input local, as its application is conditioned entirely by specific $(\sigma\sigma$ sequences in the input string. The presence of feet reduces a non-local, subsequential process to an ISL one.

7.4.2 Capanahua

In Capanahua, a process of coda glottal stop deletion occurs in even numbered syllables, counting from the left (Loos 1969; Safir 1979; González 2009). Consider the alternation of the declarative suffix /taʔ/ and morpheme /raʔ/ 'maybe' in the data below, written in bold for clarity:

(149)

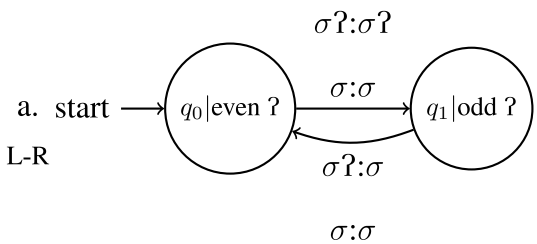
UR	SR	gloss
a. / ho.no.**taʔ**.ki /	[hó.no.**taʔ**.ki]	'It is a wild pig'
b. / ho.no.ma.**taʔ**.ki /	[hó.no.ma.**ta**.ki]	'It is not a wild pig'
c. / ʔu.tʃi.ti.**raʔ**.**taʔ**.ki /	[ʔú.tʃi.ti.**ra**.**taʔ**.ki]	'It's probably a dog'
d. / ʔu.tʃi.ti.ma.**raʔ**.**taʔ**.ki /	[ʔú.tʃi.ti.ma.**raʔ**.**ta**.ki]	'It's probably not a dog'

In (149a), the declarative suffix /taʔ/ occurs in an odd syllable, and so it surfaces with its underlying glottal coda. In (149b), however, the addition of the negation affix *-ma-* pushes /taʔ/ into an even-numbered syllable position, and so the coda glottal stop is deleted. The same pattern is observed with both /taʔ/ and /raʔ/ in (149c-d). The addition of the negation affix in (149d) pushes the two into a position with different syllable parity, and so the behavior of the coda glottal stop changes. It should also be noted that this process is divorced from stress assignment – stress in Capanahua is bound to a two-syllable window at the beginning of the word, and no secondary stresses are reported to occur.

Without feet

As in Huariapano, calculation of coda glottal stop deletion in Capanahua without reference to foot structure requires an explicit parity count of all the syllables in the word. Starting from the left edge, the function checks for the configuration ‘σʔ’ – a syllable with a glottal stop coda – in every even syllable, and deletes the glottal stop when it is found in that position. This is seen in the following transducer for coda-ʔ deletion that makes no reference to foot structure:

(150)



Glottal stop codas are preserved in syllables of odd parity – the transition to q_1 from q_0 outputs the input glottal stops faithfully. From q_1, however, coda glottal stops are deleted in the transition back to q_0. Once again parity must be tracked through the entire word, as the process can apply at any point when a glottal coda occurs in an even numbered syllable. Like Huariapano, this means it is not ISL or OSL. Instead, calculation of coda-ʔ deletion in Capanahua with no foot structure requires the power of a subsequential function. The following is a derivation for / ho.no.ma.taʔ.ki / ‘it is not a wild pig’, from (149b):

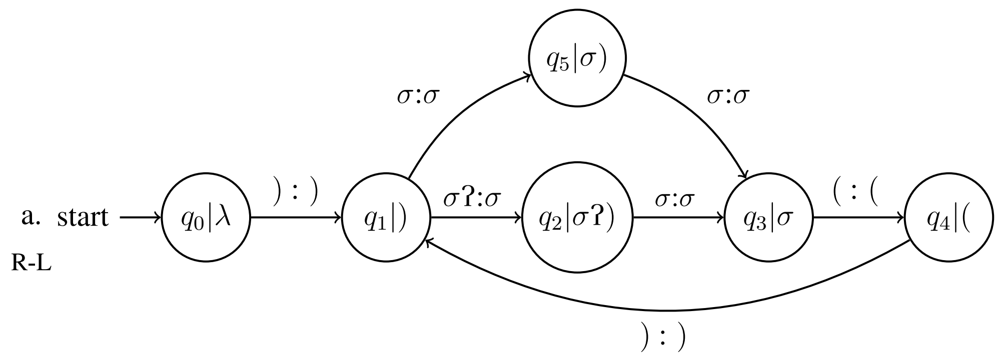
(151)

input:		ho.		no.		ma.		taʔ.		ki.	
states:	q_0	$\rightarrow$	q_1	$\rightarrow$	q_0	$\rightarrow$	q_1	$\rightarrow$	q_0	$\rightarrow$	q_1
output:		ho.		no.		ma.		ta.		ki.	

With feet

In Capanahua, if feet are constructed first before coda deletion applies, then the presence of foot structure in the input renders the process ISL. This is because it again removes the need for an explicit parity count – instead of checking for coda-ʔ in even syllables, it checks for a coda-ʔ that immediately precedes a right foot boundary. The transducer applies right to left:

(152)



Reading through the string, the key structure in the input is 'σʔ)'. When this structure is encountered, the glottal stop is removed, as seen in the transition from q_1 to q_2. Otherwise, inputs are output faithfully. Just as in Huariapano, the insertion of foot structure renders the otherwise subsequential process of coda-ʔ deletion in Capanahua ISL. The following is a derivation for input (ʔu.tʃi.)(ti.raʔ.)(taʔ.ki):

(153)

input:		(		ʔu.		tʃi.		)		(		ti.		raʔ.		)		(		taʔ.		ki.		)
states:	q_4	$\leftarrow$	q_3	$\leftarrow$	q_5	$\leftarrow$	q_1	$\leftarrow$	q_4	$\leftarrow$	q_3	$\leftarrow$	q_2	$\leftarrow$	q_1	$\leftarrow$	q_4	$\leftarrow$	q_3	$\leftarrow$	q_5	$\leftarrow$	q_1	$\leftarrow$ q_0
output:		(		ʔu.		tʃi.		)		(		ti.		ra.		)		(		taʔ.		ki.		)

Reading the string right to left, the function derives the correct outputs for Capanahua words with regards to coda-ʔ deletion. Though the fifth syllable 'taʔ' contains a glottal coda, its location within the foot ensures that it is preserved. In the fourth syllable, however, 'raʔ' falls next to a right foot boundary, and so the coda is deleted. Otherwise, the word is output faithfully. Foot structure thus allows the alternation to be determined using information that is local in the input – it is ISL, rather than subsequential.

7.4.3 Shipibo

In Shipibo, the emphatic suffix /-riβ/ surfaces variously as [-ri.ba] or [-ri.bi], depending on how many syllables precede it (Lauriault 1948; Elias-Ulloa 2006; González 2009). If an odd number of syllables precede the suffix, the allomorph [-ri.ba] appears. After an even number of syllables, [-ri.bi] is selected instead:

(154)

	UR	SR	gloss
	a. / a.**riβ**.kɯ /	[a.**ri.ba**.kɯ]	'did it again'
	b. / a.ma.**riβ**.kɯ /	[a.ma.**ri.bi**.kɯ]	'made (him) do it again'
	c. / pu.ta.ma.**riβ**.kɯ /	[pu.ta.ma.**ri.ba**.kɯ]	'made (him) throw again'
	d. / pu.ta.ya.ma.**riβ**.kɯ /	[pu.ta.ya.ma.**ri.bi**.kɯ]	'did not throw again'

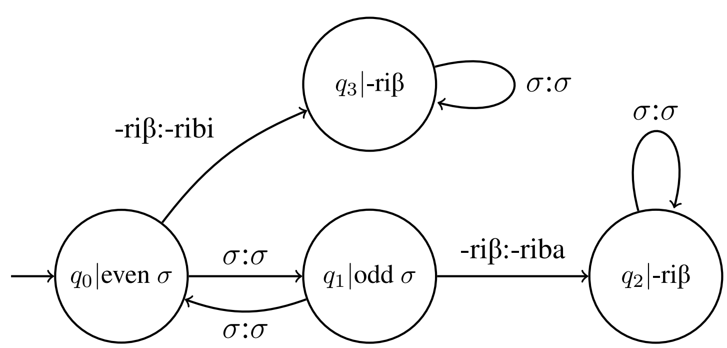
In (154a, c), the emphatic suffix is separated from the beginning of the word by an odd number of syllables, and so it surfaces as [-ri.ba]. In (154b, d), an even number of syllables intervene instead, and so the allomorph [-ri.bi] appears.

Without feet

As in Capanahua and Huaripano, alternation of /-riβ/ in Shipibo can only be determined by an explicit parity count of the syllables in the word. Starting from the left edge, the process counts syllables until the morpheme is found, at which point the current parity value determines which allomorph to select. Consider the examples of (154c,d) above:

(155)

a. L-R



The transduction alternates between states q_0 and q_1, depending on the parity of the syllable. q_1 corresponds to an odd syllable, and q_0 corresponds to an even syllable. If the suffix / -riβ/ is encountered from q_0, this indicates a preceding even-parity string of syllables, and so the [-ribi] allomorph is selected. If the suffix is encoutered from q_1, the [-riba] allomorph is selected instead. With no feet, determining the alternation requires such a properly subsequential function to track the parity of the syllables leading up to the suffix. Nothing local to the input or output can determine the alternation. The following is a derivation for input / a.ma.-riβ.kɯ/ 'made (him) do it again', from (154b):

(156)

input:		a.		ma.		-riβ.		kɯ.	
states:	q_0	$\rightarrow$	q_1	$\rightarrow$	q_0	$\rightarrow$	q_3	$\rightarrow$	q_3
output:		a.		ma.		ri.bi.		kɯ.	

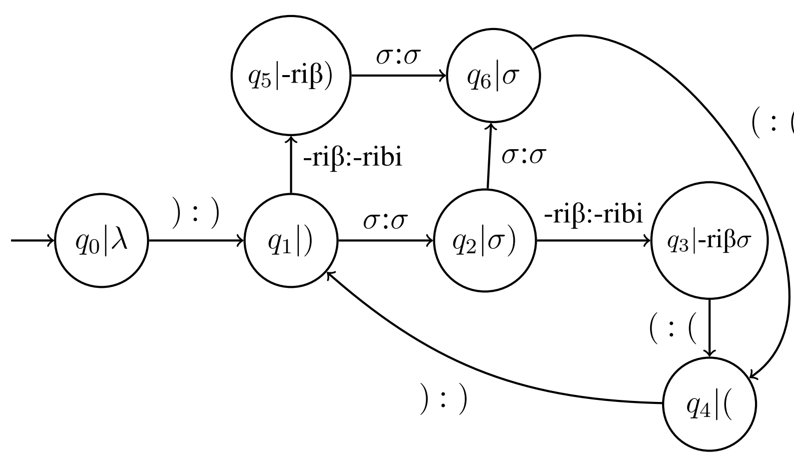
With feet

If foot structure has first been placed into the word, then the pattern of Shipibo is input local. Instead of tracking parity, the alternation of the /-riβ/ morpheme is determined instead by its location relative to the right foot boundary. If '-riβ.)' is seen in the input, this indicates that an odd number of syllables preceded the suffix, and so [-ri.ba] surfaces. Otherwise, [-ri.bi] surfaces. The following transducers demonstrate the ISL nature of the function modulo the presence of foot structure:

(157)

a. start

R-L



Reading the string right to left, if /-riβ/ is encountered next to a right foot boundary, the allomorph [-ribi] is output in the transition from q_1 to q_5. If the suffix occurs after the input sequence 'σ)' instead, the allomorph [-riba] is selected instead. The function otherwise leaves the string unchanged. Thus, by providing local input structure, feet again help derive a pseudo-parity count in a way that is fundamentally local, rendering the properly subsequential alternation in Shipibo ISL. The following example shows a derivation for input (a.ma.)(-riβ.kɯ):

(158)

input:		(	a.	ma.	)	(	-riβ.	kɯ.	)	
states:	q_4 ←	q_6 ←	q_2 ←	q_1 ←	q_4 ←	q_3 ←	q_2 ←	q_1 ←	q_0	
output:		(	a.	ma.	)	(	-ri.bi	kɯ.	)	

7.4.4 Urarina

In Urarina, an isolate spoken in Peru, the nominal plural suffix /-ʉ.rʉ/ surfaces either as [-ʉ.rʉ] or [-kʉ.rʉ], depending on the number of preceding morae (Cajas Rojas et al. 1987; Olawsky 2006; González 2011). After an even number of morae, [-ʉ.rʉ] is the form that surfaces. After an odd number of morae, [-kʉ.rʉ] surfaces instead:

(159)

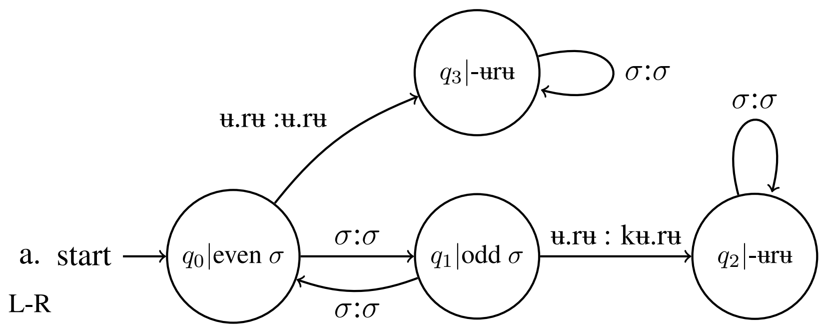
UR	SR	gloss
a. / le.ra.no.**-ʉ.rʉ** /	[le.ra.no.**-kʉ́.rʉ**]	'fishes'
b. / ha.re.hee.ri.**-ʉ.rʉ** /	[ha.re.hée.ri.**-kʉ.rʉ**]	'thin person'
c. / ka.la.mi.na.**-ʉ.rʉ** /	[ka.la.mi.ná.**-ʉ.rʉ**]	'zinc roofs'
d. / ka.tʃa.**-ʉ.rʉ** /	[ka.tʃa.**-ʉ́.rʉ**]	'men'

In (159a,b), an odd number of morae precede the nominal plural suffix – three in (a) and five (b) – and so it surfaces as [-kʉ.rʉ]. In (159c,d), an even number of morae precede the morpheme instead – four in (c) and two in (d) – and so the allomorph [-ʉ.rʉ] appears instead. The allomorphy occurs independent of main stress, and no secondary stress is reported in the language.

Without feet

As with the other patterns described above, Urarina requires a long-distance parity count of the syllables in the word to determine the alternation observed in the data if no foot structure is present. Counting from the left, the correct allomorph is selected based on the parity of the syllable that occurs before the morpheme. Consider the following transducer:

(160)

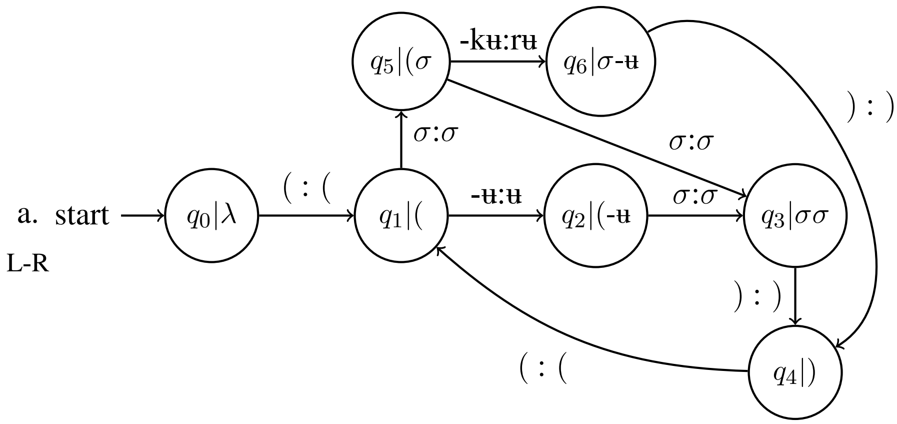


The function mirrors the footless analysis of Shipibo in (155). State q_0 represents an even span of preceding syllables, and so if the suffix /-uʉ.rʉ/ is encountered from here, the allomorph [ʉ.rʉ] is selected. If the suffix is encountered from q_1 instead, then [kʉ.rʉ] surfaces instead, as q_1 represents a preceding odd-parity string of syllables. With no feet, this requires a properly subsequential function to calculate, and is neither ISL nor OSL for the same reasons as in Shipibo.

With feet

With foot structure, however, the pattern is describable with an ISL function. If the sequence '-ʉ.)rʉ' is seen in the input, this indicates that an odd number of syllables preceded the suffix, leading to surface [kʉ.rʉ]. If no foot boundary immediately follows the first syllable of the suffix, then the preceding string of syllables was of even parity, leading to [ʉ.rʉ] instead. The following transducer provides an ISL reanalysis of (160) with feet:

(161)



From q_1, if the nominal plural suffix is encountered, it is immediately adjacent to a left foot boundary, and so the [-ʉ.rʉ] allomorph is selected. If instead a '(σ' sequence is read before the suffix, the preceding sequence of syllables is odd, and so the [-kʉ.rʉ] allomorph is chosen instead. The string is otherwise left unchanged. As in the other analyses discussed above, the presence of foot structure makes a properly subsequential process ISL. The following example shows a derivation of input (ka.tʃa.-ʉ.rʉ.) 'men', from (159d):

(162)

input:		(	ka.	tʃa.	)	(	-ʉ.	rɯ.	)	
states:	q_0 →	q_1 →	q_5 →	q_3 →	q_4 →	q_1 →	q_2 →	q_3 →		q_4
output:		(	ka.	tʃa.	)	(	-ʉ.	rɯ.	)	

7.5 Discussion

In the preceding section, four analyses were presented that show how foot structure serves to make a long-distance parity-counting process formally local. Specifically, the analyses demonstrate the exact computational effect of feet: making a process that requires a properly subsequential function with parity counting into one that requires only an ISL function. The key difference is in the type of counting that occurs – OSL and ISL counting is inseparable from computation in phonology. In fact, I argue that searching for local input or output structures, such as '$\sigma\#$' in a non-finality function, is not really counting at all – OSL and ISL functions merely enforce local structural restrictions on the string.

On the other hand, the modulo counting available to properly subsequential functions results in non-phonological typological predictions related to parity or other explicit counts of symbols in the word.

Though some properly subsequential patterns can be described by the composition of an OSL and ISL function, limitation to OSL and ISL *does* exclude patterns such as the one described above that deletes every even-numbered glottal stop. So, while a restriction on compositions of OSL and ISL processes that provides the most accurate predicted typology remains to be stated, the adherence to local functions provides a better initial hypothesis than full subsequential power. Note that as the ISL portion of the combined maps described above can apply at any point in the word, they cannot be stated using the "edge-oriented" functions described in Chapter 3. As such, other restrictions with regards to the use of markup or the properties of the functions as formal language sets may provide a better avenue for future research on restricted OSL plus ISL compositions.

It should also be pointed out that proposing a foot-based analyses of these non-stress processes, especially where this creates a tier distinct from what is necessary for the stress pattern of the language, is a matter of contention (Parker 1994; González 2007; Bennett 2013). I have demonstrated that there is good evidence from formal complexity that a foot-based analysis is indeed appropriate as it allows for a more restrictive theory involving the interaction of simple constraints. Additionally, I suggest that if the foot is a device that languages do employ, there is no *a priori* reason to think that other types of processes do not have access to feet as well, especially if the specific language employs feet for stress.

Some previous work treats parity counting as a natural property of phonology for which feet are useful to represent. McCarthy (1979) in an analysis of Arabic stress, for example, describes parity counting in stress assignment as "reasonably familiar" (p.448) and "stipulated by" the presence of feet (p.451). Hayes (1995), referring to the same pattern, states that the count of syllables is "carried out by the bimoraic foot structure" (p.70). However, the explicit computational analysis in this chapter indicates that parity counting is not a property of phonological patterns *at all*. Placement of feet allows for reference to local input structure to compute phonological processes, completely obviating any need for explicit parity counting. The argument from formal complexity also tells us why this is a desirable result – adopting properly subsequential power as a hypothesis for phonology makes worse typological predictions than a hypothesis that instead adheres to input and/or output locality. Thus, the preceding analysis demonstrates not only precisely what is at stake in terms of computation when feet are present, but why this matters for substantive proposals in phonological theory relating to foot structure.

CHAPTER 8: DISCUSSION AND CONCLUSION

8.1 Discussion

8.1.1 Function composition and complexity

This thesis presented a theory of stress and other phenomena as the composition of its constituent parts or "atoms". These atoms are functions that correspond to basic stress generalizations seen in the literature as non-finality, clash, lapse, the basic iteration of stress, and long-distance generalizations. Relying on the well-defined notion of complexity provided by formal language theory, this work proposes a restrictive theory of the atoms of stress patterns that helps explain not just what the surface patterns are, but why they take the form that they do. For example, while most phonologists agree that "stress the middle" is a pathological stress map that we should not predict, studies of formal complexity tell us exactly why this is the case – a pattern that stresses the middle syllable in the word requires a function that is more expressive than those proposed for the atoms of stress typology. Specifically, unidirectional iterative stress patterns were shown to be the composition of an OSL iteration function and a small number of EO "cleanup" functions that make changes when OSL iteration is not enough to capture the particular pattern alone. Bidirectional patterns require a pair of contradirectional OSL functions, while long-distance patterns require some combination of EO and SP functions.

The class of EO functions was motivated partially by the fact that the cleanup functions necessary for stress patterns are both OSL and ISL. As such, ISL does not provide the appropriate level of restrictiveness for compositions of stress atoms. OSL is also not a good candidate for the cleanup functions, as compositions of OSL plus OSL functions can derive pathological patterns such as sour-grapes stress. As EO functions are restricted to apply only within a bounded window at a word edge, they provide a restriction based on computational properties that encodes a typologically real aspect of stress generalizations – they are tethered to a word edge. For example, non-finality is a well-known aspect of stress patterns, but "non-middle-syllable" is not.

This atomic approach is preferable to a single-function map because it highlights the complexity of the individual atoms and demonstrates that various patterns with different surface characteristics – such as being QI versus QS, or being sensitive to a binary versus ternary weight scale – share fundamental computational properties. For local and long-distance patterns alike, it was demonstrated that allowing atoms of full subsequential power leads to unwanted typological predictions based on the ability of subsequential functions to provide explicit parity counting. In other words, subsequentiality is too *weak* of a hypothesis for the atoms of natural language stress generalizations. The limitation to EO, ISL, OSL, and SP functions provides a more restrictive hypothesis that can be continually tested and reevaluated as more patterns are discovered and described.

To form the most appropriately restrictive characterization of stress patterns, further work is needed. Function composition is a powerful operation, and the ability to overgenerate with compositions of form described in this work has not been explored. Nevertheless, by identifying a more restrictive class of possible phonological atoms, an identifiable set of bad typological predictions have been eliminated. This is because, whatever ability simpler function classes have to overgenerate, more complex classes add their own pathological predictions on top of this.

Conversely, it is also true that an explanation purely from computation will not eliminate the full range of undesirable patterns. For example, no QI language places stress on every syllable in the word, even though this can be done with a simple ISL function. So, in areas where an appeal to computation fails, explanations that make reference to substantive properties of specific phonological formalisms can and should be pursued. Studies of phonology from the lens of complexity and computation and studies of phonology that are more couched in a specific phonological formalism should thus enjoy a productive, mutually beneficial relationship.

8.1.2 Representation

In Chapter 6, it was demonstrated that adopting specific representational assumptions can have a direct effect on the complexity of the pattern being described. Specifically, it was shown that the addition of feet to the representation enables certain segmental and morphophonological alternations to be calculated with an ISL function. In the absence of foot structure, these same processes require properly subsequential power, as they appear to rely on an explicit parity count. However, foot structure obviates the need for an overt parity count by serving as local input material that the segmental patterns can make use of. This then constitutes an argument for a specific phonological device – metrical structure – from the perspective of formal language complexity.

Just as in the preceding analyses of stress, a compositional view of these patterns allows for lower expressive power of the individual functions in the map. Avoiding the use of properly subsequential functions is desirable for the same reasons just mentioned above – it eliminates unwanted pathological patterns from the predicted typology of segmental and morphological alternations. More specifically, it eliminates the kind of modulo-counting that is directly responsible for pathological patterns such as “delete every even glottal stop”. A restriction to simpler atoms ensures that only the sort of “counting” that we do observe in phonology is allowed. This amounts to checking a small window in the input or output string for the relevant structure that determines the behavior of the function. In other words, it is not really counting at all.

8.2 Directions for future work

8.2.1 Levels of stress

The distinction between primary and secondary stress was ignored in this dissertation. This is only to communicate the main points with as much clarity as possible – it is not a rejection of levels of stress or theories that refer to them. Primary stress can be added to any analysis above as the application of a separate, primary stress function. This is like the promotion of grid marks or foot-level stress to a higher level in canonical analyses of stress. As it only promotes existing secondary stresses, it enforces a version of the Continuous Column Constraint (Prince 1983; Hayes 1995). Like the "cleanup" functions instantiating the phonological properties of clash, lapse, and non-finality, the primary stress function is also an EO function. This is because in almost all known iterative patterns, no secondary stresses intervene between the main stress and the closest word edge. Possible exceptions are Banawá (Buller et al. 1993) and Paumari (Everett 2003), where the primary stress is the penultimate stress in the word. However, whether a primary stress function must locate the last secondary stress or the second-to-last, it is still an EO function. To demonstrate, suppose the transducers for Pintupi from Chapter 4 applied secondary stress instead. The resulting input-output map would be as follows:

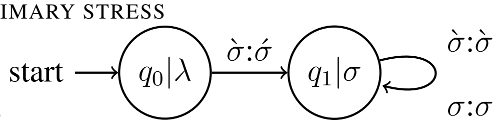
(163) a. $\sigma\sigma\sigma\sigma\sigma \mapsto \grave{\sigma}\sigma\grave{\sigma}\sigma\grave{\sigma} \mapsto \grave{\sigma}\sigma\grave{\sigma}\sigma\sigma$

b. $\sigma\sigma\sigma\sigma\sigma\sigma \mapsto \grave{\sigma}\sigma\grave{\sigma}\sigma\grave{\sigma}\sigma \mapsto \grave{\sigma}\sigma\grave{\sigma}\sigma\grave{\sigma}\sigma$

Now, the PRIMARY STRESS function can promote one of the existing secondary stresses. In Pintupi, the first syllable bears main stress, and so it is targeted by the function for promotion. The transducer below represents this function:

(164) PRIMARY STRESS

a.



L-R

Applying to the final output from (163), the function promotes the secondary stress of the initial syllable to main stress status:

(165) a. $\grave{\sigma}\sigma\grave{\sigma}\sigma\sigma \mapsto \acute{\sigma}\sigma\grave{\sigma}\sigma\sigma$

b. $\grave{\sigma}\sigma\grave{\sigma}\sigma\grave{\sigma}\sigma \mapsto \acute{\sigma}\sigma\grave{\sigma}\sigma\grave{\sigma}\sigma$

It is apparent that (164) is identical to the other EO cleanup functions presented throughout the thesis – a span of a single syllable leads to an identity state from which no further changes are made. The only string leading to a state with a non-identity transition is the empty string λ. As such, the function adheres to the same restriction to OSL plus EO. A similar function can be added to any of the analyses presented above to provide a complete picture that includes different levels of stress.

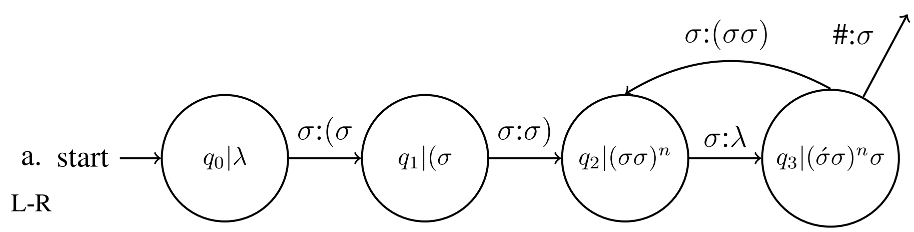
8.2.2 Bidirectional patterns and function composition

Restrictions on compositions of heterodirectional subsequential functions is a topic of recent debate (Heinz and Lai 2013; McCollum et al. 2020). It is argued that such compositions describe the *weakly deterministic* class of functions, a properly subregular function class. The restrictions relate to the ability of the latter function to utilize markup from the first function, in a manner that may be related to the restriction of OSL plus EO in the compositions seen here. Notably, the two OSL functions that comprise the combined map for Auca and Cahuilla in Chapter 5 do not interact – as both functions only apply iterative stress on one side of a morphological marker, they have no chance to interact in a way that alters the eventual output. The order in which they apply makes no difference. On the other hand, the OSL plus EO compositions proposed for unidirectional iterative patterns do interact – the NON-FINALITY function must apply after ITERATION, for example, to remove the unwanted final stress. The relationship of directionality, interaction, and complexity with regards to bidirectional patterns and its similarity to the computational properties of unidirectional patterns is thus a direction for future research.

8.2.3 Foot structure

Though the analyses of stress in Chapters 4 and 5 make no reference to feet or metrical structure, Chapter 6 demonstrated that stress assignment could be broken down into an OSL step of foot creation and a following ISL step of stress placement. However, an analysis with feet does not remove the need for subsequential power that arises when viewing, for example, non-finality or clash patterns as a single function. This is because, instead of waiting to see if the word is ending to avoid final stress – as in Pintupi – a function involving feet now must wait to see if it should place a left foot boundary or not. If the word ends, no boundary should be placed and the final syllable will remain unfooted. If the word continues, a left foot boundary can be created. The transducer is as follows:

(166)



So, although feet do parse words into local constituents, a single-function analysis of such iterative patterns using feet still masks the meaningful, local atoms present in a stress pattern. An example derivation for a five and six syllable input is given here:

(167) 5σ

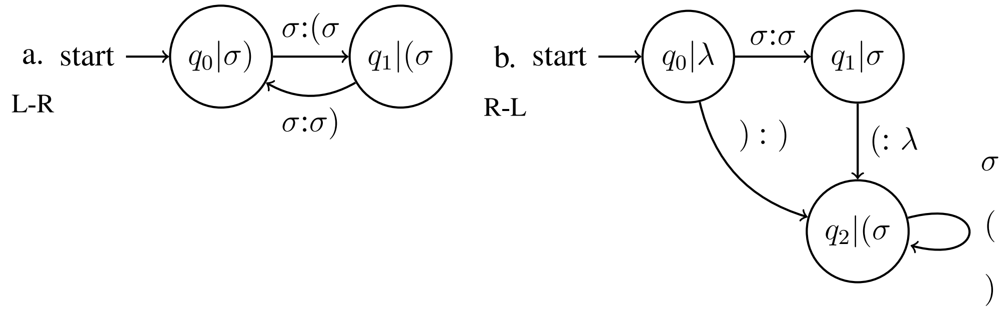
input:		σ	σ	σ	σ	σ	#
states:	q_0	$\rightarrow q_1$	$\rightarrow q_2$	$\rightarrow q_3$	$\rightarrow$	$q_2 \rightarrow$	$q_3 \rightarrow$
output:		(σ	σ)	λ	($\sigma\sigma$)	λ	σ

6σ

input:		σ	σ	σ	σ	σ	σ
states:	q_0	$\rightarrow q_1$	$\rightarrow q_2$	$\rightarrow q_3$	$\rightarrow$	$q_2 \rightarrow$	$q_3 \rightarrow$ q_2
output:		(σ	σ)	λ	($\sigma\sigma$)	λ	($\sigma\sigma$)

Just as with compositions of iterative patterns that appear subsequential as a single function, creation of feet in these cases could be broken down into an OSL function that blindly applies foot structure to the extent that it can and an EO function that erases stray foot boundaries created by the OSL function:

(168)



The first function is the OSL applicator of feet, as seen in Chapter 6. As a result of this function, any string with an odd number of syllables will end in the substring (σ, with an unclosed left foot boundary. This is because the OSL function cannot see ahead to know if the word is ending, and so it cannot place feet appropriately in a non-finality pattern, such as Pintupi. To amend this, the second function applies right to left, removing the errant foot boundary when the structure (σ is seen at the right word egde. Otherwise, the string is left unchanged. This second function is EO – σ is the only substring that reaches a non-identity transition. The compositional analysis for placement of feet thus mirrors the analysis for placement of stress in patterns such as Pintupi presented in Chapter 4. In both cases, the atomic analysis of the process allows for the pattern to be described without reference to properly subsequential power, which I have argued has a desirable effect on the restrictiveness of the theory. An example input-output map for a five and six syllable word are given here:

		(168a)		(168b)	
(169)	a. $\sigma\sigma\sigma\sigma\sigma$	$\mapsto$	$(\sigma\sigma)(\sigma\sigma)(\sigma$	$\mapsto$	$(\sigma\sigma)(\sigma\sigma)\sigma$
	b. $\sigma\sigma\sigma\sigma\sigma\sigma$	$\mapsto$	$(\sigma\sigma)(\sigma\sigma)(\sigma\sigma)$	$\mapsto$	$(\sigma\sigma)(\sigma\sigma)(\sigma\sigma)$

8.2.4 Evaluating representations

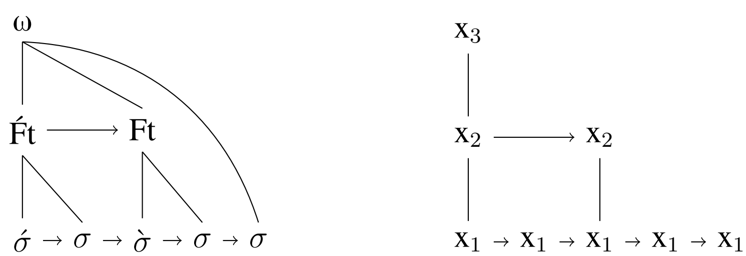
That stress is best represented using hierarchical structure, rather than a flat representation, is one of the key insights from early generative work on stress. From this work, two main theories of representation have emerged – metrical trees and metrical grids (Liberman and Prince 1977; Halle and Vergnaud 1978; Selkirk 1980; Prince 1983; Hammond 1984; Halle and Vergnaud 1987; Idsardi 1992; Hayes 1995). Given two theories of representation, it is expected that one will hold some explanatory advantage over the other, either in its empirical coverage, its restrictiveness, or both. However, it is not enough to assume that this is the case – it must be demonstrated explicitly. If the differences between two models are merely superficial and offer no distinction in restrictiveness or typological predictions, then it can be said that the two are notational variants. This argument is stated succinctly in Chomsky (1972):

> "Given alternative formulations of a theory of grammar, one must first seek to determine how they differ in their empirical consequences, and then try to find ways to compare them in the area of difference. It is easy to be misled into assuming that differently formulated theories actually do differ in empirical consequences, when in fact they are intertranslatable - in a sense, mere notational variants." (p.2)

Work in this vein has demonstrated the intertranslatability of models of syllable structure (Strother-Garcia and Heinz 2017), tone (Oakden 2020), and autosegmental representations (Jardine et al. 2021). There is, however, less work on the question of notational variance in representations of stress. While proponents of grids or trees certainly do discuss the structural differences of the two models (see Prince (1983) in particular), the matter of where and how they differ substantially in their empirical coverage and restrictiveness is still mostly an open question. van Oostendorp (1993) offers a direct comparison of trees and bracketed grids of Halle and Vergnaud (1987), showing that they differ formally under certain assumptions about how the structures may be compared. To my knowledge, there is no such comparison of the formal properties of trees and "pure" i.e. unbracketed grids.

Fromkin (2013) gives two intuitive notions for determining notational equivalence. The first is that the two models will not differ in their empirical predictions. The second is that they represent the same set of abstract properties, differing only superficially. For example, if grids represent the abstract categories of secondary and primary stress, then any notationally equivalent model will have some mechanism or structure to do so as well. Though a full evaluation of expressivity and empirical predictions must also consider the formalism in which the representations are situated, some preliminary observations on the structure of trees and grids suggest that this is an interesting direction for future research. The following diagram shows the representation for the word $\acute{\sigma}\sigma\grave{\sigma}\sigma\sigma$ as a tree (left) and a grid (right). For the tree structure, I assume a *weak layering* analysis (Ito and Mester 1992):

(170)



First, note that all structural nodes present in a tree are also present in a grid – syllables correspond to line one grid marks, foot nodes to line two gridmarks, and the prosodic word node to a line three gridmark. The ordering of these elements is also identical from structure to structure. One difference comes from the horizontal constituency that is imposed by a tree, but absent in a grid. In a tree, syllables without stress are still dominated by some higher node – the foot or prosodic word. In a grid, these syllables are not parsed by elements of a higher tier. However, all association lines present in a grid are also present in a tree. This indicates that given a tree, a corresponding grid is a subgraph of that tree. Thus, a translation between the two structures amounts to adding association lines when translating from a grid to a tree, or deleting them in the reverse case.

Though this structural similarity is certainly striking, it is not the case that *any* grid can be translated into a tree without additional machinery. Consider the following grid for the word $\sigma\acute{\sigma}\sigma\grave{\sigma}\sigma$:

(171)

$$\begin{tikzcd}[column sep=small]
& \mathrm{x}_3 \arrow[d, no head] & & & \\
& \mathrm{x}_2 \arrow[rr] \arrow[d, no head] & & \mathrm{x}_2 \arrow[d, no head] & \\
\mathrm{x}_1 \arrow[r] & \mathrm{x}_1 \arrow[r] & \mathrm{x}_1 \arrow[r] & \mathrm{x}_1 \arrow[r] & \mathrm{x}_1
\end{tikzcd}$$

Such a grid cannot be translated to a tree based on its structure alone because it is *ambiguous* with regards to how level one marks should be parsed into feet. It is consistent with both the iambic parse $(\sigma\acute{\sigma})(\sigma\acute{\sigma})\sigma$ and the trochaic parse $\sigma(\acute{\sigma}\sigma)(\acute{\sigma}\sigma)$, as well as a less conventional $(\sigma\acute{\sigma})\sigma(\acute{\sigma}\sigma)$ parse, reminiscent of "dual stress" patterns (Gordon 2002). To resolve the ambiguity, additional requirements must be imposed on the translation between grids and tree structures. One possibility is to make explicit requirements with regards to directionality – requiring that a foot parses the level one mark that is to the left of the syllable it immediately dominates creates iambs, for example. Parsing the syllable to the right of a stressed syllable creates trochees instead. Crucially, this requirement must be language specific, and is *not* a general method of translation between any tree and grid. This suggests that trees and grids are formally distinct as representations of stress, though further work is needed.

An additional difference comes from how stress is interpreted from a given structure. The stress properties of a string as represented by a grid are recoverable based on the structure alone – the grid mark on line three dominates the line one gridmark corresponding to the main stress syllable. Line two gridmarks dominate a single syllable bearing at least secondary stress. In a tree, however, being dominated by a node from a higher tier is not a guarantee of carrying stress. In fact, *all* syllables are directly or indirectly dominated by the prosodic word. As a result, the stress properties of a string as represented by a tree can only be determined by the *labels* of the nodes in the tree. For example, both feet in (170) are dominated by the prosodic word node, so the main stress foot is distinguished only by the label F́t. All syllables are parsed into feet or the prosodic word, and so the labels that those syllables have indicate their status as bearing primary, secondary, or no stress. This hearkens back to the early work of Liberman and Prince (1977), where binary-branching metrical trees were comprised of nodes labeled s or w for "strong" or "weak" syllables. Thus, grids and trees differ in what aspect of the representation encodes the stresses present in the string – the bare structure for grids, and the individual labels for structural nodes in a tree.

What these factors mean for the notational equivalence or non-equivalence of tree and grid structures is a direction for future research. The first task a formal analysis of the empirical predictions of each representation. While it is obvious that grids and trees differ in a descriptive sense, this is not enough – an exact formal characterization of where and how they differ is needed to establish conclusively that grids and trees are indeed formally distinct modes of representing the hierarchical structure of stress.

8.2.5 Stress shift

This thesis made no reference to stress phenomena such as stress shift (see Kenstowicz (1994)) that operate on underlying, lexical stresses. For example, primary stress is retracted onto a previous secondary stress when another word bearing stress is added, in a kind of clash avoidance. The word "*Mìssissíppi*" bears secondary stress on the first syllable and primary stress on the third syllable. Adding the word "*délta*", with an initial primary stress, causes the primary stress of *Mississippi* to withdraw backwards onto the previous secondary stress, giving "*Míssissìppi délta*". Cast in the terms of this dissertation, stress shift is a function that maps an underlying primary stress back onto a previous secondary stress, deriving surface forms in the following manner, where (172a) corresponds to the *Mississippi* case:

(172) a. $\grave{\sigma}\sigma\acute{\sigma}\sigma \mapsto \acute{\sigma}\sigma\grave{\sigma}\sigma$

b. $\grave{\sigma}\sigma\sigma\acute{\sigma}\sigma \mapsto \acute{\sigma}\sigma\sigma\grave{\sigma}\sigma$

c. $\grave{\sigma}\sigma\sigma\sigma\acute{\sigma}\sigma \mapsto \acute{\sigma}\sigma\sigma\sigma\grave{\sigma}\sigma$

Note that the shift is not local – it may occur over an, in principle, unbounded span of intervening unstressed syllables. Though it is long-distance, such rules of stress shift also fall within the SP boundary. An input secondary stress is output as main stress when the preceding subsequences are σ and $\acute{\sigma}$. When the subsequences are σ, $\acute{\sigma}$, and $\grave{\sigma}$ instead, an input secondary stress is left unchanged and output as $\grave{\sigma}$. It is the presence of another input $\grave{\sigma}$ that determines the different behavior. As any word whose subsequences are a superset of those subsequences will contain $\grave{\sigma}$, any input $\grave{\sigma}$ will be output as $\grave{\sigma}$ as well. Thus, this preliminary analysis suggests that rules of stress shift can be calculated with an SP function, though more work is needed to provide a more complete conception of the computational properties of such patterns.

Bibliography

Bach, E. (1975). Long vowels and stress in Kwakiutl. *Texas Linguistics Forum*, 2:9–19.

Baek, H. (2018). Computatonal representation of unbounded stress. *Proceedings of CLS 53*, pages 13–24.

Bailey, T. (1995). *Nonmetrical constraints on stress*. Doctoral dissertation, University of Minnesota, Minneapolis, Ann Arbor, MI.

Baković, E. (2004). Unbounded stress and factorial typology. In McCarthy, J., editor, *Optimality Theory in Phonology: A Reader*, pages 202–214. Blackwell, London.

Bennett, R. (2013). The uniqueness of metrical structure: rhythmic phonotactics in Huariapano. *Phonology*, 30(3):355–398.

Berry, K. and Berry, C. (1999). *A description of Abun: a West Papuan language of Irian*. Australian National University, Canberra.

Boas, F. and Deloria, E. (1941). Dakota grammar. *Memoirs of the National Academy of Sciences*, 33.

Booij, G. E. (1983). Principles and parameters in prosodic phonology. *Linguistics*, 21(1):249–280.

Boxwell, H. and Boxwell, M. (1966). Weri phonemes. *Papers in New Guinea linguistics*, 5(7):77–93.

Buckley, E. (2009). Locality in metrical typology. *Phonology*, 26:389–435.

Buller, B., Buller, E., and Everett, D. (1993). Stress placement, syllable structure, and minimality in banawá. *International Journal of American Linguistics*, 59:280–293.

Burness, P. and McMullin, K. (2020). Modeling non-local maps as Strictly Piecewise functions. *Proceedings of the Society for Computation in Linguistics*, 3.

Cajas Rojas, J., Corbera, A., Gualdieri, B., and Solís, G. (1987). *Bibliografía etnolingüística Urarina*. Universidad Nacional Mayor de San Marcos, Lima: Instituto de Lingüística Aplicada.

Chandlee, J. (2014). *Strictly Local Phonological Processes*. PhD thesis, University of Delaware.

Chandlee, J., Eyraud, R., and Heinz, J. (2015). Output strictly local functions. *Mathematics of Language 2015*.

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

Chitoran, I. (1996). Prominence vs. rhythm: The predictability of stress in romanian. In Zagona, K., editor, *Grammatical Theory and Romance Languages*, pages 47–58. Amsterdam/Philadelphia: John Benjamins.

Chomsky, N. (1956). Three models for the description of language. *IRE Transactions on Information Theory*, 2(3):113–124.

Chomsky, N. (1972). Studies on semantics in generative grammar. 107.

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

Cowan, H. (1965). *Grammar of the Sentani Language*. Part 47 of Verhandelingen van het Koninklijk Instituut voor Taal- Land- en Volkenkunde.

Crowhurst, M. and Michael, L. (2005). Footing and prominence-driven stress in Nanti (Kampa). *Language*, 81(1):47–95.

Dayley, J. (1989). Tumpisa (Panamint) Shoshone Grammar. *University of California Publications in Linguistics*, 115.

Dixon, R. M. (1983). Nyawaygi. In Robert M.W. Dixon, B. J. B., editor, *Handbook of Australian Languages, Vol. 2*, pages 430–531. Amsterdam: John Benjamins.

Dolatian, Hossep, N. K., Rawski, J., and Garcia, K. S. (2021). Computational restrictions on iterative prosodic processes. In *Supplemental Proceedings of AMP 2020*.

Donaldson, B. C. (1993). *A grammar of Afrikaans*. Mouton, New York.

Dresher, E. and Kaye, J. (1990). A computational learning model for metrical phonology. *Cognition*, 34(2):137–195.

Dresher, E. and Lahiri, A. (1991). The Germanic foot: metrical coherence in Old English. *Linguistic Inquiry*, 22:251–286.

Dubert, R. and Dubert, M. (1973). Biangai phonemes. phonologies of three languages of papua new guinea. *Ukarumpa, Papua New Guinea*, 5(36).

Echeverria, M. and Contreras, H. (1965). Araucanian phonemics. *International Journal of American Linguistics*, 31:132–135.

Eisner, J. (1997). What constraints should OT allow? Talk handout available online (22 pages), Linguistic Society of America (LSA).

Elgot, C. and Mezei, J. E. (1965). On relations defined by generalized finite automata. *IBM Journal of Research and Development*, 9(1):47–68.

Elias-Ulloa, J. (2006). *Theoretical aspects of Panoan metrical phonology: disyllabic footing and contextual syllable weight*. PhD thesis, Rutgers University.

Everett, D. (1988). On metrical constituent structure in Pirahã. *NLLT*, 6:207–246.

Everett, D. (2003). Iambic feet in paumari and the theory of foot structure. *Linguistic Discover*, 2(1):22–44.

Foster, M. K. (1982). Alternating weak and strong syllables in cayuga words. *International Journal of American Linguistics*, 48(1):59–72.

Fromkin, V. (2013). *Linguistics: An introduction to linguistic theory)*. John Wiley & Sons.

Furby, C. (1974). *Garawa phonology*, volume Series A. Pacific Linguistics, Australian National University.

González, C. (2007). Typological evidence for the separation between stress and foot structure. In Wälchli, M. M. . B., editor, *New challenges in typology: broadening the horizons and redefining the foundations*, pages 55–75. Berlin & New York: Mouton de Gruyter.

González, C. (2009). Rhythmicity in three Panoan languages. *International Journal of Basque Linguistics and Philology*, 37-1:85–137.

González, C. (2011). Word length, syllable count and rhythm: Comparing Urarina and Panoan. In *Encuentro Internacional de Lingüística en el Noroeste*, volume 10, pages 73–92. Editorial Universidad de Sonora, Hermosillo, México.

González, C. (2018). A typology of stress- and foot-sensitive consonantal phenomena. *International Journal of Basque Linguistics and Philology*, pages 87–210.

Gordon, M. (2002). A factorial typology of quantity-insensitive stress. *Natural Language & Linguistic Theory*, 20(3):491–552.

Graf, T. (2017). The power of locality domains in phonology. *Phonology*, 34:385–405.

Haas, M. (1977). Tonal accent in creek. In Hyman, L., editor, *Studies in Stress and Accent*, volume 4 of *Southern California Occasional Papers in Linguistics*, pages 195–208. Department of Linguistics, USC.

Halle, M. and Vergnaud, J.-R. (1978). Metrical structures in phonology. Ms., MIT.

Halle, M. and Vergnaud, J.-R. (1987). *An Essay on Stress*. Cambridge, Mass.: MIT Press.

Hammond, M. (1984). *Constraining Metrical Theory*. PhD thesis, University of California, Los Angeles.

Hammond, M. (1990). *Deriving ternarity*. Ms., University of Arizona, Tuscon, Tuscon, AZ.

Hansen, K. and Hansen, L. E. (1969). Pintupi phonology. *Oceanic Linguistics*, 8:153–170.

Hao, S. and Anderson, S. (2019). Unbounded stress in subregular phonology. *Proceedings of SIGMORPHON 16*.

Harms, R. (1966). Stress, voice, and length in southern paiute. *International Journal of American Linguistics*, 32:228–235.

Harrell, R. (1957). The phonology of colloquial Egyptian Arabic. *American Council of Learned Societies*.

Hayes, B. (1981). *A metrical theory of stress rules*. Ph.D thesis, MIT.

Hayes, B. (1995). *Metrical Stress Theory: Principles and Case Studies*. Chicago: The University of Chicago Press.

Hayward, R. (1990). Notes on the Aari language. In Hayward, R., editor, *Omotic Language Studies*, pages 425–493. School of Oriental and African Studies, London.

Heinz, J. (2007a). *The Inductive Learning of Phonotactic Patterns*. PhD thesis, University of California, Los Angeles.

Heinz, J. (2007b). Learning unbounded stress systems via local inference. In Elfner, E. and Walkow, M., editors, *Proceedings of the 37th Meeting of the Northeast Linguistics Society*. University of Illionois, Urbana-Champaign.

Heinz, J. (2009). On the role of locality in learning stress patterns. *Phonology*, 26(2):303–351.

Heinz, J. (2014). Culminativity times harmony equals unbounded stress. In van der Hulst, H., editor, *Word Stress: Theoretical and Typological Issues*, chapter 8. Cambridge University Press, Cambridge, UK.

Heinz, J. (2018). The computational nature of phonological generalizations. In Plank, L. M. H. . F., editor, *Phonological typology*. De Gruyter Mouton, Berlin & Boston.

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

Hercus, L. (1986). Victorian languages: A late survey. *Pacific Linguistics B77*.

Herrfurth, H. (1964). *Lehrbuch des modernen Djawanisch*. Verlag Enzyklopadie, Leipzig.

Hess, T. (1976). *Dictionary of Puget Salish*. Seattle: University of Washington Press.

Hopcroft, J. and Ullman, J. (1979). *Introduction to Automata Theory, Languages, and Computation*. Addison-Wesley.

Houghton, P. (2013). *Switch Languages: Theoretical Consequences and Empirical Reality*. PhD Thesis, Rutgers University, New Brunswick, NJ.

Hualde, J. I. (1998). A gap filled: postpostinitial accent in azkoitia basque. *Linguistics*, 36(1):99–118.

Hyde, B. (2002). A restrictive theory of metrical stress. *Phonology*, 19(3):313–359.

Hyde, B. (2008). Alignment constraints: Distance-sensitivity, order-sensitivy, and the midpoint pathology. *Ms.*, http://roa.rutgers.edu/article/view/1028.

Hyde, B. (2011b). Extrametricality and non-finality. In M. van Oostendorp, C. Ewen, E. H. and Rice, K., editors, *The Blackwell Companion to Phonology*, pages 1027–1051. Basil Blackwell, Oxford.

Hyman, L. (1977). On the nature of linguistic stress. In Hyman, L., editor, *Studies in Stress and Accent*, Los Angeles, California. USC.

Hyman, L. M. (2006). Word-prosodic typology. *Phonology*, 23(2):225–257.

Idsardi, W. (1992). *The Computation of Prosody*. PhD Dissertation, MIT.

Isac, D. and Reiss, C. (2008). *I-Language: An Introduction to Linguistics as Cognitive Science*. Oxford University Press, Oxford, UK.

Itkonen, E. (1955). Ueber die betonungsverhältnisse in den finnisch-ugrischen sprachen. *Acta Linguistica Academiae Scientiarum Hungaricae*, 5:21–23.

Ito, J. and Mester, A. (1992). Weak layering and word binarity. In Takeru, H., Okazaki, M., Tabata, T., and Tanaka, S., editors, *A New Century of Phonology and Phonological Theory*, pages 26–65. Kaitakusha, Tokyo.

Jardine, A. (2016). Computationally, tone is different. *Phonology*, 33(2):385–405.

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

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

Kager, R. (1992). Are there any truly quantity-insensitive systems? In Buszard-Welcher, L., Lee, L., and Weigel, W., editors, *Proceedings of the Eighteenth Annual Meeting of the Berkeley Linguistics Society*, pages 123–132, Berkeley. University of California Press.

Kager, R. (1993). Alternatives to the iambic-trochaic law. *NLLT*, 11:381–432.

Kager, R. (2005). Rhythmic licensing: an extended typology. In *Proceedings of the 3rd International Conference on Phonology*, Seoul. The Phonology Morphology Circle of Korea. 5?31.

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

Kager, R. (2012). Stress in windows: Language typology and factorial typology. *Lingua*, 122(13):1454–1493.

Kamprath (1987). *Suprasegmental Structure in a Raeto- Romansch Dialect: A Case Study in Metrical and Lexical Phonology*. Austin: University of Texas, Ph.D dissertation.

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

Kaye, J. (1973). Odawa stress and related phenomenon. *Odawa Language Project: Second report*.

Kenstowicz, M. (1994). *Phonology in Generative Grammar, vol. 7*. Blackwell, Cambridge, MA.

Kentsowicz, M. (1980). Notes on Cairené Arabic syncope. *Studies in the Linguistic Sciences*, 10.2:39–53.

Kormushin, I. V. (1998). *Udihe Language*. PhD thesis, Moscow, Nauka.

Koser, N. and Jardine, A. (2020a). The complexity of optimizing over strictly local constraints. *Penn Working Papers in Linguistics 26:1*.

Koser, N. and Jardine, A. (2020b). The computational nature of stress assignment. *In Proceedings of AMP 2019*.

Koser, N. and McCollum, A. (to appear). Computation of unbounded processes with a suprabinary scale requires non-determinism. In *Proceedings of NELS 52*, Rutgers University.

Larsen, R. and Pike, E. (1949). Huasteco intonations and phonemes. *Language*, 25:268–277.

Lauriault, J. (1948). Alternate-mora timing in Shipibo. *IJAL*, 14:22–24.

Levin, J. (1988a). Bi-directional foot construction as a window on level ordering. In Hammond, M. and Noonan, M., editors, *Theoretical Morphology*, pages 339–352.

Lewis, M. (1947). *Teach Yourself Malay*. London: English Universities Press.

Liberman, M. and Prince, A. (1977). On stress and linguistic rhythm. *Linguistic Inquiry*, 8.2:249–336.

Loos, E. (1969). *The Phonology of Capanahua and its Grammatical Basis*. Summer Institute of Linguistics, University of Oklahoma.

Lunt, H. (1952). *A grammar of the Macedonian literary language*. Skopje.

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

MacDonald, L. (1990). *A Grammar of Tauya*. Mouton de Gruyter, New York.

Marlo, M., Mwita, L., and Paster, M. (2015). Problems in Kuria H tone assignment. *NLLT*, 33(1):251–265.

Martin, J. (2011). *A grammar of Creek (Muskogee)*. Lincoln: University of Nebraska Press.

Martin, J. and Johnson, K. (2002). An acoustic study of "tonal accent" in Creek. *IJAL*, 68:28–50.

Matteson, E. (1965). The piro (arawakan) language. *University of California Publications in Linguistics*, 22.

McCarthy, J. (1988). Feature geometry and dependency: a review. *Phonetica*, 45:84–108.

McCarthy, J. J. (1979). On stress and syllabification. *Linguistic Inquiry*, 53.

McCollum, A., Baković, E., Mai, A., and Meinhardt, E. (2020). Unbounded circumambient patterns in segmental phonology. *Phonology*, 37(2):215–255.

McNaughton, R. and Papert, S. (1971). *Counter-Free Automata*. MIT Press.

Meinhardt, E., Mai, A., Bakovic, E., and McCollum, A. (submitted). On the proper treatment of weak determinism: Subsequentiality and simultaneous application in phonological maps. *https://ling.auf.net/lingbuzz/005565*.

Mohanan, K. (1986). *The Theory of Lexical Phonology*. Reidel, Dordrecht.

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

Niang, M. (1997). *Constraints on Pulaar Phonology*. University Press of America, New York.

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

Olawsky, K. J. (2006). *A Grammar of Urarina*. Berline: Mouton de Gruyter.

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

Osborn, H. (1966). Warao i: phonology and morphophonemics. *International Journal of American Linguistics*, 32:108–123.

Padgett, J. (1995). Partial class behavior and nasal place assimilation. In Elzinga, K. S. . D., editor, *Proceedings of the 1995 Southwestern Workshop on Optimality Theory (SWOT)*, pages 145–183. Tuscon: Department of Linguistics, University of Arizona.

Parker, S. (1994). Coda epenthesis in Huariapano. *International Journal of American Linguistics*, 60:95–119.

Parker, S. (1998). Disjoint metrical tiers and positional markedness in Huariapano. *GIAL Electronic Notes Series*.

Parker, S. (1999). A sketch of Iñapari phonology. *International Journal of American Linguistics*, 65(1):1–39.

Paster, M. (2019). Phonology counts. *Radical: A Journal of Phonology*, 1:1–61.

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

Pike, K. (1964). Stress trains in Auca. In Abercrombie, D., editor, *In honour of David Jones*. Longmans.

Prince, A. (1983). Relating to the grid. *Linguistic Inquiry*, 14(1):19–100.

Prince, A. (1985). Improving tree theory. *Berkeley Linguistics Society*, 11:471–490.

Prince, A. and Smolensky, P. (1993). Optimality theory: constraint interaction in generative grammar. *Ms, Rutgers University*.

Rice, C. (1992). *Binarity and ternarity in metrical theory: parametric extensions*. Doctoral dissertation, University of Texas at Austin, Austin, TX.

Rogers, J., Heinz, J., Bailey, G., Edlefsen, M., Visscher, M., Wellcome, D., and Wibel, S. (2010). On languages piecewise testable in the strict sense. In Ebert, C., Jäger, G., and Michaelis, J., editors, *The Mathematics of Language*, pages 255–265, Berlin, Heidelberg. Springer Berlin Heidelberg.

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

Rogers, J. and Lambert, D. (2019). Extracting subregular constraints from regular stringsets. *Journal of Language Modeling*, 7(2):143–176.

Rolle, N. and Lionnet, F. (2020). Phantom structure: a representational account of floating tone association. In Baek, H., Takahashi, C., and Yeung, A. H.-L., editors, *Proceedings of AMP 2019*.

Safir, K. (1979). Metrical structure in capanahua. In Safir, K., editor, *MIT Working Papers in Linguistics*. MIT, Cambridge, Mass.

Sakarovitch, J. (2006). *Elements of Automata Theory*. Cambridge University Press, Cambridge, UK.

Saksena, B. (1971). *Evolution of Awadhi*. Motilal Banarsidass, Delhi.

Salomaa, A. (1969). *Theory of Automata*. Pergamon Press.

Sapir, E. and Swadesh, M. (1960). *Yana Dictionary*. University of California Press, Berkeley.

Schütz, A. (1985). *The Fijian Language*. Honolulu: University of Hawaii Press.

Schutzenberger, M. (1977). Sur une variante des fonctions sequentielles. *Theoretical Computer Science*, 4:47–57.

Seiler, H. (1977). *Cahuilla Grammar*. Malki Museum Press, Banning, California.

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

Shipley, W. (1964). Maidu grammar. *University of California Publications in Linguistics*, 41.

Stanton, J. (2016). Learnability shapes typology: The case of the midpoint pathology. *Language*, 92(4):753–791.

Street, C. S. and Mollinjin, G. P. (1981). The phonology of murinbata. *Australian phonologies: Collected papers*, pages 183–244.

Strother-Garcia, K. and Heinz, J. (2017). Logical foundations of syllable representations. In *Poster presented at the 2017 Annual Meeting on Phonology*.

Subdara, J. (1944). Maithili phonetics. *Indian Linguistics*, 8:435–459.

van Oostendorp, M. (1993). Formal properties of metrical structure. In *Sixth Conference of the European Chapter of the Association for Computational Linguistics*, pages 322–331, Utrecht. ACL.

Whitman, W. (1947). Descriptive grammar of ioway-oto. *International Journal of American Linguistics*, 13(4):233–248.

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

Wilson, C. (2006). Unbounded spreading is myopic.

Winstedt, R. (1927). *Malay Grammar*. Oxford: Oxford University Press.

Zigmond, M., Booth, C., and Munro, P. (1990). Kawaiisu: A Grammar and Dictionary. *University of California Publications in Linguistics*, 119.