

The computational nature of stress assignment

Nate Koser and Adam Jardine
Rutgers University

1 Introduction

While computational studies of stress patterns as phonotactics have yielded restrictive characterizations of stress (Rogers et al., 2013) with provably correct learning procedures (Heinz, 2009), an outstanding question is the nature of stress assignment as a *function* which assigns stress to an underlying bare string of syllables. This paper fills this gap by locating stress patterns with respect to the *subsequential* class of functions (Mohri, 1997), which are argued to be important for phonology in that the vast majority of phonological functions fall within the subsequential boundary (Heinz & Lai, 2013; Chandlee, 2014), with the notable exception of tone and vowel harmony (Jardine, 2016; McCollum et al., 2018).

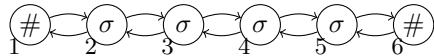
The main result is that—while most, if not all *quantity insensitive* (QI) stress systems are subsequential functions—the same does not hold for *quantity sensitive* (QS) systems. Counter-intuitively, so-called *default-to-opposite* QS patterns are subsequential, but *default-to-same* QS patterns are provably not. It also supports the claim of Jardine (2016) that certain tonal patterns are non-sequential because their suprasegmental nature allows for more a more powerful computation. As stress assignment is also suprasegmental, the existence of non-sequential stress functions adds evidence for this conclusion.

2 Logical maps

2.1 Model theoretic representations of stress In this paper we define stress assignment as a function from strings of bare syllables to a string of syllables marked with stress. We use *logical transductions* (Courcelle, 1994) between model theoretic representations of the input and output. The transductions are defined with a predicate logic written with *terms*—a variable like x , for instance—that range over positions in the input string and *atomic predicates* that evaluate the labels at those positions.

For example, consider a string of four syllables and word boundaries. We can define this string as a model with a domain \mathcal{D} , a set comprised of six elements: unary relations σ and $\#$, sets of elements whose members are labeled as syllables or word boundaries; and two functions p (predecessor) and s (successor) that describe the ordering of the elements in the model. This corresponds to the following representation of the sequence $\#\sigma\sigma\sigma\#$:

$$(1) \langle \mathcal{D}; P_\sigma, p, s \rangle \mathcal{D} = \{1, 2, 3, 4, 5, 6\}; \sigma = \{2, 3, 4, 5\}, \# = \{1, 6\}; p = x - 1, s = x + 1$$



Atomic formulae of the model take variables that are interpreted with regard to the elements in the model. For example, if $\sigma(x)$ is an atomic formula in some model, then it is true when x is evaluated at a position that is an unstressed syllable. In the model above in (1), $\sigma(x)$ is true for elements two, three, four, and five, as they are all syllables i.e. in the set of the unary relation σ .

Given a logic and a model, transductions between input and output strings can be defined. As the existence of a map between an underlying representation and a surface form is a core tenet of generative phonology, the properties of these functions are informative for phonological theory in general. For stress assignment, the input is a string of bare syllables, and the output is a string of syllables marked with stress. Stress functions preserve the order and number of syllables. Stress will appear in the output on those elements

* The authors would like to thank Adam McCollum, Sophie Hao, and the members of Jardine’s 2019 seminar on logic and the computation of long-distance processes for their comments and suggestions. All errors are our own.

for which the definition of a stress predicate $\hat{\square}(x)$ for primary stress or $\check{\square}$ for secondary stress is satisfied. For example, the definition in (2) would place stress on the peninitial syllable:

$$(2) \quad \hat{\square}(x) \stackrel{d}{=} \#(p(p(x)))$$

The formula places stress on the output syllable for which double application of the p function returns a word boundary—the peninitial syllable. We assume that p and s are undefined when their application would return a position outside of the string. If $p(x)$ is undefined at a certain position, the predicate $\hat{\square}(p(x))$ evaluated at that position is false. The definition in (2) applied to the first syllable is false, since $p(p(x))$ of the first syllable is undefined. This function describes a mapping for a string of any length:

$$(3) \quad \begin{array}{ll} \#\sigma\sigma\sigma\# & \mapsto \#\sigma\acute{\sigma}\sigma\# \\ \#\sigma\sigma\sigma\sigma\# & \mapsto \#\sigma\acute{\sigma}\sigma\sigma\# \\ \#\sigma\sigma\sigma\sigma\sigma\# & \mapsto \#\sigma\acute{\sigma}\sigma\sigma\sigma\# \\ \dots & \mapsto \dots \end{array}$$

The transduction in (2) is *quantifier-free* (QF). As it does not invoke logical quantifiers \exists or \forall , there is no way to reference positions that are an unbounded distance away. For phonology, this is appropriate in many cases—Chandlee (2014) shows that a range of phonological processes are QF-definable.

But not all phonological processes are local in this sense—including stress assignment. In languages where stress iterates throughout a word, placement of stress cannot be defined by making local reference to a position in the input. Instead, we adopt the *implicit definitions* of Rogers (1997), which allow for functions that refer to local information in the output by defining them recursively (Koser et al., 2019). An example is given here:

$$(4) \quad \check{\square}(x) \stackrel{d}{=} \text{final}(x) \wedge \check{\square}(s(s(x)))$$

The “base case” of secondary stress is the final syllable. Then, by recursively referring to $\check{\square}(x)$, secondary stresses are placed on syllables that are two elements to the left of another secondary stress. To the implicit definitions we add a restriction that recursive definitions refer to either the p or s functions, but not both. This restriction ensures that the transduction is subsequential and preserves a well defined notion of locality (Chandlee & Jardine, 2019).

2.2 The subsequential class and phonology The complexity classes of formal language theory (FLT) divide the space of possible functions based on the expressive power of those functions. Applying the study of FLT complexity to natural language has proven extremely informative. One result of this line of research has shown that phonological processes fall within the *regular* class of functions (Johnson, 1972; Kaplan & Kay, 1994). Intuitively, this means that phonological functions are calculated based on a bounded amount of information. Further work has shown that most of phonology is subregular (Rogers et al., 2013; Heinz, 2015). Study of the subregular classes and their relevance to phonology is an ongoing program that delivers precise, mathematically-defined characterizations of phonological processes.

The subsequential class of functions (Mohri, 1997) is one class that is relevant to phonology. It is properly subregular, has well-understood automata-theoretic properties, and represents an important division in the space of possible functions that includes many phonological patterns and excludes many non-phonological ones (Heinz & Lai, 2013; Chandlee, 2014; Jardine, 2016; Payne, 2017; Luo, 2017). Though not all phonological functions are subsequential (Jardine, 2016; McCollum et al., 2018), the vast majority are. Based on this fact, we take the subsequential class to be a hypothesis for stress assignment functions—most, if not all, are hypothesized to fall within this boundary.

In the sections that follow, we show that the subsequential hypothesis is too strong for stress, and that while most stress functions do fall within the class, there are some that do not.

3 Analysis: Quantity Insensitive Stress

In QI stress languages, stress is placed in a predictable location given a word of any length, regardless of any properties the syllables in the word may have. We start with the simplest cases of non-iterative stress and then progress to the more complicated patterns.

3.1 Non-iterative In non-iterative stress languages, stress always falls on a single syllable some distance from a word edge. There is no iterative secondary stress. There are five attested patterns of this type: initial, peninitial, antepenultimate, penultimate, and final stress (Hyman, 1977; Hayes, 1995; Gordon, 2002). By number, these patterns comprise around 90% of the languages seen in Gordon’s typology of QI stress.

Consider initial stress, of which Nenets (Decsy, 1966) is an example. Stress always falls on the initial syllable. A transduction from a bare, stressless input of syllables to an output marked with initial stress can be written with the following logical statement:

$$(5) \quad \acute{\square}(x) \stackrel{d}{=} \#(p(x))$$

This definition states that an element x will be labeled with stress in the output if its predecessor p in the input is a word boundary. This predicate will evaluate to true for the first syllable in any input string of syllables, as shown here in (6):

$$(6) \quad \#\sigma\sigma\sigma\sigma\# \mapsto \overset{\circ}{\#}\overset{\circ}{\sigma}\sigma\sigma\sigma\#$$

When evaluating the first syllable in the input as x , the predecessor of x is the word boundary. The stress predicate $\acute{\square}$ evaluates to true, labeling the first syllable with stress in the output. This definition will take an input string of any length and apply stress to the first syllable, thus describing an initial stress pattern.

While the explicit definition in (5) is informative, it is also useful to create user-defined predicates that make other, more complex definitions easier to understand. Examples of some useful predicates are shown in (7):

$$(7) \quad \begin{aligned} \text{initial}(x) &\stackrel{d}{=} \#(p(x)) \\ \text{penult}(x) &\stackrel{d}{=} \#(s(s(x))) \\ \text{final}(x) &\stackrel{d}{=} \#(s(x)) \\ \text{only}(x) &\stackrel{d}{=} \text{initial}(x) \wedge \text{final}(x) \end{aligned}$$

It is important to remember what is “under the hood” of user-defined predicates, but they are great aids in understanding the more complicated transductions described later in the chapter. Note that, because they are defined in terms of QF logic, using these predicates does not increase the expressive power of our formalism.

Another non-iterative pattern is the penultimate pattern. Penultimate stress targets the second to last syllable in a word. A QF transduction describing this pattern is given in (8), using the definition of $\text{penult}(x)$ seen in (7):

$$(8) \quad \acute{\square}(x) \stackrel{d}{=} \text{penult}(x)$$

The definition refers to exactly the penultimate position in a string i.e. the position in the string where double application of the successor function returns an element that is a word boundary.

Just as above in (5) for initial stress, this transduction maps any input string of syllables to a string of syllables in the output where the penultimate syllable is stressed.

$$(9) \quad \#\sigma\sigma\sigma\sigma\# \mapsto \#\sigma\sigma\overset{\circ}{\sigma}\overset{\circ}{\sigma}\#$$

For any of the five attested non-iterative stress patterns, there is a similar transduction describing the application of stress to a string of syllables. Rather than explicitly describing the remaining three, I add them to the list of user-defined predicates in (7), updated in (10):

$$(10) \quad \begin{aligned} \text{initial}(x) &\stackrel{d}{=} \#(p(x)) \\ \text{peninitial}(x) &\stackrel{d}{=} \#(p(p(x))) \\ \text{antepenult}(x) &\stackrel{d}{=} \#(s(s(s(x)))) \\ \text{penult}(x) &\stackrel{d}{=} \#(s(s(x))) \\ \text{final}(x) &\stackrel{d}{=} \#(s(x)) \\ \text{only}(x) &\stackrel{d}{=} \text{initial}(x) \wedge \text{final}(x) \end{aligned}$$

Each of the first five predicates listed describes a transduction corresponding to placement of stress in non-iterative QI stress languages. They will prove useful in defining iterative stress patterns, as the iteration in those cases originates from one of the five positions the predicates describe.

Note that placement of stress in these cases is calculated entirely based on local information in the input string. The definitions contain only input elements and p or s functions. There is no need for recursive predicates or reference to quantifiers. QF logical transductions are known to correspond to *input strictly local* functions (ISL; Chandlee, 2014; Chandlee & Jardine, 2019; Chandlee & Lindell, forthcoming). The ISL class is a very restrictive class that is well within the subsequential boundary and is relevant to phonology – around 95% of phonological patterns in Pbase (Mielke, 2004) can be characterized described with an ISL function.

Showing that stress assignment in non-iterative languages is ISL has important theoretical implications. Any linguistic theory of word stress in non-iterative QI languages need not posit an input-output function that is more powerful than ISL. It also establishes the place of non-iterative patterns on the complexity hierarchy, providing a starting point for comparison. It will be shown below that other kinds of stress patterns are formally more complex than non-iterative patterns. We now turn to iterative stress patterns.

3.2 Iterative In iterative stress languages, stress is applied throughout the word, often skipping over a syllable before continuing. Some languages avoid stressing the final syllable, such as Pintupi (Hansen & Hansen, 1969). In others, such as Murinbata (Street & Mollinjin, 1981), iteration is absolute:

- (11) Pintupi : $\acute{\sigma}, \acute{\sigma}\sigma, \acute{\sigma}\sigma\sigma, \acute{\sigma}\sigma\grave{\sigma}, \acute{\sigma}\sigma\grave{\sigma}\sigma, \acute{\sigma}\sigma\grave{\sigma}\sigma\sigma, \acute{\sigma}\sigma\grave{\sigma}\sigma\grave{\sigma}\sigma\dots$
 Murinbata : $\acute{\sigma}, \acute{\sigma}\sigma, \acute{\sigma}\sigma\grave{\sigma}, \acute{\sigma}\sigma\grave{\sigma}\sigma, \acute{\sigma}\sigma\grave{\sigma}\sigma\sigma, \acute{\sigma}\sigma\grave{\sigma}\sigma\grave{\sigma}\sigma\dots$

Consider the pattern of Pintupi. Primary stress is placed at the left edge, from where secondary stress spreads to every odd syllable thereafter excluding the final syllable. This kind of stress assignment is unbounded in the sense that the number of stresses grows as the length of the word grows. The transduction must model this unbounded iteration. Consider the following attempt at a definition of Pintupi stress modeled after the definitions for non-iterative stress above:

- (12) $\acute{\square}(x) \stackrel{d}{=} \text{initial}(x)$
 $\grave{\square}(x) \stackrel{d}{=} \text{initial}(p(p(x)))$

A transduction under this definition would produce the correct output for forms with only one secondary stress, by relying on the fact that the first syllable is two positions before the secondary stress. But for longer forms, it fails to correctly describe the Pintupi pattern. Adding disjuncts to the definition – *first(p(p(p(p(x))))*) for example – is only a temporary fix. More such disjuncts would need to be added to the word as it increases in length.

Instead, the transductions here incorporate recursive predicates. Placement of secondary stress depends on whether another syllable at some point in the *output* is also stressed. This is formally different from the non-iterative cases above that could be computed based entirely on the input, as demonstrated by the failure of the input-only formulae in (12) to capture Pintupi stress.

In order to refer to the output, we use the implicit definitions of Rogers (1997), where the definition of a predicate can refer to itself directly. The restriction that recursive functions in a single definition refer to either successor *or* predecessor ensures that the transduction is subsequential. A corrected version of (12) that accurately captures the Pintupi pattern is given in (13):

- (13) $\acute{\square}(x) \stackrel{d}{=} \text{initial}(x)$ (a)
 $\grave{\square}(x) \stackrel{d}{=} \underbrace{\text{initial}(p(p(x)))}_{(b)} \vee \underbrace{\grave{\square}(p(p(x)))}_{(c)} \wedge \underbrace{\neg \text{final}(x)}_{(d)}$

The formulae are interpreted as follows. In (13a), evaluation of $\acute{\square}(x)$ puts primary stress on the first syllable. Evaluation of $\grave{\square}(x)$ puts secondary stress on the third syllable via disjunct (13b)—its predecessor's predecessor is the first element in the string. $\grave{\square}(x)$ also puts secondary stress on the fifth syllable—the element two positions to the left bears secondary stress in the output, satisfying disjunct (13c). The conjunct in (13d) prevents stressing of the final syllable, expressing the non-finality requirement of the language. No other element satisfies the formulae, and so the transduction arrives at the correct surface form for Pintupi, as shown here for six and seven syllable forms:

$$(14) \quad \# \sigma \sigma \sigma \sigma \sigma \# \mapsto \# \overset{\sigma}{\sigma} \overset{\sigma}{\sigma} \overset{\sigma}{\sigma} \sigma \overset{\sigma}{\sigma} \# \quad \# \sigma \sigma \sigma \sigma \sigma \sigma \# \mapsto \# \overset{\sigma}{\sigma} \overset{\sigma}{\sigma} \overset{\sigma}{\sigma} \sigma \overset{\sigma}{\sigma} \sigma \#$$

A transduction for Murinbata is like the definition in (13), but with the final conjunct (d) removed. Stress is allowed to fall on the final syllable in Murinbata.

Another group of iterative QI patterns are the “bidirectional” patterns (Kager, 2007). These feature a primary stress fixed at one edge, with secondary stress iterating from the opposite edge. When this iteration would create a clash between secondary and primary stress, this is avoided, creating a word-internal lapse. One language displaying this pattern is Garawa (Furby, 1974):

$$(15) \quad \text{Garawa : } \acute{\sigma}, \acute{\sigma}\sigma, \acute{\sigma}\sigma\sigma, \acute{\sigma}\sigma\delta\sigma, \acute{\sigma}\sigma\sigma\delta\sigma, \acute{\sigma}\sigma\delta\sigma\delta\sigma, \acute{\sigma}\sigma\sigma\delta\sigma\delta\sigma, \acute{\sigma}\sigma\delta\sigma\delta\sigma\delta\sigma$$

On the surface the pattern may appear more complex than those of Pintupi or Murinbata, and the label “bidirectional” seems to suggest this as well. However, the explicit definition of the transduction reveals that this is not the case:

$$(16) \quad \begin{aligned} \overset{d}{\square}(x) &\stackrel{d}{=} \text{initial}(x) \\ \underset{d}{\square}(x) &\stackrel{d}{=} (\text{penult}(x) \vee \underset{d}{\square}(s(s(x)))) \wedge \neg \text{peninitial}(x) \end{aligned}$$

The first syllable always has main stress. The penult receives secondary stress, and further secondary stresses are placed two positions to the left. The exception is the peninitial syllable, which is barred from carrying secondary stress through the $\neg \text{peninitial}(x)$ conjunct, preventing a clash. The transduction in (16) correctly describes a Garawa-like stress assignment function for input strings of any length:

$$(17) \quad \# \sigma \sigma \sigma \sigma \sigma \sigma \# \mapsto \# \overset{\sigma}{\sigma} \overset{\sigma}{\sigma} \overset{\sigma}{\sigma} \sigma \overset{\sigma}{\sigma} \# \quad \# \sigma \sigma \sigma \sigma \sigma \sigma \sigma \# \mapsto \# \overset{\sigma}{\sigma} \sigma \sigma \overset{\sigma}{\sigma} \sigma \overset{\sigma}{\sigma} \sigma \#$$

The definition is not unlike that of Pintupi, but instead of iterating away from the main stress, Garawa iterates toward it. Just like Pintupi, iteration only occurs in one direction via the s function.

The QI iterative patterns necessitate reference to the output and so are formally more complex than the non-iterative patterns. They require recursive logical definitions while non-iterative patterns do not. Despite the difference in complexity, the patterns mentioned so far adhere to the requirement that recursive predicates refer to either the p or s functions – not both. This means that iterative stress assignment in these cases is subsequential, essentially because iteration only proceeds in one direction. This is true even in so-called “bidirectional” cases like Garawa.

However, it may not be the case that all iterative stress assignment is subsequential. A potential counterexample comes from Cahuilla (Seiler, 1977; Idsardi, 1988; Levin, 1988a; Hayes, 1995). Though in general Cahuilla is a QS language, in forms with only light syllables the following QI-like pattern emerges:

$$(18) \quad \begin{aligned} &\delta\sigma + \acute{\sigma}\sigma\delta \\ &\sigma\delta\sigma + \acute{\sigma}\sigma\delta\sigma \\ &\delta\sigma\delta\sigma + \acute{\sigma}\sigma\delta\sigma\delta \\ &\sigma\delta\sigma\delta\sigma + \acute{\sigma}\sigma\delta\sigma\delta\sigma \\ &\dots \end{aligned}$$

The root boundary is indicated with the + symbol. Primary stress falls on the first syllable of the root. From this position, secondary stresses iterate backwards into a prefix and forwards through the root and suffix. The pattern resembles that of Pintupi or Murinbata, but moving in both directions. Unlike Garawa, this is a truly bidirectional pattern that recursively uses both the p and s functions, and is not subsequential.

$$(19) \quad \begin{aligned} \overset{d}{\square}(x) &\stackrel{d}{=} +(p(x)) \\ \underset{d}{\square}(x) &\stackrel{d}{=} \overset{d}{\square}(p(p(x))) \vee \overset{d}{\square}(s(s(x))) \vee \overset{d}{\square}(p(p(x))) \vee \overset{d}{\square}(s(s(x))) \end{aligned}$$

Primary stress falls on the initial syllable of the root. Secondary stress is placed on syllables two the left or right of the initial stress via the first pair of disjuncts. The latter pair of disjuncts allow for recursive application of secondary stress to further, more distant syllables. Unlike Pintupi or Murinbata, the definition in (19) invokes both p and s recursively, and so is not subsequential. This means that not all iterative stress assignment, and possibly not all QI stress assignment, is subsequential. Given that the vast majority of phonological patterns are subsequential, this is a noteworthy finding. Along with Jardine (2016), it suggests that suprasegmental processes in general have access to non-subsequential functions.

We now turn to QS stress patterns, showing that non-subsequential patterns are present there as well.

4 Analysis: Quantity Sensitive stress

In QS stress systems, placement of stress in a word is dependent on the presence/location of “heavy” and “light” syllables. The input is a string of light syllables (L) and heavy syllables (H). Here we focus on two relevant patterns — the *default to same* (DTS) and *default to opposite* (DTO) patterns (terminology from (Prince, 1985)). Both of these patterns are unbounded in the sense that whether a syllable is stressed may depend on another syllable an arbitrary distance away. They are not iterative — there is only one stress per word.

Stress assignment in these languages must track the presence of heavy syllables in the string. In order to incorporate this into the transductions, we introduce two user defined predicates:

$$(20) \quad \begin{aligned} \textit{precede-H}(x) &\stackrel{d}{=} H(s(x)) \vee \textit{precede-H}(s(x)) \\ \textit{follow-H}(x) &\stackrel{d}{=} H(p(x)) \vee \textit{follow-H}(p(x)) \end{aligned}$$

These predicates are true of any element that precedes (or follows) a heavy syllable. Consider the string *LLLHLL*, for example. When evaluating $\textit{precede-H}(x)$, the third syllable satisfies the disjunct $H(s(x))$, as it has a heavy syllable for its successor. This means that the second syllable satisfies the latter, recursive disjunct $\textit{precede-H}(s(x))$, as the succeeding element evaluates to true for $\textit{precede-H}(x)$ in the output. Likewise for the first syllable. The recursive definition ensures that all elements preceding a heavy syllable will evaluate to true:

$$(21) \quad \overleftarrow{\textit{LLLHL}}$$

Note that, since one of the predicates in (20) uses the successor function recursively and the other uses the predecessor function recursively, a stress transduction that employs both will not be subsequential. Bearing this in mind, we demonstrate that the DTO pattern is subsequential, but the DTS pattern is not (see Hao & Andersson (2019) for a parallel result).

4.1 Default to opposite In DTO, a heavy syllable nearest an edge is stressed. In the absence of any heavy syllables, a light syllable at the opposite edge is stressed instead. There are two logical possibilities, referred to as leftmost-heavy or right (LHOR) and rightmost-heavy or left (RHOL). The following diagram gives a few examples of stress assignment in these languages (LHOR – Kwakw’ala; RHOL — Selkup (Hayes, 1995)):

$$(22) \quad \begin{array}{ll} \textit{LHOR} & \textit{RHOL} \\ \textit{LLL}\acute{\textit{L}} & \acute{\textit{L}}\textit{LLL} \\ \acute{\textit{H}}\textit{HHH} & \textit{HHH}\acute{\textit{H}} \\ \textit{L}\acute{\textit{H}}\textit{LLLL} & \textit{L}\acute{\textit{H}}\textit{LLLL} \\ \textit{L}\acute{\textit{H}}\textit{LLHL} & \textit{LHLL}\acute{\textit{H}}\textit{L} \end{array}$$

Consider the LHOR pattern. A light syllable is only stressed when in the final position and not preceded by a heavy at any point. For a heavy syllable, it is exactly the same—it is only stressed when no heavies precede it, making it the leftmost heavy. Thus, the search of the string proceeds in the *same direction* via the predecessor function when checking if either a light or heavy syllable should be labeled as stressed. This is reflected in the definition of the transduction, which invokes only one of the predicates from (20):

$$(23) \quad \acute{\square}(x) \stackrel{d}{=} (L(x) \wedge \textit{final}(x) \wedge \neg \textit{follow-H}(x)) \vee (H(x) \wedge \neg \textit{follow-H}(x))$$

The first half of the disjunct handles the case where light a syllable is stressed. This occurs when a light syllable is final and does not follow any heavy syllables. This is the “default” case of DTO — a light syllable can only be stressed when no heavies are present in the word at all.

The second half of the disjunct describes when a heavy syllable can be stressed. This occurs when a heavy does not follow any other heavies, making it the leftmost one. As the definition makes explicitly clear, search of the string for heavy syllables proceeds in the same direction via $\textit{follow-H}(x)$ when checking if both light or heavy syllables should be stressed in the output. This means that the DTO stress transduction is

subsequential. It also shows that, although stressed light syllables and heavy syllables may tend to appear on opposite ends of a word, there is nothing “opposite” about the stress function itself. It is oriented in a single direction, as shown here:

$$(24) \text{ LHOR: } \overleftarrow{\text{LL} \boxed{\acute{\text{H}}} \text{LHL}} \quad \overleftarrow{\text{LLLLLL} \boxed{\acute{\text{L}}}}$$

We now show that the same does not hold of DTS stress mappings.

4.2 Default to same In DTS systems, a heavy syllable nearest an edge is stressed. In the absence of any heavy syllables, a light syllable at the same edge is stressed instead. As with default to opposite patterns, there are two logical possibilities – leftmost-heavy or left (LHOL) and rightmost-heavy or right (RHOR). Both are attested and assign stress as shown here (LHOL – Lushootseed; RHOR – Klamath (Hayes, 1995)):

$$(25) \begin{array}{ll} \text{LHOL} & \text{RHOR} \\ \acute{\text{L}}\text{LLL} & \text{LLL}\acute{\text{L}} \\ \acute{\text{H}}\text{HHH} & \text{HHH}\acute{\text{H}} \\ \text{L}\acute{\text{H}}\text{LLLL} & \text{L}\acute{\text{H}}\text{LLLL} \\ \text{L}\acute{\text{H}}\text{LLHL} & \text{LHLL}\acute{\text{H}}\text{L} \end{array}$$

Take the LHOL pattern for example. If no heavy syllables are present, the initial light syllable is stressed. If one or more heavy syllables are present the leftmost heavy syllable is stressed instead. Whereas for LHOR stress the unbounded search to check if a light or a heavy syllable should be stressed proceeded via the p function, now there is no way to define the transduction without mixing p and s :

$$(26) \quad \acute{\square}(x) \stackrel{d}{=} (L(x) \wedge \text{initial}(x) \wedge \neg \text{precede-H}(x)) \vee (H(x) \wedge \neg \text{follow-H}(x))$$

The disjunct handling the case of heavy syllables is unchanged—heavy syllables receive stress when they do not follow any other heavies, and so are leftmost. However, now light syllables can only be stressed when they are initial and do not *precede* any heavies. This means that the computation of the LHOL function (or any DTS function) must be sensitive to information that may lie unboundedly far away in both directions:

$$(27) \text{ LHOL: } \overleftarrow{\text{LL} \boxed{\acute{\text{H}}} \text{LHL}} \quad \overrightarrow{\boxed{\acute{\text{L}}}\text{LLLLLL}}$$

The true bidirectional nature of the function means that DTS stress assignment is non-subsequential and formally more complex than DTO stress assignment. (We have shown this informally here; for a proof see Hao & Andersson (2019), who discovered this independently.) This result is counter-intuitive—the “opposite-sided” nature of the DTO patterns might lead one to assume it is more complex, but a computational study reveals that the reverse is true.

4.3 Beyond subsequential So far, we have seen that not all QS stress or iterative QI stress is subsequential. A natural question to ask is—if these stress functions are not subsequential, then what are they? What class of functions do truly bidirectional stress patterns belong to?

One possibility is that the bidirectional cases are describable by a *weakly deterministic* transduction (WD; Heinz & Lai, 2013; McCollum et al., 2018). In short, a WD function is the composition of a left subsequential function and a right subsequential function, where the two functions cannot interact in a formally-defined sense. An intuitive notion of interaction as formalized by McCollum et al. is that if reordering the functions results in different final outputs, then the functions interact and are not WD.

The LHOL function, broken down into the composition of two functions, adheres to this restriction. First, a left-to-right function reads the string, outputting a stressed H for the first input H it encounters. This expresses the “leftmost heavy” portion of LHOL stress and crucially is only change it can make to the string. A subsequential transducer describing this function is given in Figure 1. Basically, a transducer is subsequential if it is *deterministic*. That is, for every input symbol at every state, there is only one transition leaving that state for that symbol. This means that every input string corresponds to at most one output string.

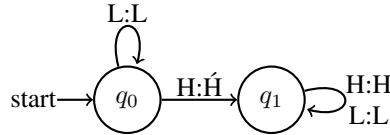


Figure 1: left-to-right transducer for WD LHOL stress map

A second function, reading right-to-left, applies stress to the initial L in words comprised of only Ls. It waits when the first input symbol is an L, finally outputting a stressed L when the string ends after some sequence L^n . Otherwise, the function faithfully outputs symbols from the input.

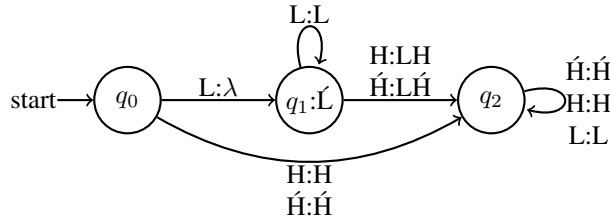


Figure 2: right-to-left transducer for WD LHOL stress map

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. Consider the string LHHLLL, for example. The left-to-right function will label the first H with stress, and leave other elements unchanged. Since the string contains heavy syllables, the right-to-left transducer will leave it unchanged, giving the correct output LHHLLL. Crucially, if the order of functions is reversed, the output is the same. The right-to-left function leaves the string unchanged, and the left-to-right function marks the first H with stress. There indicates that there is no interaction and that the stress function for LHOL (and DTS in general) is WD.

An example of a pattern that falls outside the WD boundary is the pathological “sour grapes” stress pattern (Koser & Jardine, 2019):

$$(28) \quad \acute{\sigma}, \acute{\sigma}\sigma, \acute{\sigma}\sigma\sigma, \acute{\sigma}\sigma\acute{\sigma}\sigma, \acute{\sigma}\sigma\sigma\sigma\sigma, \acute{\sigma}\sigma\acute{\sigma}\sigma\sigma, \acute{\sigma}\sigma\sigma\sigma\sigma\sigma$$

The pattern is like cases of sour grapes in harmony, where some process is completely abandoned under certain conditions when it should apply partially instead (Wilson, 2003, 2006). In this case, stress only iterates in forms with an even number of syllables. Such patterns are unattested. Stress assignment is not sensitive to the length of the string, and so the pattern is pathological.

An input-output mapping describing sour grapes stress can be written as the composition of two subsequential functions. Crucially, however, this composition interacts and so is not WD under the definition of (McCollum et al., 2018). The transduction is as follows, starting with the left-to-right function:

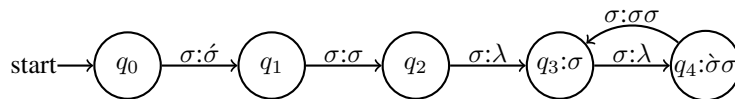


Figure 3: right-to-left transducer for sour grapes stress map

This transducer places primary stress on the first syllable. When reading a string longer than three syllables, it waits. In strings of odd length, no secondary stress is added to the output. In strings of even length, however, a secondary stress is added to the penult.

(29)	6-syllable	7-syllable
input	$\underline{\sigma} \underline{\sigma} \underline{\sigma} \underline{\sigma} \underline{\sigma} \underline{\sigma}$	$\underline{\sigma} \underline{\sigma} \underline{\sigma} \underline{\sigma} \underline{\sigma} \underline{\sigma} \underline{\sigma}$
output	$\acute{\sigma} \underline{\sigma} \underline{\lambda} \underline{\lambda} \underline{\sigma} \underline{\lambda} \grave{\sigma}$	$\acute{\sigma} \underline{\sigma} \underline{\lambda} \underline{\lambda} \underline{\sigma} \underline{\lambda} \underline{\sigma} \underline{\sigma} \sigma$
	$\acute{\sigma}\sigma\sigma\grave{\sigma}$	$\acute{\sigma}\sigma\sigma\sigma\sigma\sigma$

The function thus encodes information about the length of the string into the penult—in even syllable forms it will be stressed, and in odd syllable forms it will not. Now reading right-to-left, the second function takes advantage of this.

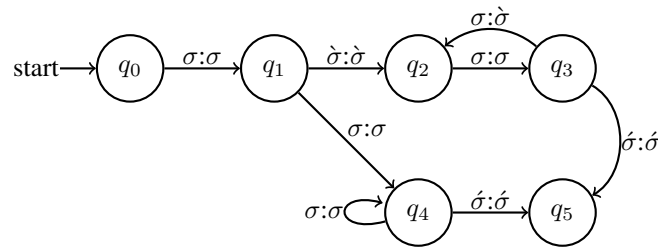


Figure 4: right-to-left transducer for sour grapes stress map

Reading right to left and taking the output of the first transducer as its input, the second transducer uses information left by the first to make a decision. In even syllable forms, it encounters a stressed penult, and so enters a loop between states two and three where it applies alternating secondary stresses until ending. In odd syllable forms the unstressed penult leads it to state four, where it makes no changes, only outputting unstressed syllables. Both loops end when the primary stress is reached and output faithfully.

(30)	6-syllable	7-syllable
input	$\acute{\sigma} \ \underline{\sigma} \ \underline{\sigma} \ \underline{\sigma} \ \acute{\sigma} \ \underline{\sigma}$	$\acute{\sigma} \ \underline{\sigma} \ \underline{\sigma} \ \underline{\sigma} \ \underline{\sigma} \ \underline{\sigma} \ \underline{\sigma}$
	$\acute{\sigma} \ \underline{\sigma} \ \grave{\sigma} \ \underline{\sigma} \ \grave{\sigma} \ \underline{\sigma}$	$\acute{\sigma} \ \underline{\sigma} \ \underline{\sigma} \ \underline{\sigma} \ \underline{\sigma} \ \underline{\sigma} \ \underline{\sigma}$
output	$\acute{\sigma} \ \sigma \ \grave{\sigma} \ \sigma \ \grave{\sigma} \ \sigma$	$\acute{\sigma} \ \sigma \ \sigma \ \sigma \ \sigma \ \sigma \ \sigma$

This is exactly the kind of interaction using intermediate markup that definitions of WD aim to exclude. If the order in which the functions apply is reversed, it is not difficult to see that a different output is obtained for even syllable words. If the right-to-left transducer reads the string first, it will never find a stressed penult. Lacking this hint from the left-to-right transducer, there is no indication that it should enter the loop where iterative application of secondary stress occurs. The order of the functions matter and so this pathological sour grapes stress mapping is excluded by McCollum et al. (2018)’s definition of WD.

It is possible that WD is an appropriate function class for describing stress assignment, but this must be worked out further. Are there pathological WD stress mappings? Is there a more restrictive class corresponding to stress that is as of yet undefined? What is a logical characterization of WD functions? These questions and others are topics for future research.

5 Discussion

Most, if not all QI stress assignment is subsequential. Non-iterative patterns can be described with a QF transduction, indicating that they belong to the ISL class of functions. Researchers looking to build theories of QI stress benefit from this explicit formal definition—no functions beyond the power of ISL need be proposed. Iterative stress, when the iteration occurs in one direction, is also subsequential. This is true even of so-called “bidirectional” systems such as Garawa. A notable possible exception is all light-syllable forms in Cahuilla, a true bidirectional pattern that requires recursive use of both p and s . The QI-ness of this pattern is up for debate. One might argue that the requirement for primary stress to land on the root-initial syllable is itself a kind of QS generalization, being sensitive to a specific kind (or position) of syllable. One might also argue that since the language counts mora i.e. weight to determine location of stress, it is QS—even if syllable weight is equal in light-only forms.

For QS stress assignment, the subsequential boundary bisects part of the typological space—DTO patterns such as LHOR are subsequential, while DTS patterns such as LHOL are not. This is a counter-intuitive result. It seems reasonable to assume that DTO would be more complex, since stress gravitates towards opposite edges, but the definition of the transduction shows this is not the case. Search of the string for heavy syllables proceeds in the same direction for DTO patterns, but in opposite directions for DTS patterns.

A question that arises is *why* some stress functions should be non-subsequential. Jardine (2016) proposes that tonal processes can access non-subsequential functions. We take the results of this paper as evidence that this should be extended to all suprasegmental processes. This may be representational—if tone and stress are defined over multi-tiered representations, then the computational power needed to parse these representations may be higher than is needed for simpler segmental phenomena (with the possible exception of vowel harmony (McCollum et al., 2018)). Future work will further articulate this hypothesis.

Another point of interest is the difference between analyses of stress as a phonotactic pattern i.e. formal *languages*, as in Heinz (2009) and Rogers et al. (2013), versus analyses of stress assignment as a string-to-string function, as in this paper. In Rogers et al., most stress patterns fall into sub-Star Free (SF) complexity classes, SF being describable by First Order (FO) logic with the precedence relation. So while most stress patterns as *stringsets* are of relatively low complexity, the corresponding stress *functions* do not necessarily preserve their complexity properties. Further investigating the relationship between stress stringsets and stress functions may help us understand the relationship between formal language complexity and function complexity.

The distinction between DTS and DTO patterns also raises questions of learnability. Oncina et al. (1993) show that subsequential functions are learnable from positive data. If DTS is not subsequential, this then means that learners of such languages are undertaking a different kind of task than learners of DTO languages.

A final question is what other non-subsequential stress patterns exist. Are they really exceptions to a general rule of subsequentiality, or are they more common than initially thought? Hindi, for example, is known to have a three-way weight distinction that determines how stress is assigned (Hayes, 1995). Is such a pattern also non-subsequential? As the distinction is three-way, it is possible the stress function may rely on information an unbounded distance to the left and right. These and other questions will hopefully come into focus in the near future.

6 Conclusion

This paper analyzed stress assignment as a function from a logical, computational perspective. The goal was to understand the complexity of different kinds of stress. Taking the subsequential class as a hypothetical boundary, we introduced logical transductions using implicit, recursive definitions, but limited to predecessor *or* successor functions, to model the input-output map.

The main results are that most, if not all, QI stress is subsequential and that DTO stress is less complex than DTS stress, which is non-subsequential. This places DTS stress assignment among circumambient processes in terms of complexity (Jardine, 2016), suggesting that stress has access to more powerful functions than most other phonological processes.

References

- Chandlee, Jane (2014). *Strictly Local Phonological Processes*. PhD Dissertation, University of Delaware.
- Chandlee, Jane & Adam Jardine (2019). Quantifier-free monadic least-fixed point transductions and sequential functions. Unpublished ms., Haverford College and Rutgers University.
- Chandlee, Jane & Steven Lindell (forthcoming). A logical characterization of strictly local functions. Heinz, Jeffrey (ed.), *Doing Computational Phonology*, OUP.
- Courcelle, Bruno (1994). Monadic second-order definable graph transductions: a survey. *Theoretical Computer Science* 126, 53–75.
- Decsy, Gyula (1966). Yurak chrestomathy. *Uralic and Altaic Series* 50.
- Furby, Christine (1974). *Garawa phonology*, vol. Series A. Pacific Linguistics, Australian National University.
- Gordon, Matthew (2002). A factorial typology of quantity-insensitive stress. *Natural Language & Linguistic Theory* 20:3, 491–552.
- Hansen, Kenneth & L. E. Hansen (1969). Pintupi phonology. *Oceanic Linguistics* 8, 153–170.
- Hao, Sophie & Samuel Andersson (2019). Unbounded stress in subregular phonology. *Proceedings of SIGMORPHON 16*.
- Hayes, Bruce (1995). *Metrical Stress Theory: Principles and Case Studies*. Chicago: The University of Chicago Press.
- Heinz, Jeffrey (2009). On the role of locality in learning stress patterns. *Phonology* 26:2, 303–351.
- Heinz, Jeff (2015). The computational nature of phonological generalizations. Plank, Larry M. Hyman & Frans (ed.), *Phonological typology*, De Gruyter Mouton, Berlin & Boston.
- Heinz, Jeffrey & Regine Lai (2013). Vowel harmony and subsequentiality. Kornai, Andras & Marco Kuhlmann (eds.), *Proceedings of the 13th Meeting on Mathematics of Language*, Sofia, Bulgaria.
- Hyman, Larry (1977). On the nature of linguistic stress. Hyman, L. (ed.), *Studies in Stress and Accent*, USC, Los Angeles, California.
- Idsardi, William (1988). *The Computation of Prosody*. PhD Dissertation, MIT.
- Jardine, Adam (2016). Computationally, tone is different. *Phonology* 33(2), 385–405.
- Johnson, Douglas (1972). Formal aspects of phonological description.
- Kager, Rene (2007). Feet and metrical stress. de Lacy, Paul (ed.), *The Cambridge Handbook of Phonology*, Cambridge University Press, Cambridge, England, 195–227.
- Kaplan, Ronald & Martin Kay (1994). Regular models of phonological rule systems. *Computational Linguistics* 20, 331–378.
- Koser, Nate & Adam Jardine (2019). The complexity of optimizing over strictly local constraints. *Proceedings of PLC 43*.
- Koser, Nathan, Christopher Oakden & Adam Jardine (2019). Tone association and output in non-linear structures. Hout, Katherine, Anna Mai, Adam McCollum, Sharon Rose & Matthew Zaslansky (eds.), *Supplemental Proceedings of the 2018 Annual Meeting on Phonology*, LSA.
- Levin, Juliette (1988a). Bi-directional foot construction as a window on level ordering. Hammond, M. & M. Noonan (eds.), *Theoretical Morphology*, 339–352.
- Luo, Huan (2017). Long-distance consonant agreement and subsequentiality. *Glossa* 2:52.
- McCollum, Adam, Eric Bakovic, Anna Mai & Eric Meinhardt (2018). The expressivity of segmental phonology and the definition of weak determinism. [ling.auf.net/lingbuzz/004197](http://lingbuzz/004197).
- Mielke, Jeff (2004). *P-base: database of sound patterns in 500+ languages*. <http://pbase.phon.chass.ncsu.edu/query>.
- Mohri, Mehryar (1997). Finite-state transducers in language and speech processing. *Computational Linguistics* 23, 269–311.
- Oncina, J., P. Garcia & E. Vidal (1993). Learning subsequential transducers for pattern recognition tasks. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 15, 448–458.
- Payne, Amanda (2017). All dissimilation is computationally subsequential. *Language: Phonological Analysis* 93:4, e353–371.
- Prince, Alan (1985). Improving tree theory. *BLS* 11, 471–490.
- Rogers, James (1997). Strict LT2 : Regular :: Local : Recognizable. Retoré, Christian (ed.), *Logical Aspects of Computational Linguistics: First International Conference, LACL '96 Nancy, France, September 23–25, 1996 Selected Papers*, Springer Berlin Heidelberg, Berlin, Heidelberg, 366–385.
- Rogers, James, Jeffrey Heinz, Margaret Fero, Jeremy Hurst, Dakotah Lambert & Sean Wibel (2013). Cognitive and sub-regular complexity. Morrill, Glyn & Mark-Jan Nederhof (eds.), *Formal Grammar*, Springer Berlin Heidelberg, Berlin, Heidelberg, 90–108.
- Seiler, Hansjakob (1977). *Cahuilla Grammar*. Malki Museum Press, Banning, California.
- Street, Chester S. & Gregory P. Mollinjin (1981). The phonology of murinbata. *Australian phonologies: Collected papers* 183–244.
- Wilson, Colin (2003). Analyzing unbounded spreading with constraints: marks, targets, and derivations.
- Wilson, Colin (2006). Unbounded spreading is myopic.