Computational Characterizations of Vowel Harmony Patterns and Pathologies

Brian Gainor, Regine Lai, and Jeffrey Heinz
University of Delaware

1. Introduction

This paper provides a computational analysis of 39 attested vowel harmony patterns in a recent typological analysis (Nevins, 2010), in addition to two unattested harmony patterns which have attracted considerable discussion: 'majority rules' (Lombardi, 1999; Baković, 2000) and 'sour grapes' (Padgett, 1995). It is shown that these attested patterns, unlike the two unattested ones, are *subsequential*. We therefore hypothesize that subsequentiality is a *universal* property of vowel harmony. Furthermore, this property is stronger (i.e. more restrictive) than previously established computational universals.

The two unattested patterns have been referred to as 'pathologies' (Wilson, 2003) because they are predicted to occur through the interaction of standard constraints within Optimality Theory (OT) (Prince & Smolensky, 2004). The fact that these pathological patterns are unattested in natural languages could either be an accidental gap or due to principled factors. The results of this paper are consistent with the latter; in particular the hypothesis that phonological patterns must be subsequential.

This paper relies on finite state grammar (FSG) representations of phonological patterns. There are some advantages to using finite state grammars in analyses of phonology. First, they are adequately expressive. Johnson (1972) and Kaplan & Kay (1994) showed that SPE-style rewrite rules and grammars (Chomsky & Halle, 1968) describe regular relations and since SPE-style grammars are adequately expressive, it follows that FSGs are likewise adequately expressive. Second, operations to manipulate and combine these machines are well understood since FSGs form a fundamental chapter of theoretical computer science (Hopcroft et al., 2001). Third, and perhaps most importantly, it becomes possible to obtain insights with FSGs that are much difficult, if not impossible, to realize in either SPE or OT (Kaplan & Kay, 1994; Riggle, 2004; Heinz, 2009).

2. Background

Phonological grammars which map underlying forms to surface forms describe mathematical relations. It is of interest to determine which properties are necessary and sufficient to make such relations *phonological*. Readers are referred to Heinz (2011a,b) for further background.

Regular relations are relations describable with *finite-state transducers* (FSTs) and are a proper subclass of all logically possible relations. This paper focuses on *subsequential relations*, which are describable with *subsequential finite-state transducers* (SFSTs) and which are a proper subclass of the regular relations. Although both types are discussed, only SFSTs are defined formally here; for formal definitions of FSTs, readers are referred to Kaplan & Kay (1994) and Beesley & Kartunnen (2003).

2.1. Regular relations

FSTs are machines with only finitely many states. For example, consider the SPE-style vowel nasalization rule in (1a) and some mappings exemplifying it (1b).

(1) a. V $\longrightarrow$ [+nasal] / [+nasal] ___

b. i /ba/ $\rightarrow$ [ba] iii. /nana/ $\rightarrow$ [nãnã]

ii. /naga/ $\rightarrow$ [nãga] iv. /naikanai/ $\rightarrow$ [nãĩkanãĩ]

*We thank the audience at WCCFL, especially Diana Archangeli, Mike Hammond, and Paul Smolensky for valuable feedback. We also thank participants in the Fall 2010 Formal Models in Phonology seminar at the University of Delaware.

The FST in Figure 1 is a machine which implements this rule; it takes a phonological form as input and then outputs the phonological form which reflects the application of rule (1a). The machine processes

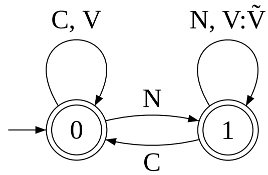

Figure 1: A finite state transducer computing the same function as the nasalization rule. V indicates an oral vowel, Ṽ a nasal vowel, N a nasal consonant, and C a non-nasal consonant.

strings one letter at a time, beginning in state 0. Transitions between states read an input symbol and output a symbol. For example, if a transition reads input a and outputs b, the transition is labeled $a{:}b$. However, following the notation in Beesley & Kartunnen (2003), the colon notation is only used when the output differs from the input, otherwise just a single symbol is used. So instead of $a{:}a$, we just write a. Multiple transitions between states are shown pictorially with a single arc with the labels of each transition separated by commas. The output of an input string is found by reading the input one letter at a time, and producing the output according to the transitions followed. If, at the end of reading the whole input string, the machine is in a *final* state (those indicated with double peripheries) then the process ends and the output string stands. Otherwise the process rejects the input string and there is no output.

For example, the transition sequence of the machine when processing (1b.ii) /naga/ → [nãga] is $0 \xrightarrow{\text{n}} 1 \xrightarrow{\text{a:ã}} 1 \xrightarrow{\text{g}} 0 \xrightarrow{\text{a}} 0$. More generally, vowels remain oral before as long as the FST stays in state 0. But once the machine observes a nasal, the FST enters state 1. Consequently, all subsequent vowels are nasal, until a non-nasal consonant is encountered, which returns the machine to state 0. This machine thus accepts (equivalently, generates or recognizes) the input/output pair /naga/ → [nãga].

The set of input/output pairs that this machine accepts is a relation, and a relation is regular if and only if there is a FST which accepts all and only the pairs in that relation.

2.2. Subsequential Relations

Subsequential relations are a proper subclass of the regular relations, and are exactly those accepted by a special kind of transducer. There are two kinds of subsequential relations: *left* and *right*, which are accepted by *subsequential* and *reverse subsequential* transducers, respectively (Mohri, 1997).

Subsequential transducers differ in three ways from standard transducers. First, they are *deterministic on the input*. This means at any given state, for any given input symbol, there is at most one state the machine can transition to. Second, every state is final. Third, a string is associated with each state. This string is appended to the end of the output when a machine finishes in that state. In every state of every machine considered here, this string is the empty string (λ) and so it plays no important role here.

Before we define subsequential transducers and relations formally, we will first look at an example. Kikongo has a long-distance nasalization rule Ao (1991). Suffixes with underlying /d,l/ surface faithfully unless a nasal occurs in the stem, in which case they surface as [n]. A first approximation of this alternation could be stated as the SPE rewrite rule shown in (2).[1]

(2) Kikongo long-distance nasalization: [+voice,+coronal] ⟶ [+nasal] / [+nasal]X* ____

a. /sakid-ila/ → [sakid-ila] ‘to congratulate for’
b. /mant-ila/ → [mant-ina] ‘to climb for’
c. /kudumukis-ila/ → [kudumukis-ina] ‘to cause to jump for’
d. /tu-nik-idi/ → [tunik-ini] ‘we ground’

This rule can be described with a subsequential transducer as shown in Figure 3.

We read diagrams of subsequential transducers just as before. Following Beesley & Kartunnen (2003), the [?] symbol represents any segment other than /d,l,m,n/. As shown in Figure 2 when in state 1,

[1]Ao (1991) notes that nasals in NC clusters do not trigger the assimilation, but we ignore this detail here as our primary goal here is to illustrate subsequential transducers.

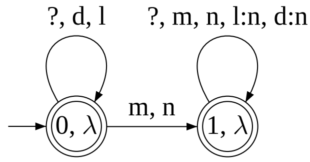

Figure 2: A finite state transducer computing the same function as the nasalization rule.

if the input is /l/ or /d/, then [n] is the output. Everywhere else, the output is exactly the same as the input. For example, the transition sequence for the input /mantila/ is $0 \xrightarrow{\text{m}} 1 \xrightarrow{\text{a}} 1 \xrightarrow{\text{n}} 1 \xrightarrow{\text{t}} 1 \xrightarrow{\text{i}} 1 \xrightarrow{\text{l:n}} 1 \xrightarrow{\text{a}} 1$. Since state 1 is a final state, the string associated with this state (the empty string λ) appended to the end of the output sequence yielding [mantina].

Formally, a subsequential transducer is a machine $\mathcal{M} = (Q, X, Y, q_0, T, \sigma)$ where Q is a set of states, q_0 is the initial state, X is the input alphabet, Y is the output alphabet, and T is the transitions, which are a subset of $Q \times X \times Y^* \times Q$ such that if (q, x, u, r) and (q, x, v, s) belong to δ then $u = v$ and $r = s$ (the determinism condition). The function $sigma$ maps states to output strings ($\sigma : Q \to Y^*$).

A *path* through a machine $\mathcal{M}$ is a sequence of transitions $\pi = (q_0, x_1, y_1, q_1)(q_1, x_2, y_2, q_2)\ldots(q_{n-1}x_n, y_n, q_n)$. Due to the determinism condition, such a sequence can be written as $\pi = (q_0, x_1x_2 \ldots x_n, y_1y_2 \ldots y_n, q_n)$. Let $\Pi_{\mathcal{M}}$ denote all possible paths over $\mathcal{M}$. For all $(q_0, x, y, q) \in \Pi_{\mathcal{M}}$, let $\mathcal{M} : X^* \to Y^*$ be the partial function such that $\mathcal{M}(x) = y\sigma(q)$. A relation is *left subsequential* is there is a subsequential transducer accepting it.

The *reverse* of a string u is u^r. For example, $abcd^r = dcba$. The reverse of a relation is $R^r = \{(x^r, y^r) \mid (x, y) \in R\}$. A relation is *right subsequential* iff its reverse relation is left subsequential. A relation is *subsequential* if and only if it is either left or right subsequential.

To illustrate right subsequentiality, consider the language *Ognokik* which is like Kikongo above except nasalization of /l,d/ occurs only if a nasal occurs *later* in words.

(3) Ognikik long-distance nasalization: [+voice,+coronal] ⟶ [+nasal] / ___ X*[+nasal]

a. /ali-dikas/ → [ali-dikas]
b. /ali-tnam/ → [ani-tnam]
c. /idi-kin-ut/ → [ini-kinut]

The reader may verify that there is no left subsequential transducer which describes this pattern. However, the reverse relation of this pattern is left subsequential (it is the SFST in Figure 2); thus, this pattern is right subsequential.

2.3. Research Questions

As mentioned, subsequential relations are a proper subclass of regular relations (Berstel, 1979; Oncina et al., 1993; Mohri, 1997). While it is known that regular relations are sufficiently expressive to account for individual phonological generalizations (Kaplan & Kay, 1994), it is not known whether individual phonological generalizations belong to the smaller class of subsequential relations. Thus the hypothesis that subsequentiality is a universal property of individual phonological generalizations constitutes a strictly stronger hypothesis.

This paper shows this stronger hypothesis is viable with respect to vowel harmony. Studies of metathesis (Chandlee et al., 2011) and local processes like epenthesis, deletion and substitution in (Koirala, 2010) also indicate the correctness of this hypothesis.

3. Attested vowel harmony patterns are subsequential

3.1. Typology of attested patterns

Vowel harmony (VH) has been extensively studied by several researchers (Ringen, 1988; Archangeli & Pulleyblank, 1994; van der Hulst & van de Weijer, 1995; Baković, 2000; Krämer, 2003; Nevins, 2010). This paper studies 39 VH patterns from Nevins (2010). We drew patterns from this book because each pattern has been computationally implemented, data files existed for each pattern, and one of the authors (BG) has extensive experience with these files.[2] The results presented here are addressed in the context of dominant/recessive theories of vowel harmony (Baković, 2000; Krämer, 2003) in the conclusion.

Nevins' theory of vowel harmony is presented in a Principles and Parameters setting. It assumes that morphemes which undergo vowel harmony contain segments underlyingly underspecified for the feature in question. These features receive their values via an operation, Harmonize, which copies the value from the closest visible source. Which segments are visible are determined by a number of parameters, the most important of which are summarized in Table 1. Once set, the Harmonize algorithm is fully specified. Nevins shows that these parameters account for virtually all types of vowel harmony patterns, including those with transparent or opaque vowels, and with or without bounding. For the present discussion the only relevant parameter will be the directionality of the search since this corresponds directly to whether the pattern is accepted by a left or right subsequential transducer.

Direction of search	Sonority threshold	Various feature requirements
Search domain	Type of distance bound	Distance bound

Table 1: Important Parameters of Harmonize (Nevins, 2010).

3.2. Methodology

As mentioned, each pattern discussed in Nevins (2010) has a data file. These 44 data files document the phonological inventory, values for each of the Harmonize parameters and search variables, and also include example input-output pairs for each language from the literature. Since we are primarily interested in purely phonological VH patterns, we excluded Chumash (consonantal harmony), Terena (morphemic harmony), and three VH patterns from consideration where Nevin's analysis depends upon morphological factors (Akan, Kalenjin, and Londengese). This left 39 patterns for study.

We created a Python script which took as input one such data file and output a Xerox finite-state tools (xfst) script, which defined a finite-state transducer computing the same vowel harmony pattern. Xfst is a software package which implements finite-state acceptors and transducers for use in morphology and phonology (Beesley & Kartunnen, 2003). The resulting transducers were tested for accuracy against the example input-output pairs for each language automatically, and checked for subsequentiality manually.

It was found that all but one of the vowel harmony patterns are either left or right subsequential, and that the remaining "bidirectional" pattern of Woleaian can be straightforwardly factored into a left subsequential relation and a right subsequential one.

3.3. Left-to-right vowel harmony

The majority of vowel harmony patterns in (Nevins 2010) are left-to-right. In such a system, the "direction of search" parameter is set to leftwards, and the search begins at the segment preceding the underspecified segment. On each following step, the search algorithm moves one segment to the left, until it encounters a relevant segment (source or blocker), or reaches the beginning of the word.

An example of such a pattern is that found in Classical Mongolian. The ablative suffix is /VčV/, where V is a vowel specified for every feature except [back], resulting in [a] if it follows a [+back] vowel, and [e] if it follows a [−back] one (except for [i], which is transparent).

(4) (Nevins, 2010:82)

a.	ulus	ulus-ača	'nation-abl.'	d.	mören	mören-eče	'river-abl.'
b.	aman	aman-ača	'mouth-abl.'	e.	morin	morin-ača	'horse-abl.'
c.	üker	üker-eče	'ox-abl.'				

A representation of the FST output for this vowel harmony pattern in Mongolian is shown in Figure 3.

[2] The Harmonizer algorithm described by Nevins is available at `http://mitpress.mit.edu/supplementary/vowel_harmony/`.

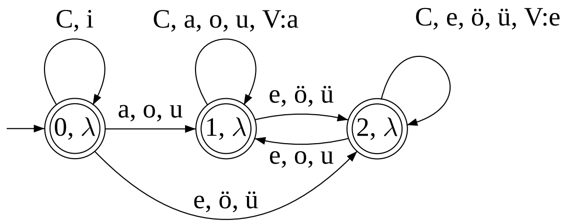

Figure 3: A subsequential FST for the vowel harmony pattern in Mongolian.

The transducer in (3) is left subsequential, because it is deterministic on the input (i.e. from any state, there is only one outgoing transition for any particular input), and every state is final. In fact, all left-to-right VH patterns from Nevins (2010) are left subsequential. This is because a FST implementing such a pattern only need keep track of the most-recently seen value of the relevant feature (in visible segments) so every state will be final. The FST will also be deterministic, since each input will lead to only one state: either the current state if the segment is transparent to the search, or a different state if it is not. Since the machine is deterministic and each state is final, it is subsequential.

3.4. Right-to-left vowel harmony

If the "direction of search" parameter is set to rightwards, then right-to-left vowel harmony patterns are observed. An example is Yoruba. In Standard Yoruba, high vowels are always [+ATR] and act as opaque blockers for [ATR] harmony. The disyllabic words in (5) all show [ATR] harmony, although it is impossible to determine the direction of the harmony (Nevins, 2010:117).

(5) Yoruba [ATR] Harmony in Disyllabic Roots.

a.	éwé	'lip'	c.	èrò	'crowd'	e.	ɛ̀ɡɛ́	'cassava'	g.	ɛ̀fɔ́	'vegetable'
b.	olè	'thief'	d.	òd͡ʒò	'rain'	f.	ɔsɛ	'soap'	h.	ɔwɔ́	'hand'

In (6), the middle vowel of the trisyllabic words is high, and is thus [+ATR]. However, these vowels also block the first vowel from participating in [ATR] harmony, which shows the harmony is right-to-left.

(6) Standard Yoruba Harmony in Trisyllabic Roots.

a.	orúkɔ	'name'	c.	ewúrɛ́	'goat'	e.	òtítɔ́	'truth'
b.	èlùbɔ́	'yam flour'	d.	odídɛ	'parrot'			

The transducer for Yoruba is shown in Figure 4. As can be seen, not all of its states are final. Additionally, the transducer is not deterministic on the input, as there are two transitions leaving state 0 which both have E and O (the underspecified mid vowels) as inputs.

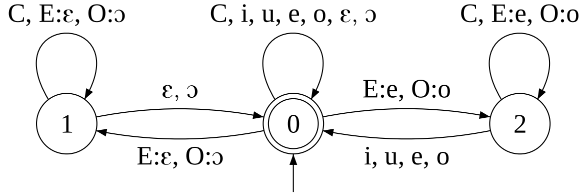

Figure 4: A non-deterministic FST for the vowel harmony pattern in Standard Yoruba. /E,O/ indicate underspecified mid-vowels.

Of course, the fact that this transducer is not left subsequential does not prove that there is no such transducer implementing this relation. However, any transducer which implemented such a relation could not be deterministic. To see why, consider a language in which mid vowels participate in vowel harmony but high vowels are transparent, rather than opaque (the Ife dialect of Yoruba exhibits just such a pattern). Once the machine saw a needy mid vowel on the input, it would have to remember that segment until it saw another mid vowel. Since a finite-state transducer cannot go back and "rewrite" a segment preceding one it's already outputted, it would have to remember every segment which comes between the underspecified segment and its eventual source. Since high vowels are transparent to the search, it could in theory continue to any length, requiring an unbounded amount of "memory". In other words, such a machine would in fact require an infinite number of states! Since this is not possible, any transducer implementing such a pattern cannot be deterministic.

On the other hand, the *reverse relation* of the Yoruba pattern (i.e. one in which both input and output strings are reversed) results in the transducer shown in Figure 5. This transducer, like the one in Figure 3, is deterministic and contains only final states. It is left subsequential, so its reverse, the actual Yoruba relation, is right subsequential. Other right-to-left patterns are likewise right subsequential by the same arguments that showed left-to-right patterns to be left subsequential.

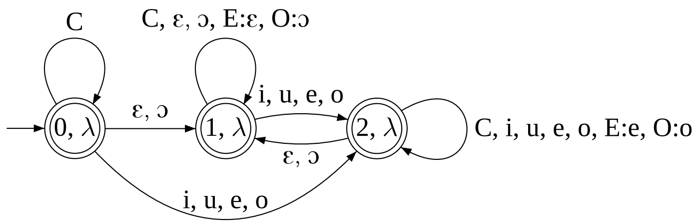

Figure 5: A subsequential transducer for the reverse of the vowel harmony pattern in Yoruba. /E,O/ indicate underspecified mid-vowels.

3.5. Bidirectional vowel harmony

Of the 39 VH patterns investigated in Nevins (2010), only one resulted in a FST which was neither left nor right subsequential. This was the bidirectional pattern in Woleaian. This apparent contradiction to the hypothesis is resolved by recognizing that the bidirectional harmony pattern of Woleaian naturally factors into a left-to-right harmony pattern and a right-to-left one. While the intersection of two such relations may be neither left nor right subsequential, the generalization still holds: individual VH patterns are subsequential. In other words, bidirectional patterns simply instantiate both a left-to-right pattern and a right-to-left pattern simultaneously. Factoring bidirectional patterns in this way obviates the need for this third category while maintaining the subsequentiality hypothesis.

3.6. Summary

All of the attested vowel harmony patterns in Nevins (2010) are either left subsequential or right subsequential relations, or are the intersection of such relations.

4. Unattested Vowel harmony patterns

This section focuses on two unattested vowel harmony patterns which have been well-studied by previous researchers because they are predicted to occur by classic OT: Majority Rules (Lombardi, 1999; Baković, 2000) and Sour Grapes (Padgett, 1995; Wilson, 2003).

4.1. The Majority Rules and Sour Grapes Patterns

Majority Rules (MR) describes a phenomenon in which vowels harmonize to the feature shared by the *majority* of segments. Assuming, for exposition, an alphabet which only contains '$+$' (for segments which are [$+$F]) and '$-$' (for segments which are [$-$F]). Then MR would map an underlying form like /$+--$/ to [$---$] because segments with [$-$F] outnumber the [$+$F] segments in the underlying form. Likewise MR maps underlying forms like /$++-$/ to [$+++$] because the segments with [$+$F] outnumber the [$-$F] segments in the underlying form.

Sour Grapes (SG) patterns are more complicated because they involve opaque vowels, i.e. *blockers*. Again assume an alphabet which only contains '$+$' and '$-$' symbols and let the blocker be specified as [$-$F] (we denote such vowels B_-). For concreteness, we assume SG spreads the value of [F] from left to right if no blocker is present in the word. Thus underlying /$+--$/ would map to [$+++$]. However, underlying /+−B_/ surfaces faithfully because of the blocker. The name "Sour Grapes" comes from the attitude the process is said to exhibit: It gives up on spreading if it cannot spread all the way (i.e. when the spreading is blocked by a Blocker). Also note that blockers start a new harmony domain.

MR and SG are summarized in Table 2 along with a canonical left-to-right pattern.

	UR	Directional (L-to-R)	Majority Rules	Sour Grapes
1.	/+ + −/	[+ + +]	[+ + +]	[+ + +]
2.	/− − +/	[− − −]	[− − −]	[− − −]
3.	/+ − −/	[+ + +]	[− − −]	[+ + +]
4.	/− + ++/	[− − −−]	[+ + ++]	[− − −−]
5.	/+ + B_ −/	[+ + B_ −]	[− − B_ −]	[+ + B_ −]
6.	/B_ + −/	[B_ − −]	[B_ − −]	[B_ − −]
7.	/− + B_/	[− − B_]	[− − B_]	[− − B_]
8.	/+ − B_/	[+ + B_]	[− − B_]	[+ − B_]
9.	/+ − − − − − B_/	[+ + + + + + B_]	[− − − − − − B_]	[+ − − − − − B_]

Table 2: The underlying forms and surface forms predicted by left-to-right harmony, Majority Rules, and Sour Grapes. Shaded regions indicate where the outputs of the Majority Rules and Sour Grapes patterns differ from the canonical Left-to-Right pattern.

Classic OT predicts that both MR and SG are possible patterns. For MR, the constraint AGREE ranks above IDENT which outranks ANCHOR-L as shown in Table 3. Candidate (b) is not chosen under this ranking because it has one more IDENT violation than candidate (c).

/+ − −/	AGREE	IDENT	ANCHOR-L
a. + − −	*!		
b. + + +		**!	
c. ☞ − − −		*	*

Table 3: OT tableaux illustrating Majority Rules with the underlying form /+ - - /.

Classic OT also predicts SG with the ranking ANCHOR-L above AGREE above IDENT and assuming that a higher ranked constraint prohibits the blocker from surfacing as [+F]. Table 4 illustrates.

/+ − B_/	ANCHOR-L	AGREE	IDENT
a. ☞ + − B_		*	
b. − − B_	*!		*
c. + + B_		*	*!

Table 4: OT tableaux illustrating Sour Grapes with the underlying form /+ - B_ /

4.2. Complexity of Majority Rules and Sour Grapes

Majority rules is not a regular relation. Every regular relation is accepted by a transducer with finitely many states. However, whether the first segment surfaces faithfully or not cannot be determined until the difference in the number of [+F] segments and [-F] segments is made. This difference can grow arbitrarily large, and a state will be required for each possible difference. Thus the set of states cannot be finite. Therefore, MR is not regular.

On the other hand, Sour Grapes is regular because it can be represented by a finite-state transducer as shown in Figure 6. However, this transducer is not deterministic. In Figure 6 in state 2, if the input segment is a [−], transitions to either state 3 or state 4 can be followed.

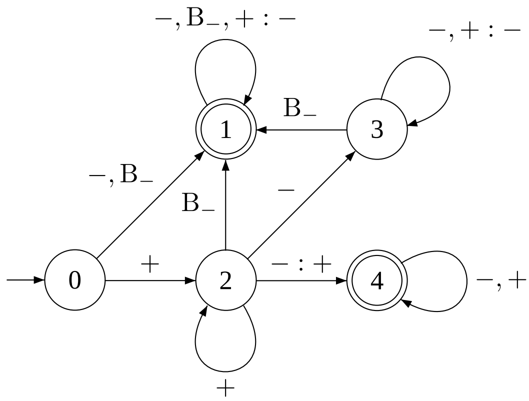

Figure 6: A nondeterministic FST computing “Sour Grapes”.

Although SG is regular, it is not subsequential. The proof of this claim is presented elsewhere (Heinz & Lai, 2011). The proof uses the fact that every subsequential relation has a canonical form; i.e. there is a unique (up to isomorphism) subsequential transducer with fewest states (Oncina et al., 1993). Heinz and Lai show that this canonical subsequential transducer for SG must have infinitely many states in an argument similar to the one made for MR. Hence, no finite-state machine which is subsequential (either forward or reverse) recognizes this pattern.

5. Conclusion

To conclude, attested VH patterns are subsequential, and two key, unattested pathological vowel harmony patterns are not. These findings suggest a tighter computational boundary for phonological patterns than previously thought. Additionally, this boundary provides a principled explanation for the absence of the two pathological vowel harmony types. Figure 7 summarizes these results.

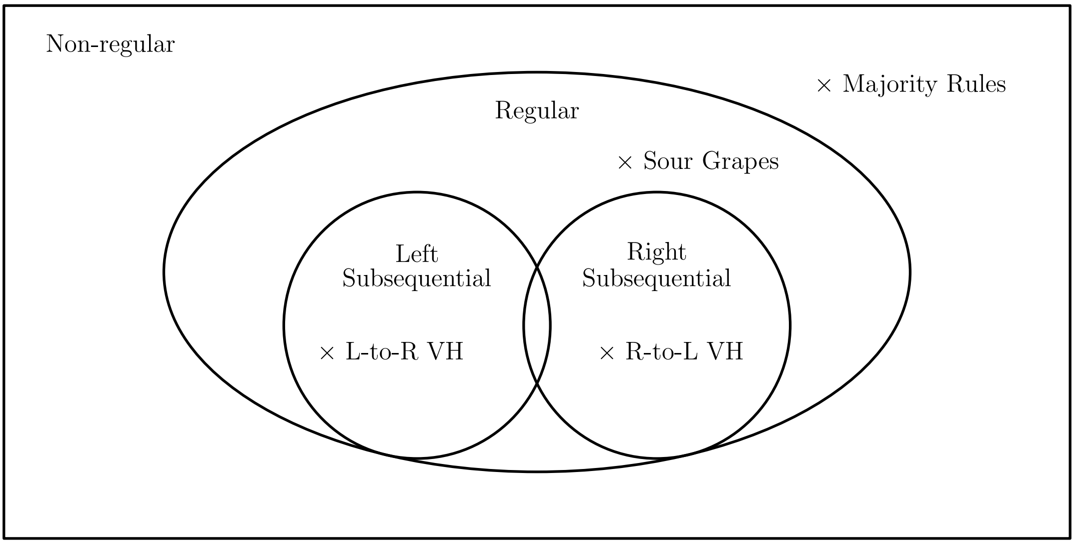

Figure 7: Vowel harmony patterns located within computational complexity classes.

Nevin’s theory of vowel harmony relies on directionality and underspecification as primitive theoretical elements. Do these results speak to theories that reject these assumptions, such as theories which view VH patterns as stem-controlled or dominant-recessive (Baković, 2000; Krämer, 2003)?

By relying on underspecified lexical representations, Nevin’s theory also requires Morpheme Structure Rules (MSRs) to achieve the same level of explanation as these other theories. For example, while no disyllabic forms in Standard Yoruba permit mid vowels which disagree in ATR, this fact is due to the lexicon in Nevins' account, and not to the VH process itself. As the FSTs in Figures 4 and 5 make clear, Nevin's Yoruba analysis maps underlying /odɛ/ to [odɛ]. However, in Baković's (2000:146) analysis, /odɛ/ maps to [ɔdɛ]. In Nevin's analysis, underlying /odɛ/ must first be converted to /Odɛ/ by a MSR to explain why there are no surface forms like [odɛ].

If the Yoruba MSRs are right subsequential, then these other analyses of Yoruba VH describe right subsequential mappings too. This is because subsequentiality is closed under functional composition (Mohri, 1997:Theorem 1), and these other analyses describe mappings that are obtained by composing the MSRs with Nevins' analyses.

Future research includes definitively evaluating the subsequentiality of the mappings provided by other theories of VH, verifying the FSTs' subsequentiality with the determinization algorithm given in Mohri (2000), and investigating whether the subsequential boundary is psychologically real with artificial language learning experiments (Folia et al., 2010).

References

Ao, Benjamin (1991). Kikongo nasal harmony and context-sensitive underspecification. *Linguistic Inquiry* 22:1, 193–196.

Archangeli, Diana & Douglas Pulleyblank (1994). *Grounded Phonology*. Cambridge, MA: MIT Press.

Archangeli, Diana & Douglas Pulleyblank (2007). Harmony. de Lacy, Paul (ed.), *The Cambridge Handbook of Phonology*, Cambridge University Press, Cambridge, chap. 15, 353–378.

Baković, Eric (2000). *Harmony, Dominance, and Control*. Ph.D. thesis, Rutgers University, New Brunswick, NJ.

Beesley, Kenneth & Lauri Kartunnen (2003). *Finite State Morphology*. CSLI Publications.

Berstel, Jean (1979). *Transductions and Context-Free languages*. Teubner-Verlag.

Chandlee, Jane, Angeliki Athanasopoulou & Jeffrey Heinz (2011). Evidence for classifying metathesis patterns as subsequential. *Proceedings of the 29th West Coast Conference on Formal Linguistics*, Cascadilla Press. To appear.

Chomsky, Noam & Morris Halle (1968). *The Sound Pattern of English*. Harper and Row, New York.

Folia, Vasiliki, Julia Uddén, Meinou de Vries, Christian Forkstam & Karl Magnus Petersson (2010). Artificial language learning in adults and children. *Language Learning* 60, 188–220. Supplement 2.

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

Heinz, Jeffrey (2011a). Computational phonology part I: Foundations. *Language and Linguistics Compass* 5:4, 140–152.

Heinz, Jeffrey (2011b). Computational phonology part II: Grammars, learning, and the future. *Language and Linguistics Compass* 5:4, 153–168.

Heinz, Jeffrey & Regine Lai (2011). Subsequentiality is a necessary property of vowel harmony. MS, submitted.

Hopcroft, John, Rajeev Motwani & Jeffrey Ullman (2001). *Introduction to Automata Theory, Languages, and Computation*. Boston, MA: Addison-Wesley.

van der Hulst, Harry & Jeroen van de Weijer (1995). Vowel harmony. Goldsmith, John A. (ed.), *The Handbook of Phonological Theory*, Blackwell, Cambridge, Mass., and Oxford, UK, 495–534.

Johnson, C. Douglas (1972). *Formal Aspects of Phonological Description*. The Hague: Mouton.

Kaplan, Ronald & Martin Kay (1994). Regular models of phonological rule systems. *Computational Linguistics* 20:3, 331–378.

Koirala, Cesar (2010). Strictly local relations. MS.

Krämer, Martin (2003). *Vowel Harmony and Correspondence Theory*. Berlin: Mouton de Gruyter.

Lombardi, Linda (1999). Positional faithfulness and voicing assimilation in Optimality Theory. *Natural Language and Linguistic Theory* 17, 267–302.

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

Mohri, Mehryar (2000). Minimization algorithms for sequential transducers. *Theoretical Computer Science* 234, 177–201.

Nevins, Andrew (2010). *Locality in Vowel Harmony*. The MIT Press, Cambridge, MA.

Oncina, José, Pedro García & Enrique Vidal (1993). Learning subsequential transducers for pattern recognition tasks. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 15, 448–458.

Padgett, Jaye (1995). Partial class behavior and nasal place assimilation. Suzuki, K. & D. Elzinga (eds.), *Proceedings of the 1995 Southwestern Workshop on Optimality Theory*.

Prince, Alan & Paul Smolensky (2004). *Optimality Theory: Constraint Interaction in Generative Grammar*. Blackwell Publishing.

Riggle, Jason (2004). Generation, recognition and ranking with compiled OT grammars. Boston. LSA 78.

Ringen, Catherine (1988). *Vowel Harmony: Theoretical Implications*. Garland Publishing, Inc.

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

Proceedings of the 29th West Coast Conference on Formal Linguistics

edited by Jaehoon Choi, E. Alan Hogue, Jeffrey Punske, Deniz Tat, Jessamyn Schertz, and Alex Trueman

Cascadilla Proceedings Project Somerville, MA 2012

Copyright information

ISBN 978-1-57473-451-5 library binding

Ordering information

Orders for the library binding edition are handled by Cascadilla Press.
To place an order, go to www.lingref.com or contact:

Cascadilla Press, P.O. Box 440355, Somerville, MA 02144, USA
phone: 1-617-776-2370, fax: 1-617-776-2271, sales@cascadilla.com

Web access and citation information