

RESEARCH

Long-distance consonant agreement and
subsequentiality

Huan Luo

Department of Linguistics and Cognitive Science, University of Delaware, Newark, DE 19716, US
hluo@udel.edu

Johnson (1972) and Kaplan & Kay (1994) showed that phonological processes belong to the computational class of regular relations. This paper provides a computational analysis of long-distance consonant agreement and shows that it belongs to a more restricted computational class called subsequential. This paper further argues that subsequentiality is a desirable computational characterization of long-distance consonant agreement for the following reasons. First, it is sufficiently expressive. Second, it is restrictive as it accounts for the absence of pathological patterns like Majority Rules and Sour Grapes from the typology (Heinz & Lai 2013), standing in contrast to Agreement by Correspondence analysis in Optimality Theory (Rose & Walker 2004; Hansson 2007).

Keywords: computational phonology; consonant harmony; subsequentiality; long-distance phonology; finite-state transducers

1 Introduction

Long-distance consonant agreement (hereafter LDCA) is a phonological phenomenon whereby consonants of a particular type are required to agree with each other in some property. Two conditions must be met so as to be regarded as long-distance in nature: first, consonants involved in alternation are separated by at least one intervening segment; second, the intervening segments do not participate in the harmony process in any obvious way (Hansson 2001; 2010). Data in (1) show an example of LDCA process in Samala.

- | | | | |
|-----|------------------------------------------|---|----------------------------------------------------------------|
| (1) | Samala sibilant harmony Applegate (1972) | | |
| | /k-su-fojin/ | → | k-fu-fojin 'I darken it.' |
| | /s-api-tf ^h o-it/ | → | f-api-tf ^h o-it 'I have a stroke of good luck.' |
| | /s-api-tf ^h o-us-waf/ | → | f-api-tf ^h o-uf-waf 'he had a stroke of good luck.' |
| | /ha-s-xintila-waf/ | → | ha-f-xintila-waf 'his former gentile name.' |
| | /s-if-tifi-jep-us/ | → | s-is-tisi-jep-us 'they (two) show him.' |

In Samala, all sibilants agree in anteriority with the rightmost sibilant sound. For instance, /f/ in suffix /-waf/ triggers the harmony: all preceding sibilant sounds, namely, /s/ in prefix /s-/ and /s/ in suffix /-us/, become [f] to agree with it in anteriority. Notice that the alternating segments are separated by several intervening segments, but none of these are affected by the harmony process.

Based on Hansson's typological study of LDCA patterns in worlds' languages, this paper presents a novel analysis of LDCA from a computational perspective (Hansson 2010).¹ It

¹ A number of cases in Hansson's study are static cooccurrence restrictions (morpheme structure constraints, MSCs), since these cases are not manifested in alternations, this paper will not consider such cases.

has been argued that phonological processes belong to the computational class of regular relations (Johnson 1972; Kaplan & Kay 1994). A stronger hypothesis is that phonological processes in fact belong to particular subregular regions, that is, more restricted than being regular (Heinz 2009; 2010). Recent work have shown that phonological patterns such as epenthesis, metathesis, and vowel harmony are subregular (Chandlee & Heinz 2012; Heinz & Lai 2013). This paper will show that LDCA patterns also belong to a subregular region called subsequential. Specifically, this paper will show that the attested LDCA patterns divided into 3 major subtypes—Type 1: unbounded regressive $R \rightarrow L$, Type 2: asymmetric regressive $R \rightarrow L$, and Type 3: asymmetric progressive $L \rightarrow R$ —are all subsequential (by way of showing each type is describable by subsequential finite-state transducers).²

This paper contributes to the field of long-distance phonology in that, first of all, it provides an alternative analysis of LDCA from a computational perspective. The advantage of this analysis as compared to traditional approaches like Agreement by Correspondence (ABC) is that, as will be shown, it accounts for the attested LDCA cases at least as equally well; in addition, it is successful in eliminating pathological patterns like Majority Rules and Sour Grapes which are predicted under the ABC approach. Second, the computational approach offers a way to evaluate the complexity of phonological processes on the basis of expressivity. Being subsequential (describable by subsequential finite-state transducers) is less complex than being regular (describable by finite-state transducers), which suggests that LDCA patterns are computationally less complex than previously realized. This paper therefore identifies a stronger computational property that characterizes the nature of LDCA processes, lending support to the hypothesis that phonological processes are subregular. Together with previous work on long-distance vowel harmony (Heinz & Lai 2013), long-distance consonant dissimilation (Payne 2014), this paper contributes to a better understanding of the computational nature of long-distance phenomenon.

The paper is structured as follows. Section 2 reviews Agreement by Correspondence (ABC) approach to LDCA and shows that under the ABC approach, pathological patterns are predicted. Section 3 provides background information about the computational approach taken by this paper. Section 4 presents the computational analysis of LDCA processes and establishes that they are subsequential. The analysis shows that most unbounded LDCA processes, divided into 3 types are subsequential, and that the bounded transvocalic processes are also subsequential. Section 5 discusses implications and contributions of this work, possible directions for future research, and concludes.

2 Pathological patterns

2.1 Review of ABC

The Agreement by Correspondence (ABC) analysis suggested that participating segments of LDCA in the output must be in correspondence with each other (checked via $\text{Corr-C} \leftrightarrow \text{C}$ constraints), and that agreement is determined by Identity constraints (i.e. ID-CC (F)) that check feature matching in corresponding segments (Rose & Walker 2004). The generalized schema for these constraints, following Rose & Walker (2004), is given in (2).

² Hansson suggested that there are no instances of absolute directionality involving progressive assimilation, and that when it occurs it is a by-product of the influence of morphological constituent structure (Hansson 2010: 152). In this paper “progressive” is used only as a descriptive term, emphasizing the directionality is left-to-right as it is important to the proposed analysis.

- (2) a. Corr-C \leftrightarrow C: Let S be an output string of segments and let C_i, C_j be segments that share a specified set of features F. If C_i, C_j \in S, then C_i is in a relation with C_j; that is, C_i and C_j are correspondents of one another.
- b. ID-CC (F): Let C_i be a segment in the output and C_j be any correspondent of C_i in the output. If C_i is [F], then C_j is [F].

In order to incorporate directionality Rose & Walker (2004) modified ID-CC (F) constraints as defined in (3). Following Hansson (2007), this paper assumes a “local” evaluation of correspondence n-tuples, that is, segments stands in a correspondence relation is treat as a set of pairwise relations.³ For a 4-tuple standing in the correspondence relation [...C_{i,1}...C_{i,2}...C_{i,3}...C_{i,4}...], for instance, each ID-CC (F) constraint is evaluated on three pairs, C₁/C₂, C₂/C₃, and C₃/C₄.

- (3) a. ID-C_LC_R (F): Let C_L be a segment in the output and C_R be any correspondent of C_L such that C_R follows C_L in the sequence of segments in the output (R > L). If C_L is [F], then C_R is [F].
- b. ID-C_RC_L (F): Let C_L be a segment in the output and C_R be any correspondent of C_L such that C_R follows C_L in the sequence of segments in the output (R > L). If C_R is [F], then C_L is [F].

Tableau (4) illustrates with a hypothetical example of progressive sibilant harmony, triggered by the leftmost [-ant] sibilant.⁴ ID_{IO}(F) constraints are used to enforce faithfulness between input and output, formulated in accordance with the general IDENT(F) schema given in McCarthy & Prince (1995). Under local evaluation, candidate (4a) incurs two violations of the ID-C_LC_R[-ant] constraint because of the initial and final [f_x...s_x] pairs. The middle [s_x...f_x] pair does not incur any violation as only [-ant] triggers agreement. Candidate (4b) incurs three violations of the Corr-s \leftrightarrow f constraint as all three pairs of adjacent sibilant segments are not in correspondence relation; however, it does not violate the ID-C_LC_R[-ant] constraint as each pair of corresponding sibilant segments agrees in anteriority. Candidate (4c) and (4d) exemplify progressive and regressive harmony patterns, respectively. Candidate (4c) is the winner as the ID_{IO}[-ant] is higher ranked than ID_{IO}[+ant]. Candidate (4e) represents a “shifting pattern” that will never surface as it is harmonically bound by candidates like (4c) and (4d).

- (4) Hypothetical example of progressive sibilant harmony:

/f...s...f...s/	Corr-s \leftrightarrow f	ID-C _L C _R [-ant]	ID _{IO} [-ant]
a. f _x ...s _x ...f _x ...s _x		**!	
b. f _x ...s _y ...f _x ...s _y	**!		
☞ c. f _x ...f _x ...f _x ...f _x			
d. s _x ...s _x ...s _x ...s _x			**!
e. f _x ...f _x ...s _x ...f _x		*!	

³ Hansson argued against a “global” evaluation on the grounds that it makes bizarre typological predictions, which will be discussed in the next section.

⁴ Neutral segments, i.e. consonants and vowels that do not participate in the harmony process, are omitted from the presentation.

2.2 Majority Rules in ABC

Majority Rules (MR) is a bizarre type of unattested harmony patterns where the harmony trigger is determined solely by the majority of segments in the input with a particular feature value (Lombardi 1999; Bakovic 2000). For instance, let [+]/[-] represents [+F]/[-F] segments. If the input contains two +'s and one - (e.g. /+-+/), then the output will have three +'s [+++], but if the input contains two -'s and one + (e.g. /--+/), the output will be [---].

Hansson argued against a “global” evaluation of correspondence n-tuples on the grounds that it makes bizarre MR-type typological predictions. A simple example in (5), adapted from Hansson (2007), illustrates. Tableau (5) shows a hypothetical progressive sibilant harmony system where both [+ant] and [-ant] trigger agreement (ID-C_LC_R[+ant] and ID-C_LC_R[-ant] are collapsed into ID-C_LC_R[±ant] for convenience, S represents potential sibilant targets). The global evaluation will create a tie between candidates (5b) and (5c), and which one surfaces as the winner will depend on the lower ranked faithfulness constraints (i.e. ID_{IO}[+ant] and ID_{IO}[-ant]). This creates a MR-type effect as whichever [±ant] value matches the majority of the three potential targets will become the harmonizing feature. To see this more clearly, Tableau (6) shows that if two of the three unspecified S's were /s/'s and one were /ʃ/, among the four candidates that are being considered,⁵ the winning candidate will be (6c) where the last three sibilants agree with [s] based on the ranking of faithfulness constraints ID_{IO}[-ant] >> ID_{IO}[+ant]. Tableau (7) shows, on the other hand, that under the same constraint ranking, if two of the three unspecified S's were /ʃ/'s and one were /s/, the winning candidate will be (7d) where the last three sibilants agree with [ʃ].

(5) MR-type effects under a “global” evaluation:

/ʃ...z...s...s...s/	*ʒ	Corr-s↔ʃ	ID-C _L C _R [±ant]
a. $\int_x \dots z_y \dots \int_x \dots \int_x \dots \int_x$		****!	
b. $\int_x \dots z_x \dots \int_x \dots \int_x \dots \int_x$			****
c. $\int_x \dots z_x \dots s_x \dots s_x \dots s_x$			****
d. $\int_x \dots \int_x \dots \int_x \dots \int_x \dots \int_x$	*!		

(6) More underlying /s/'s:

/ʃ...z...s...s...s/ʃ/	*ʒ	Corr-s↔ʃ	ID-C _L C _R [±ant]	ID _{IO} [ant]
a. $\int_x \dots z_y \dots s_x \dots s_x \dots \int_x$		****!		
b. $\int_x \dots z_x \dots s_x \dots s_x \dots \int_x$			*****!	
c. $\int_x \dots z_x \dots s_x \dots s_x \dots s_x$			****	*
d. $\int_x \dots z_x \dots \int_x \dots \int_x \dots \int_x$			****	*!
e. $\int_x \dots \int_x \dots \int_x \dots \int_x \dots \int_x$	*!			**

⁵ Candidates like $s_x \dots z_x \dots s_x \dots s_x \dots s_x$ are not considered in Tableaux (5) – (7), as this example is intended to show a progressive harmony process. It is assumed that there are higher ranked constraints that enforce the direction of the harmony process.

(7) More underlying /ʃ/'s:

/ʃ...z...ʃ...ʃ...s/	*ʒ	Corr-s↔ʃ	ID-C _L C _R [± ant]	ID _{IO} [ant]
a. ʃ _x ...z _y ...ʃ _x ...ʃ _x ...s _x		****!		
b. ʃ _x ...z _x ...ʃ _x ...ʃ _x ...s _x			*****!	
c. ʃ _x ...z _x ...s _x ...s _x ...s _x			****	**!
d. ʃ _x ...z _x ...ʃ _x ...ʃ _x ...ʃ _x			****	*
e. ʃ _x ...ʒ _x ...ʃ _x ...ʃ _x ...ʃ _x	*!			*

Under a “local” evaluation, as Hansson pointed out, the tie between candidate (5b) and (5c) disappears. As shown in (8), (5b) incurs two violations of ID-C_LC_R[± ant] constraint against one for candidate (5c).⁶ However, this solution is not entirely satisfactory as it still predicts another type of well-known pathological patterns called Sour Grapes (Wilson 2003a;b; Finley 2008).

(8) A “local” evaluation breaks the tie:

/ʃ...z...s...s...s/	*ʒ	Corr-s↔ʃ	ID-C _L C _R [± ant]
a. ʃ _x ...z _y ...ʃ _x ...ʃ _x ...ʃ _x		****!	
b. ʃ _x ...z _x ...ʃ _x ...ʃ _x ...ʃ _x			**
c. ʃ _x ...z _x ...s _x ...s _x ...s _x			*
d. ʃ _x ...ʒ _x ...ʃ _x ...ʃ _x ...ʃ _x	*!		

2.3 Sour Grapes in ABC

Sour Grapes (SG) refers to a type of logically possible, but unattested vowel harmony pattern. Padgett (1995) defined Sour Grapes harmony as “Either all features must spread, or none will...” That is, SG patterns require all vowels agree with respect to some feature unless there is an opaque vowel present. Following Heinz & Lai (2013), such mappings can be schematically represented as in (9), where [+]/[-] represents [+F]/[-F] vowels, and [⊕]/[⊖] represents [+F]/[-F] opaque vowels. Consonants are not relevant to the process and are omitted from the representation.

(9) SG mappings:

UR	SF
+ - - - -	→ + + + + +
+ - - - ⊖	→ + - - - ⊖

Tableau (10) and (11) show that SG patterns are predicted under the ABC approach with local evaluation. Tableau (10) shows that when an opaque vowel (i.e. ⊖) is present, [+F]

⁶ This is not to say that local evaluation solves the MR problem, but that it is advantageous to global evaluation because it has a better chance to avoid the MR-type effects without relying on the lower ranked faithfulness constraints.

feature does not spread and the winner is the faithful candidate. Tableau (11) shows, on the other hand, under the same constraint ranking, [+F] feature spread throughout the word when there is no opaque element present.

(10) SG patterns, when opaque element present:

/+ - - - □/	*□	ID _{IO} [+]	Corr- + ↔ -	ID-C _L C _R [+]	ID _{IO} [-]
[☞] + _x - _x - _x - _x □ _x				*	
+ _y - _x - _x - _x □ _x			*!		
+ _x + _x - _x - _x □ _x				*	*!
+ _x + _x + _x - _x □ _x				*	**!
+ _x + _x + _x + _x □ _x				*	***!
+ _x + _x + _x + _x □ _y			*!		***
+ _x + _x + _x + _x □ _x	*!				****
- _x - _x - _x - _x □ _x		*!			

(11) SG patterns, when no opaque element present:

/+ - - - -/	*□	ID _{IO} [+]	Corr- + ↔ -	ID-C _L C _R [+]	ID _{IO} [-]
+ _x - _x - _x - _x - _x				*!	
+ _y - _x - _x - _x - _x			*!		
+ _x + _x - _x - _x - _x				*!	*
+ _x + _x + _x - _x - _x				*!	**
+ _x + _x + _x + _x - _x				*!	***
[☞] + _x + _x + _x + _x + _x					****
- _x - _x - _x - _x - _x		*!			*

As Majority Rules and Sour Grapes are referred to as pathological, a desirable phonological theory should be able to eliminate these unattested patterns, while adequately account for the attested ones. The ABC approach is successful to the extent that MR-type effects can be avoided under the local evaluation of constraints, however, it still makes pathological predictions about the SG patterns. In the following sections this paper proposes an alternative approach to LDCA from a computational perspective, and suggests that it is desirable to characterize LDCA by the computational class called “subsequential”. Section 3 introduces some basic concepts of the computational approach and how it can be useful in modeling phonological processes like LDCA. Section 4 presents the computational analysis and shows that it adequately accounts for the attested LDCA processes. Section 5.1 discusses how subsequentiality is effective in eliminating MR and SG patterns.

3 The computational approach

3.1 Pattern complexity

The computational approach taken by this paper employs the notion of the Chomsky Hierarchy, which allows for comparisons of the relative complexity of patterns within and across linguistic domains. The Chomsky Hierarchy classifies patterns according to their computational complexity (Chomsky 1956; Partee et al. 2012), that is, the expressive power of the grammar needed to generate such patterns. Each region is formally more expressive (i.e. more complex) than the region nested within it. For instance, regular patterns are less complex than context-free patterns, which is less complex than context-sensitive patterns. Previous work in computational analysis of natural language patterns have shown that syntactic patterns require grammars higher up in the Chomsky Hierarchy. English Nested Embedding, for instance, is context-free (Chomsky 1956), and Swiss German Crossing dependencies are context-sensitive (Schieber 1985).

- (12) The Chomsky Hierarchy:
 finite \subset regular \subset context-free \subset context-sensitive \subset recursively enumerable

3.2 LDCA mappings are subsequential

This paper focuses on the computational nature of phonological processes, which can be thought of as string-to-string mappings, that is, the mapping from underlying forms (/UR/ as strings) to surface forms ([SF] as strings). It has been shown that phonological processes are regular relations (Johnson 1972; Kaplan & Kay 1994). The regular region shown in (12) refers to sets of strings. Regular relations, in contrast, refer to sets of string pairs (e.g. the set of (/UR/, [SR]) pairs when considering phonological processes).

Recent work have suggested that phonological processes may be subregular, i.e. less complex than regular (Heinz 2010; Heinz & Lai 2013). It is desirable to find a narrower bound as it is known that many regular relations generate patterns that are unattested in natural language phonology. The class of subsequential functions are nested within the regular region, therefore are less complex than regular. This paper will show that LDCA mappings are subregular, specifically, they are subsequential. Chandlee defined three subclasses of subsequential functions—Input Strictly Local (ISL) and Output Strictly Local (OSL, Left-OSL and Right-OSL)—and showed that ISL and OSL functions are sufficient to model a range of local phonological processes such as final voice devoicing, insertion, deletion etc. (Chandlee 2014; Chandlee et al. 2014; 2015). ISL and OSL functions crosscut the subsequential region, as shown in Figure 1, which suggests that local phonological processes are even less complex than subsequential.⁷ However, as Chandlee discussed, these functions are not sufficient to model long-distance processes.

3.3 Subsequential finite-state transducers

The class of subsequential functions are describable with subsequential finite-state transducers (hereafter SFSTs). Figure 2 shows a simple example of a SFST, which consists of the following elements:⁸

⁷ Subsequential functions will be defined as either left-subsequential or right-subsequential in the next section.

⁸ These are to introduce the basic notions to readers who are not familiar with the formalism, but not intended to be definitions. For formal definitions of SFSTs, readers are referred to Oncina et al. (1993).

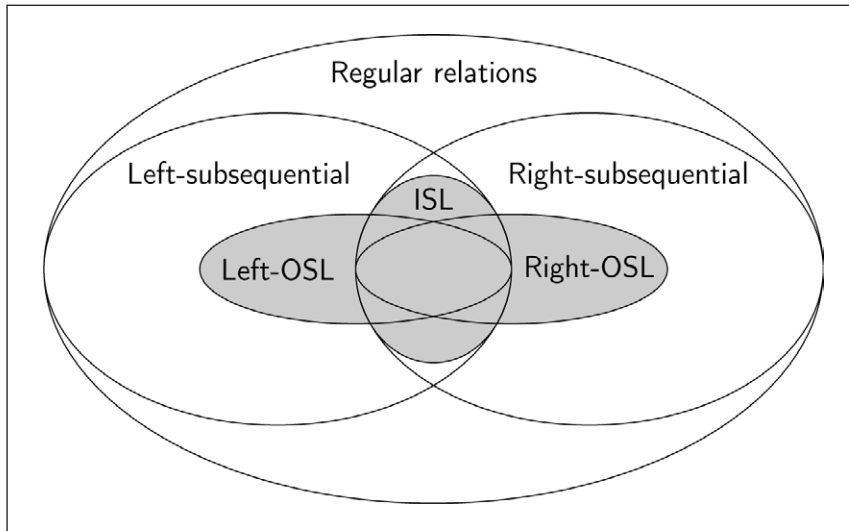


Figure 1: The Subregular Hierarchy for string-to-string mappings.

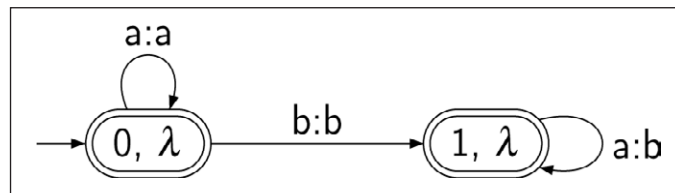


Figure 2: An example of a SFST.

- states: SFSTs consist of a set of states, represented with circles.⁹
- initial state: start state, with an arrow pointing to it.
- final states: marked with double circles, also called accepting states.
- transitions: represented with labeled arrows that connect the states.
- loops: transitions that start and end in the same state.
- alphabet: a finite set of symbols.¹⁰ A string or word, w , is a finite sequence of symbols from the alphabet. λ represents the empty string.
- labels on transitions: $a:b$ in Figure 2, for instance, means output b when input is a .

SFSTs are deterministic, that is, at any state of such transducers, no two outgoing transitions are labeled with the same input symbol, and they allow an additional output string at final states.¹¹ Since subsequential transducers are deterministic, the relations they recognize are functions. With these restrictions, mappings describable by SFSTs (i.e. subsequential functions) are a proper subset of the regular relations (Mohri 1997).¹²

Starting from the initial state, the SFST reads a given input string one symbol at a time and takes the transition that matches the current input symbol. A left-subsequential function is obtained by applying a SFST left-to-right over the input string. Consider the SFST

⁹ States represented with single circles are not final/accepting states, i.e. strings that end in such states are not accepted. All states are final/accepting in SFSTs.

¹⁰ When considering phonological processes, for instance, the alphabet can be the set of IPA symbols for sounds that occur in a given language. Input alphabet and output alphabet can be different sets of symbols.

¹¹ The additional output strings are indicated by the labels inside each state. For instance, $1, \lambda$ inside the double circled final state in Figure 2 means if the process ends at state 1, output an additional λ , which is equal to saying no additional output.

¹² Regular relations are describable with finite-state transducers (FSTs), for more on regular relations and formal definitions of FSTs, see Hopcroft & Ullman (1979); Oncina et al. (1993).

in Figure 2 and the mapping from the input string *abaa* to the output string *abbb*. The SFST reads the leftmost symbol *a* first, takes the loop labeled *a:a* in state 0, stays in state 0 and outputs *a*; the SFST then reads the second symbol *b* and takes the outgoing transition labeled *b:b* to state 1, outputs *b*; next the SFST reads the third symbol *a* and takes the loop labeled *a:b* in state 1, stays in state 1 and outputs *b*; finally the SFST reads the last symbol *a* and takes the loop labeled *a:b* again, stays in state 1 and outputs another *b*; the process ends at state 1, the additional output is λ , and the final output string is *abbb*. A right-subsequential function, similarly, is obtained by applying a SFST right-to-left over the input string. Equivalently, this can be thought of as applying the left-subsequential function over the reversed input string ($T(w^r)$), and then reverse the output string ($[T(w^r)]^r$). Applying a right-subsequential function to the input string *abaa* using the SFST in Figure 2, for instance, would mean to process the input string from right-to-left, i.e. *aaba*. which is equivalent to applying the SFST to the reversed input string, $w^r = aaba$, the output is *aabb*, and then reverse the output. The final output $[T(w^r)]^r$ is *bbaa*.

SFSTs can be thought of as accepting pairs of strings. For instance, the pair (w_1, w_2) is accepted by a SFST if the SFST produces the output w_2 for the input w_1 . The string-to-string mappings accepted by the SFST in Figure 2 thus include (aa, aa) , (ab, ab) , (ba, bb) , (aba, abb) , $(abaa, abbb)$ etc. A mapping like (aba, aba) , for instance, is not accepted as at state 1 after reading the second symbol *b*, there is no outgoing transition labeled with input symbol *a*. To see how SFSTs can be helpful in modeling phonological processes, consider the fact that the mappings accepted by the SFST in Figure 2 have in common that if there is a *b* in the input string, all subsequent symbols can only be *b*'s, which can be expressed as “after seeing *b*'s all subsequent *a*'s become *b*, otherwise the mapping is not accepted”. This is similar in spirit to a description of an asymmetric sibilant harmony process where the leftmost $[-ant]$ sibilant sound triggers the harmony process, all subsequent sibilant sounds become $[-ant]$. It will be discussed in more detail in the following sections how attested LDCA processes can be modeled by SFSTs.

4 Computational analysis

This section presents a computational analysis of the attested LDCA processes. Based on a survey of approximately 170 distinct cases of LDCA, Hansson suggested that the directionality of LDCA is anticipatory (regressive, right-to-left) by default, and that perseveratory (progressive, left-to-right) harmony is also attested but they can almost always be explained as a by-product of stem control where the direction of agreement is governed by morphological constituent structure, or occasionally as a dominant-recessive harmony system where only one feature value (the “dominant” one) triggers agreement (Hansson 2010). The attested LDCA patterns are either unbounded (in the sense that there are more than one intervening segments between the trigger and the targets) or transvocalic (the trigger and the target cannot be separated by more than one vowel). The main concern of this paper is the unbounded LDCA patterns, and it will be shown that such patterns can be divided into three major types—Type 1: unbounded regressive $R \rightarrow L$; Type 2: asymmetric regressive $R \rightarrow L$; and Type 3: asymmetric progressive $L \rightarrow R$. All three types will be shown to be subsequential. Section 4.4 discusses the bounded transvocalic processes, like other bounded processes discussed in Chandlee & Heinz (2012), they are subsequential. Section 4.5 discusses treatments of special cases that do not fall into the major types and shows that they do not contradict the main claims of this paper.

4.1 Type 1: Unbounded regressive $R \rightarrow L$

The sibilant harmony process in Navajo provides an example of a Type 1 process. In Navajo, sibilant sounds agree in anteriority with the rightmost sibilant sound in the word, irrespective of morpheme structure, as shown in (13). In the word for ‘I’m rolling

along.’ for instance, the rightmost sibilant sound [s] is [+ant], and the preceding [ʃ] assimilates to [s] in anteriority to become [s].

- (13) Navajo regressive harmony (McDonough 1991)
- | | | | |
|-----------------|---|-----------|-------------------------------------|
| /j-i-ʃmas/ | → | jismas | ‘I’m rolling along.’ |
| /ʃ-is-ná/ | → | sisná | ‘he carried me.’ |
| /si-dʒé:ʔ/ | → | ʃidʒé:ʔ | ‘they lie (slender stiff objects).’ |
| /dz-iʃ-l-ta:l/ | → | dʒiʃta:l | ‘I kick him [below the belt].’ |
| /dz-iʃ-l-ts’in/ | → | dzists’in | ‘I hit him [below the belt].’ |

This process is unbounded in the sense that the harmony process is not sensitive to the amount of intervening material. The rightmost sibilant sound decides the feature value for anteriority for all the other sibilant sounds in the word, whether the target (s) is in prefix, root, or suffix is irrelevant to the process. It is regressive in the sense that the trigger is to the right of the target (s). R → L is used to specify that Type 1 processes proceed from the right edge of the words to the left. The mappings are right-subsequential, as summarized in (14).

(14) For all w, URH (w) = [T (w^r)]^r

Figure 3 shows a SFST that models Type 1 processes. All Type 1 processes can be modeled by the SFST by substituting the labels on the transitions to relevant segments/natural classes involved in a given language. A list of languages involving Type 1 processes and relevant substitutions can be found in Appendix A.

To illustrate, consider the mapping from /j-iʃ-mas/ ‘I’m rolling along.’ to its surface form [jismas]. The input string is w = jifmas in this case. According to (14), the first step is to reverse the order of the string: w^r = (jifmas)^r = samʃij. The next step is to treat w^r = samʃij as input string and process one symbol at a time. The output string obtained is samsij. (15) shows the derivation. The final step is to reverse the output string to obtain the correct surface form: [T (w^r)]^r = (samsij)^r = jismas.

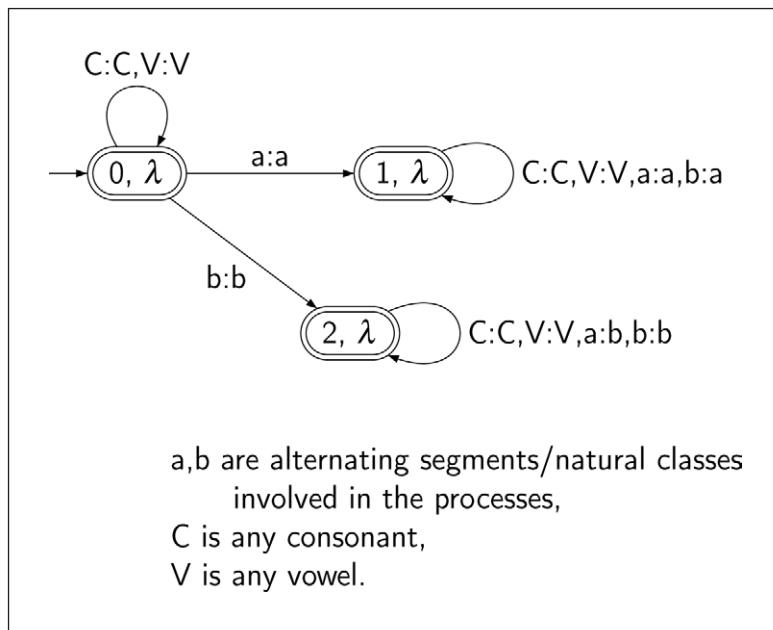


Figure 3: SFST for Type 1 processes.

- (15) Derivation for the mapping /samfij/ → [samsij]:
- | | | | | | | | |
|---------|---|-----|-----|-----|-----|-----|-----|
| Input: | s | a | m | f | i | j | |
| State: | 0 | → 1 | → 1 | → 2 | → 2 | → 2 | → 2 |
| Output: | s | a | m | s | i | j | |

4.2 Type 2: Asymmetric regressive R → L

Type 2 processes can be considered as a subtype of Type 1 processes. The difference is that it is asymmetric in the sense that only one feature value ([+F] or [-F]) triggers the harmony process. In Kera, for instance, the feminine prefix /t-/ is realized as /tʃ/ if another /tʃ/ follows in the stem. The process is regressive as the trigger is to the right of the target (s), and it is asymmetric assimilation between /t/ and /tʃ/ as */t...tʃ/ sequences are disallowed, but /tʃ...t/ sequences are permissible.

- (16) Coronal harmony in feminine prefix /t-/ (Ebert 1979)
- | | | | |
|-----------|---|----------|---------------|
| /t-Ó:já/ | → | t-Ó:já | ‘dog’ (FEM) |
| /t-e:ŋa/ | → | t-e:ŋa | ‘dry’ (FEM) |
| /t-ə:tʃá/ | → | tʃ-ə:tʃá | ‘small’ (FEM) |

Similar to Type 1 processes, Type 2 processes are also right-subsequential (i.e. the reverse steps summarized in (14) apply). Figure 4 shows a SFST that models the harmony process in Kera, where $x = t:t$, $y = tʃ:tʃ$, $z = t:tʃ$, $C =$ any other consonant, $V =$ any other vowel. Type 2 processes in general can be modeled by a two-state SFST similar to the one shown in Figure 4, by substituting the labels on transitions with the relevant alternating segments/natural classes of segments involved in the given process.

To illustrate, consider the mapping from /tə:tʃá/ ‘small (fem)’ to its surface form [tʃə:tʃá]. The input string is $w = tətʃá$. The first step is to reverse the order of the string: $w^r = (tətʃá)^r = áʃtət$. The next step is to treat $w^r = áʃtət$ as input. The output obtained is $áʃtətʃ$. (17) shows the derivation. The final step is to reverse the output string to obtain the correct surface form: $[T(w^r)]^r = (áʃtətʃ)^r = tʃə:tʃá$.

- (17) Derivation for the mapping /áʃtət/ → [áʃtətʃ]:
- | | | | | | |
|---------|---|-----|-----|-----|-----|
| Input: | á | tʃ | ə: | t | |
| State: | 0 | → 1 | → 1 | → 1 | → 1 |
| Output: | á | tʃ | ə: | tʃ | |

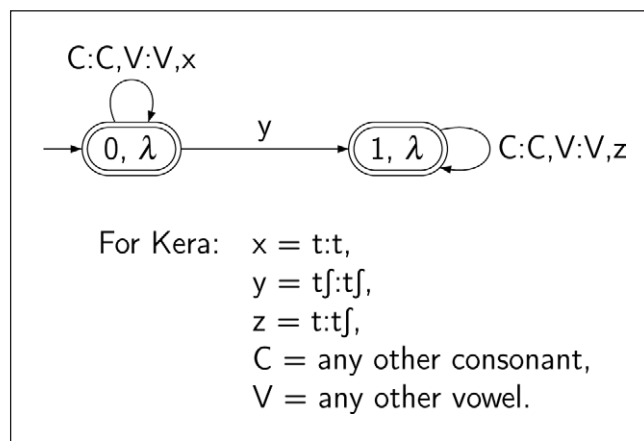


Figure 4: SFST for Type 2 processes.

4.3 Type 3: Asymmetric progressive L → R

The nasal consonant harmony in Yaka provides an example of a Type 3 process. Similar to Type 2 processes, Type 3 processes are asymmetric as only one of feature values, [+F] or [-F], triggers the harmony process. The directionality of agreement is from left to right for Type 3 processes, as specified with L → R. In Yaka, a suffixal /l,d/ will surface as [n] if preceded by a nasal earlier in the stem, as shown in (18). The process is asymmetric as only the [+nasal] feature triggers the l/n and d/n alternation, i.e. *[+nasal]...l,d sequences are disallowed while [-nasal]...l,d sequences are permissible.

- (18) Yaka nasal consonant harmony (Hyman 1995)
- a. Harmony in perfective suffix /-idi/:
 - /-ján-idi/ → -ján-ini ‘cry out in pain’
 - /-tsúm-idi/ → -tsúm-ini ‘sew’
 - /-jád-idi/ → -jád-idi ‘spread’
 - /-tsúb-idi/ → -tsúb-idi ‘wander’
 - b. Harmony in perfective suffix /-ele/:
 - /-són-ele/ → -són-ene ‘color’
 - /-kém-ele/ → -kém-ene ‘moan’
 - /-sól-ele/ → -sól-ele ‘deforest’
 - /-kéb-ele/ → -kéb-ele ‘be careful’

Type 3 processes are left-subsequential, and can be modeled by the two-state SFST shown in Figure 5 by substituting the transitions labels x, y, and z with the relevant alternating segments/natural classes of segments involved in the given process. For the nasal consonant harmony in Yaka, let x = d:d, l:l, y = [+nasal]:[+nasal], z = d:n, l:n, C = any other consonant, V = any other vowel. (19) illustrates how the SFST maps the underlying /-ján-idi/ ‘cry out in pain’ to its surface form [-ján-ini].

- (19) Derivation for the mapping /-ján-idi/ → [-ján-ini]:
- | | | | | | | | |
|---------|---|-----|-----|-----|-----|-----|-----|
| Input: | j | á | n | i | d | i | |
| State: | 0 | → 0 | → 0 | → 1 | → 1 | → 1 | → 1 |
| Output: | j | á | n | i | n | i | |

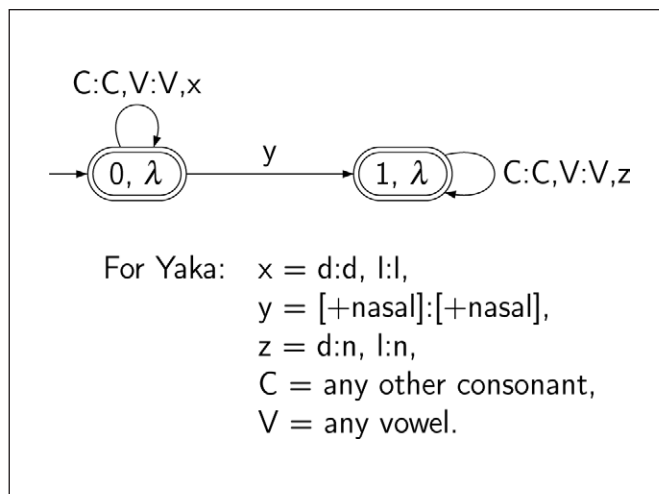


Figure 5: SFST for Type 3 processes.

It is natural to wonder if there are any cases of symmetric unbounded progressive L → R process analogous to Type 1 processes, where both [+F] and [-F] features trigger the harmony process. Evidence suggests that there is not. Some unbounded cases of progressive harmony are potentially ambiguous between symmetric and asymmetric, but there are not enough data to distinguish. In Aari, for instance, sibilant sounds agree in anteriority with the leftmost sibilant sound in the word, as shown in (20).

- (20) Aari progressive harmony (Hayward 1988; 1990)
- a. Harmony in causative /-sis/:
 - /nəf-sis/ → nəf-fiʃ ‘cause to love’
 - /fa:n-sis/ → fa:n-fiʃ ‘cause to urinate’
 - b. Harmony in perfective /-s/:
 - /ʔuf-s-it/ → ʔuf-f-it ‘I cooked.’
 - /tʃʔa:q-s-it/ → tʃʔa:q-f-it ‘I swore.’

This process is clearly unbounded in the sense that the harmony process is not sensitive to the amount of intervening material. But as Hansson pointed out, all alternating suffixes have alveolar /s/, and thus the alternation is only seen from /s/ → [f] (Hansson 2010). Examinations of the source material in Benchnon (Rapold 2006), Mayak (Andersen 1999), and Dime (Seyoum 2008) likewise revealed no evidence of any symmetry in progressive harmony processes. The absence of symmetric progressive LDCA processes is interesting and worth trying to obtain a better understanding of, but does not detract from the main claims of this paper. This paper suggests that the attested LDCA processes are all subsequential. However, subsequentiality could be a necessary condition, but not a sufficient one, of phonological processes. An implication is that a tighter bound can be found on the range of phonological processes, which will be discussed in more detail in Section 5.2.

4.4 Transvocalic processes

A few cases discussed in Hansson (2010) are strictly transvocalic, that is, the harmony only applies when the trigger is separated from the target by no more than a vowel. The nasal consonant harmony in Lamba provides an example, as shown in (21).¹³ The transvocalic processes count as LDCA according to Hansson’s definition, but they are different than the unbounded processes discussed in the previous sections, they are “bounded” in the sense that there is a bound on the number of intervening segment permissible.

- (21) Transvocalic nasal consonant harmony in Lamba (Odden 1994)
- a. Transitive reversive suffix [-ulul-]:
 - fis-ulul-a ‘reveal’
 - min-unun-a ‘unswallow’
 - mas-ulul-a ‘unplaster’ (*[mas-unun-a])
 - b. Intransitive reversive suffix [-uluk-]/[-olok-]:
 - fis-uluk-a ‘get revealed’
 - min-unuk-a ‘get unswallowed’
 - mas-uluk-a ‘get unplastered’ (*[mas-unuk-a])

The transvocalic processes are also subsequential. Figure 6 shows a SFST that models the transvocalic harmony process in Lamba, where l = [l], n = [n], N = [+nasal], C = any other consonant, V = any vowel. In fact, the “boundedness” of transvocalic

¹³ For similar transvocalic harmony processes, see Hansson (2010: 87–96).

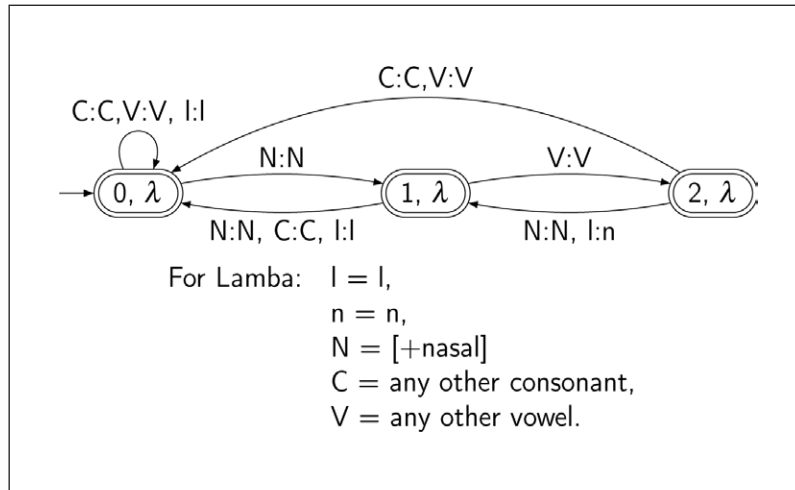


Figure 6: SFST for the transvocalic process in Lamba.

processes makes them describable by the class of Input Strictly Local functions as defined in Chandlee et al. (2014), which is more restricted than the class of subsequential functions.

4.5 Special cases

There are a few cases that do not easily fall into one of the three major types discussed in the previous sections, this section discusses treatments of such cases and suggests that they do not contradict the main claims of this paper.

4.5.1 Gooniyandi

Gooniyandi, a Northern Australia language, has a process where coronal segments interact with each other.¹⁴ In Gooniyandi, word-initial apicals harmonize with a following apical, as shown in (22). Note that the harmony process does not apply to the words for ‘grass’ and ‘I bring them.’, as [l] and [d] are not word-initial.

- (22) Coronal harmony in Gooniyandi apicals (McGregor 1990)
- | | | |
|-------------|-----------------|-----------------------------|
| tili | ‘light’ | (*t̪ili) |
| t̪iɾippindi | ‘he entered.’ | (~ t̪iɾippindi only rarely) |
| kiliŋi | ‘grass’ | (*kiŋi) |
| wadgʉluna | ‘I bring them.’ | (*wadgʉluna) |

It is a regressive harmony process as the trigger is to the right of the target. It is different than Type 1 processes in that the process targets only the word-initial apicals, the non-word-initial apicals do not alter when they are followed by another apical. The SFST in Figure 3 for unbounded regressive harmony does not capture this process as it cannot check whether the target is word-initial. However, this process is still (right-)subsequential. Figure 7 shows a SFST for the Gooniyandi process, where a = [+ant, +cor, +dist], b = [+ant, +cor, -dist], X = any other segment (consonants and vowels).¹⁵

The derivation for the word /kiliŋi/ ‘grass’ is shown in (23), the surface form obtained is [T (w^r)]^r = kiliŋi. Consider, for instance, a hypothetical underlying form /liŋi/ with the word-initial apical, the harmony process would apply. The SFST would take as input /iŋi/ and output iŋi, whose reverse string is ŋi. The derivation is shown in (24).

¹⁴ Similar process can be found in a related language, Gaagudju (Gafos 1996).

¹⁵ To clarify, this SFST has more states and transitions in order to ensure that only word-initial apical is targeted, but these do not increase the computational complexity.

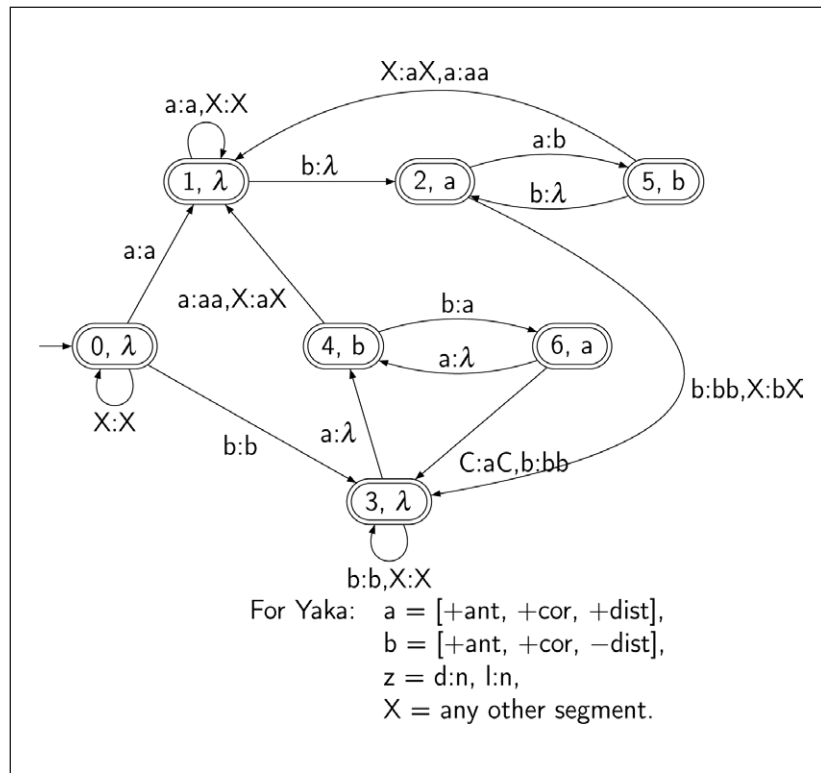


Figure 7: SFST for Gooniyandi coronal harmony.

(23) Derivation for the mapping /iŋilik/ → [iŋilik]:

Input:	i	ŋ	i	l	i	k	
State:	0	→ 0	→ 3	→ 3	→ 4	→ 1	→ 1
Output:	i		ŋ	i	λ	li	k

(24) Derivation for the mapping /iŋil/ → [iŋi]:

Input:	i	ŋ	i	l		
State:	0	→ 0	→ 3	→ 3	→ 4	→
Output:	i	ŋ	i	λ	l	

4.5.2 Gwendolyn's labiodental harmony

Stemberger (1988; 1993) discussed a labiodental harmony process in the speech of Gwendolyn, a child of around age four.¹⁶ It was reported that as Gwendolyn was learning English, she would produce bilabial /m/ as [m̥] when it is preceded or followed by a labiodental [f] or [v] in her speech. The process is special as it is bidirectional in the sense that the harmonizing feature spreads both leftwards and rightwards, as is shown by the last example in (25).

(25) Labiodental harmony in the speech of Gwendolyn (Stemberger 1988; 1993)

[lʌv m̥ais]	'love mice'
[snɪf m̥ais]	'sniff mice'
[fɪŋeɪ m̥ais]	'smell mice'
[fɪŋeɪ ɪm]	'smell him'
[m̥aj fɪŋeɪ m̥ais]	'my smelly mice'

¹⁶ Hansson (2010: 77–78) noted that Gwendolyn's labial consonant harmony is a rather unusual case, and given the data available it is hard to decide the exact nature of the process and its relation to the kinds of harmony processes observed in adult language.

Heinz & Lai (2013) suggested that bidirectional processes can be analyzed as a composition of left-subsequential and right-subsequential functions. Consider, for instance, the mapping: /mai fm̥ui mais/ → [m̥ai fm̥ui m̥ais]. This process can be modeled with a Type 3 SFST where $x = m:m$, $y = f:f$, $v:v$, $z = m:m̥$. A composition of left-subsequential and right-subsequential functions would mean to first process the input string mai fm̥ui mais left-to-right, the output obtained is mai fm̥ui m̥ais. Then, treat the output string mai fm̥ui m̥ais as input to a right-subsequential function, i.e. process the string right-to-left. The final output obtained is the correct surface form [m̥ai fm̥ui m̥ais]. It should be noted that bidirectional processes are very rare, in fact, it is suggested that there are no genuine attested cases of bidirectional “dominant-recessive” consonant harmony system (Hansson 2010).

4.5.3 Sanskrit n-retroflexion

Sanskrit n-retroflexion is a process where a continuant retroflex consonant (i.e. /ʃ/ or /r/) causes a following dental nasal /n/ to become retroflex /ɳ/, as shown in (26a). This first appears to be a progressive LDCA process, but it has a number of unusual properties, data in (26b) show two of them: first, intervening coronals block this progressive retroflexion assimilation; and second, the process targets only the first nasal segment in a sequence of nasals. The example in (26c) makes it more peculiar as it shows that the process fails if the trigger also occurs to the right of the target, similar to the pathological Sour Grapes patterns. which would make the process not subsequential (Heinz & Lai 2013).

(26) Sanskrit n-retroflexion (Schein & Steriade 1986; MacDonell 1910)

- a. n-retroflexion:

/iʃ-na:-/	→	iʃ-ɳa:-	‘seek’ (present stem)
/cakʃ-a:na-/	→	cakʃ-a:ɳa-	‘see’ (middle participle)
/brahman-i/	→	brahmanɳ-i	‘brahman’ (LOCSG)
- b. Opaque coronals:

marj-a:na-	*	(marj-a:ɳa-)	‘wipe (middle participle)’
kʃveda:na-	*	(kʃveda:ɳa-)	‘hum (middle participle)’
- c. Process fails if trigger is also to the right:

pari-naḁṣati	*	(pari-ɳaḁṣati)	‘encompasses’
--------------	---	----------------	---------------

It should be noted that Hansson (2010) argued extensively that it is not appropriate to treat Sanskrit n-retroflexion as an instance of LDCA because it exhibits segmental opacity, among other reasons. This paper follows Hansson’s arguments and assume that Sanskrit n-retroflexion is not an instance of LDCA.

To summarize, Section 4 establishes that attested LDCA processes are subsequential. Specifically, the unbounded LDCA processes, divided into three types, are shown to be subsequential; next, the bounded cases of LDCA are also shown to be subsequential; and lastly, treatments of special cases of LDCA are discussed.

5 Discussion and conclusion

5.1 Pathological patterns revisited

It has been known that Majority Rules patterns are easily generated in OT-type framework with unbounded constraints (Bakovic 2000; Lombardi 1999), resulting from the comparative nature of OT (Frank & Satta 1998). Not much work has eliminated MR problem entirely without generating other problems (Walker 1997; Karttunen 1998; Gerdemann & van Noord 2000; Lombardi 2001). This section discusses how Majority

Rules and Sour Grapes can or cannot be handled in three OT variants and suggests that a computational analysis, especially characterizing LDCA with subsequenceality as proposed by this paper, is promising in delimiting the range of possible phonological processes. Two variants of OT that are more successful in eliminating MR patterns will first be discussed, both employ finite-state constraints, similar in line with the current proposal, then a derivational version of OT that has received much attention—Harmonic Serialism—will be discussed.

One of the proposed variant of OT that completely eliminates the MR problem is Eisner's directional OT (Eisner 2000). Eisner proposed the finite-state directional constraints where violations of the constraints closer to the specified edge of the form are strictly worse than violations further from that edge. Under left-to-right evaluation, for instance, the candidates that incur violations of the constraints toward the right edge of the form are preferred. (27) and (28) show the difference between left-to-right and right-to-left evaluation when applied to the same hypothetical sibilant harmony process considered in Section 2.2 (cf. Tableau (7)). Loc1 indicates the leftmost location where AGREE-type constraints checking whether sibilants agree in anteriority need to be evaluated, and one violation is assigned if the two sibilants disagree in anteriority at this location. Similarly, Loc2 indicates the second location where the constraints need to be evaluated, Loc3 indicates the third location, and Loc4 indicates the rightmost location. In a sense directional evaluation is similar to the ABC approach with local evaluation in that violations are evaluated locally between adjacent sibilant pairs, but it is different as violations are checked iteratively at different locations. Overall, directional evaluation entails sensitivity to the location of violations as opposed to total number of violations, thus eliminating the possibility to compare unbounded counts as in the original OT framework, which gives rise to MR patterns.

(27) Directional evaluation left-to-right:

/f...z...f...f...s/	* \mathfrak{z}	Loc1	Loc2	Loc3	Loc4
a. f...z...f...f...s		*	*!		*
b. f...z...f...f...f		*	*!		
\mathfrak{z} c. f...z...s...s...s		*			
d. f... \mathfrak{z} ...f...f...f	*!				

(28) Directional evaluation right-to-left:

/f...z...f...f...s/	* \mathfrak{z}	Loc4	Loc3	Loc2	Loc1
a. f...z...f...f...s		*!		*	*
b. f...z...f...f...f				*!	*
\mathfrak{z} c. f...z...s...s...s					*
d. f... \mathfrak{z} ...f...f...f	*!				

Another OT variant that employs the finite-state constraints is Riggle's dissertation work (Riggle 2004). Riggle proposed the Optimality Transducer Construction algorithm (OTCA), which takes a ranked set of constraints and constructs a finite-state transducer (FST) incrementally that generates optimal output forms from underlying forms (Riggle 2004). In

this work, Riggle adopted a finite-state assumption for the constraints—all constraints are expressed as finite-state machines.¹⁷ All constraints of a given grammar are combined into a single finite-state machine that acts as the evaluation function (by taking intersections of individual finite-state machines). OTCA provides a procedure for FST construction, and Riggle showed that the procedure fails with a grammar that generates MR patterns as the algorithm will be unable to introduce a variable in constructing the FST, thus eliminating the possibility for generating any MR patterns. The successfulness of OTCA in eliminating MR is essentially due to the fact that MR is not regular, that is, the mappings cannot be described by FSTs. For a formal proof of MR is not regular, readers are referred to Heinz & Lai (2013).

However, although the regular boundary (finite-state constraints) can eliminate MR-type mappings, it can still generate SG-type mappings as they are regular (Heinz & Lai 2013). Restricting phonological processes to subsequential functions (i.e. describable by SFSTs), on the other hand, eliminates MR and SG entirely.

As discussed in Section 3.2, subsequential functions are a subset of regular relations, since MR is not regular, it follows that MR is not subsequential. Heinz & Lai (2013) presented a formal proof establishing that SG is not subsequential. Following Heinz & Lai, SG mappings are formally defined as follows: SG is a length-preserving function which at a minimum includes the following mappings for all $n \in \mathbb{N}$: $SG(+ -^n) = + +^n \wedge SG\Box(+ -^n\Box) = + -^n\Box$.¹⁸ Consider $n = 2$, for instance, SG will map underlying $+ -^2 = +- -$ to $+++$ with total harmony, and underlying $+ -^2\Box = +- - \Box$ to $+ - -\Box$. This paper explains the intuition why SG mappings are not subsequential, readers are referred to the original paper for formal proof and technical details. One way to show that SG mappings are not subsequential is to show that it would require infinitely many states to model such functions. This is because all subsequential functions can be modeled by SFSTs, which have only finitely many states, if SG functions cannot be modeled by SFSTs they must not be subsequential. To see why a SG function needs infinitely many states to model, consider inputs $+-\Box$ and $+-$. When the $+$ is read, the transducer can output $+$. When the $-$ is read, the transducer must output λ because it is not known whether there is a blocker coming later or not. Now if the blocker \Box is read the transducer can output $-\Box$. But if the $-$ is the end of the input the output function for that state instead outputs one $+$. Now consider inputs $+- - \Box$ and $+- -$. With each $-$ read, the transducer must output λ in case there is a blocker coming later. If no blocker is read, then the output function for this state (which is a different state than in the above case) would output the same number of $+$'s as λ 's, in this case two $+$'s. If a blocker is read the transducer can output $++\Box$. So the idea is that the state after reading the last $-$ need to be distinct for different n 's so that it can keep track of how many λ 's have been read and output the same number of $+$'s. Since there is no upper limit on the number n of $-$'s, and each n requires a different final state, it follows that there need to be infinitely many states. Hence SG functions cannot be modeled by SFSTs and are not subsequential.

One last OT variant that will be discussed is the Harmonic Serialism (HS) (McCarthy 2010; 2011). HS is a derivational version of OT where GEN is limited to making “one change at a time”. At each step, a limited set of candidates (the ones that differ from the step's input by at most one GEN-imposed change) are evaluated, and the winner is submitted as the input to another pass through GEN and EVAL, until the unchanged candidate wins (“convergence”) (McCarthy 2011). McCarthy (2011) showed that HS successfully accounts for the nasal spreading process found in Johore Malay with SHARE (F)

¹⁷ For more details on how to represent constraints with finite-state machines, see Riggle (2004: 33).

¹⁸ A relation R is length-preserving iff for all $(x, y) \in R$, it is the case that $|x| = |y|$.

constraints (as defined in (29)) without any SG-type effect. However, it is unclear how HS can eliminate SG in LDCA, at least not in its present form.

- (29) SHARE (F): Assign one violation mark for every pair of adjacent elements that are not linked to the same token of [F].

In Johore Malay, nasal spreads rightwards to vowels and glides. In a hypothetical form /mawara/, then, the liquid [r] will act like a blocker that blocks the spreading of the nasal feature. McCarthy showed that an AGREE-type constraint would fail as it predicts for Johore Malay that hypothetical /mawa/ → [mãwã] with total harmony, but hypothetical /mawara/ → [mawara] with no harmony at all, creating a SG-type effect. HS with SHARE ([nasal]) constraint, on the other hand, will correctly map /mawara/ → [mãwãra], as shown in (30).

- (30) Derivation for the mapping /mawara/ → [mãwãra], adapted from McCarthy (2011)

Step 1:

/m a w a r a/	*ř	SHARE ([nasal])	ID _{IO} [nasal]
a. mã w a r a		****	*
b. m a w a r a		*****!	

Step 2:

/mã w a r a/	*ř	SHARE ([nasal])	ID _{IO} [nasal]
a. mãw a r a		***	*
b. mã w a r a		*****!	

Step 3:

/mãw a r a/	*ř	SHARE ([nasal])	ID _{IO} [nasal]
a. mãwã r a		**	*
b. mãw a r a		***!	

Step 4: “convergence”

/mãwã r a/	*ř	SHARE ([nasal])	ID _{IO} [nasal]
a. mãwã r a		**	
b. mãwãr a	*!	*	

The successfulness of SHARE (F) in eliminating SG-type predictions for nasal spreading lies in the fact that it awards agreement, so that at each step in the derivation from /mawara/ → /mãwara/ → /mãwãra/ → /mãwãra/ the winning candidate incurs less and less SHARE ([nasal]) constraint, the blocker [r] can be protected by the highest ranked *ř constraint. A potential problem for HS is that the successfulness in eliminating SG-type predictions is partly dependent on how the underlying forms are linked to the feature nodes. Consider, for instance, the same example in (10). Under the basic constraint ranking * \boxplus >> SHARE (F) >> ID_{IO}, HS will map underlying /+|-|-|-| \boxplus / to [+ + + + \boxplus], avoiding the SG-type prediction successfully, similar to the nasal spreading case discussed in (30). In contrast, HS will make a different prediction for /+|- - -| \boxplus / where all four [-F]

segments are linked to the same [-F] feature node. /+|- - -□/ incurs only one violation of SHARE (F), and it is hard to see that there exists a harmonically improving path from /+ - - -□/ → /+ + - -□/ → /+ + + -□/ → [+ + + + □]. In general if the underlying form starts off with one violation of SHARE (F), then other candidates are harmonically bound by the faithful candidate. At Step 1, for instance, the faithful candidate /+|- - -□/ and /+ + +|-□/ both violate SHARE (F) once while /+ + +|-□/ will also incur violation of the relevant ID₁₀ constraint. HS will thus predict the winner to be [+ - - - □] without spreading in this case. Therefore, in conjunction with the principle of the rich base, HS predicts that underlying forms /+|- - - □/ and /+ + +|-□/ are contrastive in the sense that they lead to different surface forms. Overall, HS with SHARE (F) may solve the SG problem, but it is less clear whether this solution avoids other problematic typological predictions.¹⁹

5.2 Future work

Recent work have suggested that the computational nature of phonological processes might be less complex than the class of regular relations (Johnson 1972; Kaplan & Kay 1994). This paper has established that attested LDCA patterns are subsequential. Payne (2014) showed that long-distance dissimilation processes from Suzuki (1998) and Bennett (2013) are also subsequential. The classes of Input Strictly Local (ISL) and Output Strictly Local (OSL) functions, which are shown to be able to model local phonological processes are even less complex than subsequential functions (Chandlee et al. 2014; Chandlee 2014). However, ISL and OSL functions are too restrictive as they are not sufficient to model non-local mappings such as long-distance consonant or vowel harmony processes, nonetheless, this work points to an interesting direction for future research.

Chandlee et al. defined ISL and OSL functions based on properties of the class of subregular formal language called Strictly Local (SL) language (Rogers & Pullum 2011; Rogers et al. 2013). Different classes of subregular languages are useful in capturing natural language phonotactics. The SL language can model natural language phonotactics that are describable by constraints which pick out contiguous substrings bounded by some length *k*. Figure 8 shows the relevant Subregular Hierarchy. Similar to the Chomsky Hierarchy, the classes refer to sets of strings, and are classified according to the expressive power of each language class.

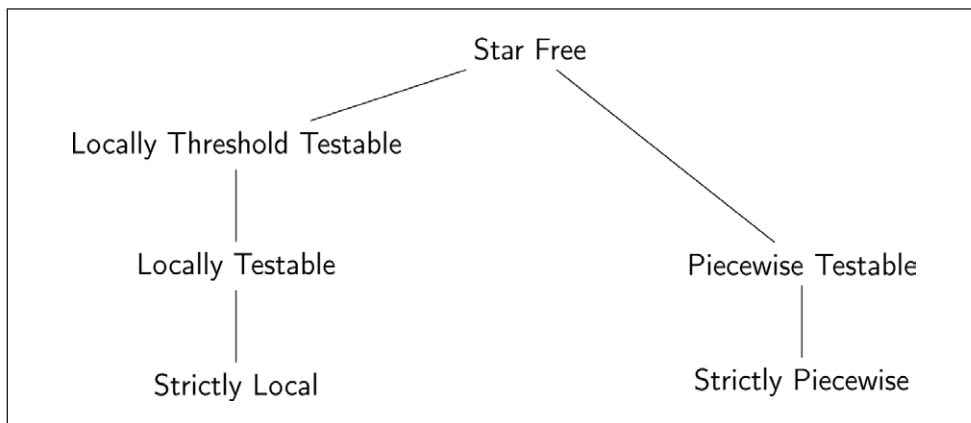


Figure 8: The Subregular Hierarchy for sets of strings.

¹⁹ An anonymous reviewer suggested that the problematic typological predictions could possibly be avoided depending on how features are linked and de-linked. A more detailed review of the possibilities within HS is beyond the scope of this paper.

Heinz (2010) suggested that unbounded harmony patterns are Strictly 2-Piecewise. Heinz et al. (2011) showed that long-distance phonotactics with blocking are Tier-based Strictly 2-Local. McMullin & Hansson argued that long-distance consonant agreement and disagreement are both Tier-based Strictly Local, and that transvocalic patterns are a special case of blocking (McMullin & Hansson 2016). Analogous to the fact that local phonotactic patterns can be characterized by Strictly Local languages, long-distance phonotactic patterns can be characterized by Strictly Piecewise (SP) languages (Heinz 2010). The sibilant harmony process in Sarcee, for example, is characterizable by a Strictly 2-Piecewise (SP_2) grammar. In Sarcee, sibilants agree in anteriority with the rightmost [-ant] sibilant. A SP_2 grammar will capture this by forbidding [+ant] sibilant followed by a [-ant] sibilant.²⁰ Informally this is equivalent to what Heinz (2010) referred to as the precedence grammar, specified by a list of precedence relations in a language. The Sarcee case, for instance, can be translate into a list of precedence relations in (31), specifying all precedence relations that are allowed in the language.

(31) A precedence grammar for Sarcee sibilant harmony:

[s] can be preceded by [s].
 [s] can be preceded by [ʃ].
 [s] can be preceded by [t].
 ...
 [t] can be preceded by [s].
 [t] can be preceded by [ʃ].
 ...
 [ʃ] can be preceded by [ʃ].
 [ʃ] can be preceded by [t].
 ...

The goal is to develop a sufficiently restrictive theory of phonology, irrespective of whether it takes a computational approach, or within the constraint-based or rule-based frameworks. This paper has taken a computational approach, and suggested that the subsequential boundary is sufficient to model the attested LDCA processes and effectively eliminates the pathological patterns. Yet as discussed previously, it is still possible that tighter bound exists. For future work, then, it appears promising to further investigate the various properties of SP languages (or other classes of languages shown in Figure 8) and develop the functional counterparts of such classes of languages to model phonological processes.

5.3 Conclusion

To conclude, this paper has pointed out that attested LDCA can be well accounted for by the existing theories, but they tend to have trouble eliminating the pathological patterns like Majority Rules and Sour Grapes from the typology. This paper has proposed an alternative approach from a computational perspective, and suggested that LDCA can be better characterized by the computational class of subsequential functions. It has been shown that attested LDCA patterns, divided into three major types, are all subsequential (by way of showing each type is describable by SFSTs). This paper further argues that subsequentiality is a desirable characterization of LDCA for the following reasons. First, it is sufficiently expressive, as it accounts for the attested patterns equally well as other existing theories. Second, it is restrictive as it has been shown that Majority Rules and

²⁰ For formal definition of the SP language readers are referred to Definition 2 in Rogers et al. (2010).

Sour Grapes are not subsequential; the subsequential boundary is therefore successful in eliminating the pathological patterns from LDCA typology. Additionally, such a computational characterization offers a way to evaluate complexity on the basis of expressivity. Being subsequential suggests that LDCA processes are computationally less complex than regular, lending support to the hypothesis that the computational nature of phonological processes might in fact be more restricted than previously realized.

Abbreviations

FEM = feminine, LOCSG = locative singular

Competing Interests

The author has no competing interests to declare.

References

- Andersen, Torben. 1999. Consonant alternation and verbal morphology in Mayak (northern Burun). *Afrika und Übersee* 82(1). 65–97.
- Applegate, Richard B. 1972. *Ineseño Chumash grammar*. Berkeley, CA: University of California, Berkeley dissertation.
- Bakovic, Eric. 2000. *Harmony, dominance and control*. New Brunswick, NJ: Rutgers University dissertation.
- Bennett, William G. 2013. *Dissimilation, consonant harmony, and surface correspondence*. New Brunswick, NJ: Rutgers University dissertation.
- Chandlee, Jane. 2014. *Strictly local phonological processes*. Newark, DE: University of Delaware dissertation.
- Chandlee, Jane & Jeffrey Heinz. 2012. Bounded copying is subsequential: Implications for metathesis and reduplication. In *Proceedings of the 12th meeting of the special interest group on computational morphology and phonology*, 42–51.
- Chandlee, Jane, Rémi Eyraud & Jeffrey Heinz. 2014. Learning strictly local subsequential functions. *Transactions of the Association for Computational Linguistics* 2. 491–503.
- Chandlee, Jane, Rémi Eyraud & Jeffrey Heinz. 2015. Output strictly local functions. In *Proceedings of the 14th meeting on the mathematics of language*. 112–125. DOI: <https://doi.org/10.3115/v1/w15-2310>
- Chomsky, Noam. 1956. Three models for the description of language. *IRE Transactions on Information Theory* 2(3). 113–124. DOI: <https://doi.org/10.1109/TIT.1956.1056813>
- Ebert, Karen Heide. 1979. *Sprache und tradition der Kera (Tschad)*. Berlin: Reimer.
- Eisner, Jason. 2000. Directional constraint evaluation in Optimality Theory. In *Proceedings of the 18th conference on computational linguistics*, 257–263. DOI: <https://doi.org/10.3115/990820.990858>
- Finley, Sara. 2008. *The formal and cognitive restrictions on vowel harmony*. Baltimore, MD: Johns Hopkins University dissertation.
- Frank, Robert & Giorgio Satta. 1998. Optimality Theory and the generative complexity of constraint violability. *Computational Linguistics* 24(2). 307–315.
- Gafos, Adamantios I. 1996. *The articulatory basis of locality in phonology*. Baltimore, MD: Johns Hopkins University dissertation.
- Gerdemann, Dale & Gertjan van Noord. 2000. Approximation and exactness in finite state Optimality Theory. In Lauri Karttunen, Jason Eisner & Alain Thériault (eds.), *Siphon 2000, finite state phonology*.
- Hansson, Gunnar Ólafur. 2001. *Theoretical and typological issues in consonant harmony*. Berkeley, CA: University of California, Berkeley dissertation.
- Hansson, Gunnar Ólafur. 2007. Blocking effects in agreement by correspondence. *Linguistics Inquiry* 38(2). 395–409. DOI: <https://doi.org/10.1162/ling.2007.38.2.395>


- Hansson, Gunnar Ólafur. 2010. *Consonant harmony: Long-distance interaction in phonology*. Berkeley, CA: University of California Press.
- Hayward, Richard J. 1988. *Remarks on the Omotic sibilants*. Hamburg: Helmut Buske.
- Hayward, Richard J. 1990. Notes on the Aari language. *Omotic Language Studies*. 425–493.
- Heinz, Jeffrey. 2009. On the role of locality in learning stress patterns. *Phonology* 26. 303–351. DOI: <https://doi.org/10.1017/S0952675709990145>
- Heinz, Jeffrey. 2010. Learning long-distance phonotactics. *Linguistics Inquiry* 41. 623–661. DOI: https://doi.org/10.1162/LING_a_00015
- Heinz, Jeffrey, Chetan Rawal & Herbert G. Tanner. 2011. Tier-based strictly local constraints for phonology. In *Proceedings of the 49th annual meeting of the association for computational linguistics*.
- Heinz, Jeffrey & Regine Lai. 2013. Vowel harmony and subsequentiality. In *Proceedings of the 13th meeting on mathematics of language*, 52–63.
- Hopcroft, John E. & Jeffery D. Ullman. 1979. *Introduction to automata theory, languages, and computation*. Reading, MA: Addison-Wesley.
- Hyman, Larry M. 1995. Nasal consonant harmony at a distance: The case of Yaka. *Studies in African Linguistics* 24(1). 5–30.
- Johnson, C Douglas. 1972. *Formal aspects of phonological description*. Berlin: De Gruyter Mouton.
- Kaplan, Ronald M. & Martin Kay. 1994. Regular models of phonological rule systems. *Computational Linguistics* 20(3). 331–378.
- Karttunen, Lauri. 1998. The proper treatment of optimality in computational phonology. In *Proceedings of the international workshop on finite state methods in natural language processing*, 1–12. DOI: <https://doi.org/10.3115/1611533.1611534>
- Lombardi, Linda. 1999. Positional faithfulness and voicing assimilation in Optimality Theory. *Natural Language and Linguistic Theory* 17. 267–302. DOI: <https://doi.org/10.1023/A:1006182130229>
- Lombardi, Linda. 2001. Why place and voice are different: Constraint-specific alternations in Optimality Theory. In Linda Lombardi (ed.), *Segmental phonology in Optimality Theory: Constraints and representations*, Cambridge: Cambridge University Press. DOI: <https://doi.org/10.1017/CBO9780511570582.002>
- MacDonell, Arthur A. 1910. *Vedic grammar*. Strassburg.
- McCarthy, John J. 2010. An introduction to harmonic serialism. *Language and Linguistics Compass* 4(10). 1001–1018. DOI: <https://doi.org/10.1111/j.1749-818X.2010.00240.x>
- McCarthy, John J. 2011. Autosegmental spreading in Optimality Theory. *Tones and Features: Phonetic and Phonological Perspectives*. 195–222.
- McCarthy, John J & Alan Prince. 1995. Faithfulness and reduplicative identity. In *Papers in Optimality Theory, University of Massachusetts occasional papers*, Amherst, MA: University of Massachusetts Press, 249–384.
- McDonough, Joyce. 1991. On the representation of consonant harmony in Navajo. In *Proceedings of west coast conference on formal linguistics*.
- McGregor, William. 1990. *A functional grammar of Gooniyandi*. Amsterdam: John Benjamins. DOI: <https://doi.org/10.1075/slcs.22>
- McMullin, Kevin & Gunnar Ólafur Hansson. 2016. Long-distance phonotactics as tier-based strictly 2-local languages. In *Proceedings of annual meetings on phonology*, 19–21. DOI: <https://doi.org/10.3765/amp.v2i0.3750>
- Mohri, Mehryar. 1997. Finite-state transducers in language and speech processing. *Computational Linguistics* 23(2). 269–311.
- Odden, David. 1994. Adjacency parameters in phonology. *Language* 70. 289–330. DOI: <https://doi.org/10.2307/415830>

- Oncina, José, Pedro García & Enrique Vidal. 1993. Learning subsequential transducers for pattern recognition interpretation tasks. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 15(5). 448–458. DOI: <https://doi.org/10.1109/34.211465>
- Padgett, Jaye. 1995. Partial class behavior and nasal place assimilation. In *Proceedings of the arizona phonology conference: Workshop on features in optimality theory*. 145–183.
- Partee, Barbara B.H., Alice G. ter Meulen & Robert Wall. 2012. *Mathematical methods in linguistics*, vol. 30. New York, NY: Springer Science & Business Media.
- Payne, Amanda. 2014. Dissimilation as a subsequential process. In *Proceedings of 44th meeting of the north east linguistic society*. 79–90.
- Rapold, Christian J. 2006. *Towards a grammar of Benchnon*. Leiden: Leiden University dissertation.
- Riggle, Jason Alan. 2004. *Generation, recognition, and learning in finite state Optimality Theory*. Los Angeles, CA: University of California, Los Angeles dissertation.
- Rogers, James & Geoffrey K. Pullum. 2011. Aural pattern recognition experiments and the subregular hierarchy. *Journal of Logic, Language and Information* 20. 329–342. DOI: <https://doi.org/10.1007/s10849-011-9140-2>
- Rogers, James, Jeffrey Heinz, Gil Bailey, Matt Edlefsen, Molly Visscher, David Wellcome & Sean Wibel. 2010. On languages piecewise testable in the strict sense. In *The mathematics of language*, 255–265. Berlin: Springer. DOI: https://doi.org/10.1007/978-3-642-14322-9_19
- Rogers, James, Jeffrey Heinz, Margaret Fero, Jeremy Hurst, Dakotah Lambert & Sean Wibel. 2013. Cognitive and sub-regular complexity. In Glyn Morrill & Mark-Jan Nederhof (eds.), *Formal grammar*, vol. 8036 (Lecture Notes in Computer Science), Berlin: Springer. DOI: https://doi.org/10.1007/978-3-642-39998-5_6
- Rose, Sharon & Rachel Walker. 2004. A typology of consonant agreement as correspondence. *Language* 80(3). 475–531. DOI: <https://doi.org/10.1353/lan.2004.0144>
- Schein, Barry & Donca Steriade. 1986. On geminates. *Linguistics Inquiry* 17(4). 691–744.
- Schieber, Stuart M. 1985. Evidence against the context-freeness of natural language. In Walter J. Savitch, Emmon Bach, William Marsh & Gila Safran-Naveh (eds.), *The formal complexity of natural language*, 320–334. Dordrecht: Springer Netherlands.
- Seyoum, Mulugeta. 2008. *A grammar of Dime*. Leiden: Leiden University dissertation.
- Stemberger, Joseph Paul. 1988. Between-word processes in child phonology. *Journal of Child Language* 15. 39–61. DOI: <https://doi.org/10.1017/S0305000900012046>
- Stemberger, Joseph Paul. 1993. Glottal transparency. *Phonology* 10. 107–138. DOI: <https://doi.org/10.1017/S0952675700001755>
- Suzuki, Keiichiro. 1998. *A typological investigation of dissimilation*. Tucson, AZ: University of Arizona dissertation.
- Walker, Rachel. 1997. Faith and markedness in Esimbi feature transfer. *Phonology at Santa Cruz* 5. 103–115.
- Wilson, Colin. 2003a. Analyzing unbounded spreading with constraints: Marks, targets, and derivations. Unpublished manuscript, UCLA.
- Wilson, Colin. 2003b. Experimental investigation of phonological naturalness. In *Proceedings of the 22nd west coast conference on formal linguistics*, vol. 22. 533–546.

How to cite this article: Luo, Huan. 2017. Long-distance consonant agreement and subsequentiality. *Glossa: a journal of general linguistics* 2(1): 52. 1–25, DOI: <https://doi.org/10.5334/gjgl.42>

Published: 31 May 2017

Copyright: © 2017 The Author(s). This is an open-access article distributed under the terms of the Creative Commons Attribution 4.0 International License (CC-BY 4.0), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited. See <http://creativecommons.org/licenses/by/4.0/>.

 *Glossa: a journal of general linguistics* is a peer-reviewed open access journal published by Ubiquity Press.

OPEN ACCESS 