Precedence is Pathological: The Problem of Alphabetical Sorting

Andrew Lamont

1. Introduction

This paper presents two results on the computational expressivity of parallel Optimality Theory (OT) and Harmonic Serialism (HS) (Prince & Smolensky, 1993/2004; McCarthy, 2000) and their sensitivities to different types of markedness constraints. The computational expressivity of a model refers to the complexity of the functions it can compute. Phonological transformations, i.e. functions that map underlying representations onto surface forms, are computationally REGULAR (Johnson, 1972; Kaplan & Kay, 1994), meaning they are only as expressive as FINITE STATE TRANSDUCERS (FST). Models of phonology need to be expressive enough to model regular relations, and should be restrictive enough, so that they cannot model more complex functions. This is not the case for OT, which has been shown to model processes more expressive than the regular relations (Eisner, 1997, 2000; Frank & Satta, 1998).

While OT is known to be too computationally expressive, the extent of its expressivity is not known (Heinz, 2018), and the expressivity of HS has only recently been studied (Hao, 2017, 2018). This paper demonstrates that both OT and HS are capable of modeling alphabetical sorting, a process so complex that it requires the use of a TURING MACHINE. Turing Machines are far more expressive than FSTs, and are able to model any function that a computer program can model (Hopcroft et al., 2008). Crucially, this extraordinary expressivity is achieved not with extraordinary means, but rather with standard types of markedness constraints, termed here *local* and *global*. Local constraints penalize *substrings* of a certain length; a common example being AGREE(PLACE), which assigns violations to contiguous pairs of segments with different specifications of the feature [place]. Global constraints penalize *subsequences* of a certain length; a common example being *[+lateral]. . . [+lateral], which assigns violations to possibly non-contiguous pairs of laterals. These are discussed in detail in §2. This paper shows that whereas OT can model alphabetical sorting with both local and global constraints, HS can only model alphabetical sorting with global constraints. HS displays a dramatic sensitivity to the type of markedness constraint used, and cannot model alphabetical sorting with local constraints.

The main result of this paper is that both OT and HS severely overgenerate. Phonological models need only be as expressive as FSTs, but OT and HS can model far more expressive functions. However, the secondary result is that whether HS overgenerates in this way crucially depends on the type of markedness constraint used. In this particular test case, HS is only able to model alphabetical sorting with global markedness constraints. This suggests that there may be a variant of HS that is restricted to modeling regular relations, making it a promising model to study from a computational perspective. A general argument is beyond the scope of this paper; for now, I hypothesize that CON does not contain global markedness constraints. This removes alphabetical sorting from HS and is supported by the hypothesis that all phonological interactions involve nodes that are adjacent on some tier (Odden, 1994).

The rest of the paper is organized as follows. Section 2 defines alphabetical sorting and provides formal background on the types of markedness constraints under investigation. Section 3 presents the implementations of alphabetical sorting in OT and HS. Section 4 concludes.

* University of Massachusetts Amherst, alamont@linguist.umass.edu. This work has greatly benefited from discussions with Jeff Heinz, Gaja Jarosz, John McCarthy, Charlie O'Hara, Joe Pater, Kristine Yu, participants in the UMass Sound Workshop, audiences at Swarthmore College, PhoNE 2018, and WCCFL 2018, and three anonymous reviewers for WCCFL 2018. All remaining errors are of course my own.

 Proceedings of the 36th West Coast Conference on Formal Linguistics, ed. Richard Stockwell et al., 243-249. Somerville, MA: Cascadilla Proceedings Project.

2. What alphabetical sorting is and why it is pathological

This section provides formal background on local and global markedness constraints. But first, it is important to define the pathology under investigation. Alphabetical sorting is the process of reordering a finite list of objects according to a prescribed order; this is a well-studied problem for which numerous algorithms have been developed (Knuth, 1973). In this paper, the lists to be sorted are strings made of letters from the Latin alphabet. For example, an input ✗/phonology/ is not in-order, as, for example, p comes after h in the alphabet, and, after sorting, is transformed to the in-order output ✓[ghlnooopy]. There are two distinct but interrelated aspects of alphabetical sorting: the *language*, the set of strings whose letters are in alphabetical order, and the *map*, the function that transforms inputs into alphabetized outputs. While the language is computationally simple, having the same expressivity as most phonotactic patterns (Heinz, 2018), the map is pathologically complex, requiring the power of a Turing Machine.

Viewing sorting as a function, the language is its range, or the set of legal strings. This is exactly the set of strings whose letters are in alphabetical order; these can be thought of as phonotactically well-formed. For example, the language contains the string ✓[aegilops], where the letters are in alphabetical order, but does not contain the string ✗[spoonfeed], where the letters are in reverse alphabetical order. Just like all attested phonotactic generalizations, the language is regular: the set of all legal strings is generated by the REGULAR EXPRESSION $a^*b^*c^*d^*e^*f^*g^*h^*i^*j^*k^*l^*m^*n^*o^*p^*q^*r^*s^*t^*u^*v^*w^*x^*y^*z^*$.

Not only is the alphabetical language regular, but it belongs to the same subregular classes that most phonotactic patterns belong to (Heinz, 2018). Specifically, the language can be expressed as a CONJUNCTION OF NEGATIVE LITERALS, a finite list of banned substructures (Rogers et al., 2013), all of length 2: $(\neg ba) \wedge (\neg ca) \wedge (\neg cb) \wedge \ldots \wedge (\neg zx) \wedge (\neg zy)$. Literals refer to substructures that are defined by one of two ordering parameters: *successor* or *precedence* relations. Successor relations define literals as *substrings*; using them defines the alphabetical language as the set of strings that do not contain the banned substrings $\{ba, ca, cb, \ldots\}$. That approach bans ✗[spoon] because it contains the out-of-order substrings $\{sp, po, on\}$. Precedence relations define literals as *subsequences*; using them defines the alphabetical language as the set of strings that do not contain the banned subsequences $\{b \ldots a, c \ldots a, c \ldots b, \ldots\}$. That approach bans ✗[spoon] because it contains the out-of-order subsequences $\{s \ldots p, s \ldots o, s \ldots n, p \ldots o, p \ldots n, o \ldots n\}$. The choice of successor or precedence relations does not matter in defining the language. However, after formalizing the banned substructures as markedness constraints (§3), this choice crucially affects HS, which can only model sorting when precedence relations are used.

Not only can the alphabetical language be defined in terms of the same *types* of substructures as phonotactics, but also there are attested patterns that impose set orders on substrings and subsequences. For example, Bayso allows /n/ + coronal obstruent clusters, but the reverse undergo metathesis (Garrett & Blevins, 2009): /wod-ne/ → [wo<u>nd</u>e] *[wodne] 'we drove' ✗[dn] ✓[nd]. In Sarcee, [−anterior] sibilants can precede [+anterior] sibilants, but the reverse undergo harmony (Cook, 1979; Heinz, 2010): /si-tʃiz-aʔ/ → [<u>ʃ</u>ítʃídzà] *[sí<u>ts</u>ídzà] 'my duck' ✗[s…ʃ] ✓[ʃ…s]. This reinforces the notion that the language is anything but pathological. It is both computationally simple and empirically grounded.

In contrast to the language, the map is far more complex. The map is the function between inputs and outputs, or the actual process of sorting. Computationally, it is the map that is pathological, as it cannot be modeled by an FST. I do not provide a formal proof here, just the intuition: sorting requires infinite memory and FSTs can only remember a finite number of things. Suppose we have an input with some number m of a's and some number n of b's, all shuffled up. This should be mapped onto the output $a^m b^n$. As the FST reads the input, it can safely write all the a's it reads directly to the output, but it has to wait to start writing the b's. The FST only gets one pass through the input, so it has to remember exactly how many b's there are in order to write the correct number to the output. Because we cannot know in advance how many b's there are, this requires counting arbitrarily high, which is beyond the capacity of FSTs. Because alphabetical sorting cannot be modeled by an FST, it is not a regular relation, and therefore requires more computational power than any phonological transformation.

As a reviewer points out, the attested phonotactic patterns I draw comparison to are different in that they state ordering preferences over single features, not entire inventories. This is true, but as noted above, even if sorting is restricted to 2 segments, it still exceeds the expressivity of FSTs and is pathological. The main result of this paper could equally be made with a hypothetical variant of Sarcee where sibilants are reordered rather than undergoing harmony.

3. How to model alphabetical sorting in OT & HS

As Section 2 laid out, the alphabetical language can be defined either by banning out-of-order substrings or out-of-order subsequences. These two approaches can be formalized in OT and HS as the markedness constraints (1) and (2) below, respectively. *BA assigns violations to out-of-order substrings, and *B...A assigns violations to out-of-order subsequences. The inequality notation indicates relative order in the alphabet, e.g., p is higher in the alphabet than h, which is expressed by the inequality $p > h$.

(1) *BA Assign one violation for every substring YX where $Y > X$ (local)

(2) *B...A Assign one violation for every subsequence $Y \ldots X$ where $Y > X$ (global)

As defined in (1-2), each markedness constraint ranges over every out-of-order pair of segments. These can also be replaced with families of constraints that penalize specific ordered pairs. For example, *BA could be replaced with *ba, *ca, *cb, and so on. Because the relative ranking between such atomic constraints would be irrelevant for alphabetical sorting, the cover constraints are used for succinctness.

The terms *local* and *global* markedness are used to refer to how segments can interact in defining loci of violation. With the local markedness constraint *BA, segments only interact with their immediate neighbors, and any given segment can define at most 2 loci: one with the adjacent segment to its left, and one with the adjacent segment to its right. For example, in the out-of-order ✗[spoon], the p defines two loci of violation of *BA: sp and po. In contrast, with the global markedness constraint *B...A, all segments interact with each other, and any given segment can define as many loci of violation as there are other segments in the string: one with each preceding and following segment. The p in ✗[spoon] defines four loci of violation of *B...A: $s \ldots p$, $p \ldots o$ (one for each o), and $p \ldots n$.

Alphabetical sorting involves reordering segments, so these constraints must dominate a faithfulness constraint that penalizes metathesis. This paper adopts the constraint LINEARITY, which assigns violations for subsequences in the input whose correspondents in the output are in a different order (McCarthy & Prince, 1995). The locally defined CONTIGUITY family (Heinz, 2005) can also be used. This choice of faithfulness constraint will only decide between candidates that satisfy the markedness constraints, and so is irrelevant for our purposes. Because the markedness constraints can also be satisfied by changing or deleting segments, IDENT and MAX are assumed to be ranked high enough to make metathesis the only possible repair.

3.1. Alphabetical sorting in parallel Optimality Theory

OT can model sorting with both local and global markedness constraints. Tableaux (3-4) illustrate the out-of-order input /bba/ being mapped onto the in-order output [abb] with local and global markedness, respectively. Underlining indicates which segments have been reordered in candidates. Candidates where a b comes before the a violate the markedness constraints, because they contain out-of-order substrings or subsequences. These candidates lose to the alphabetized candidate [abb], which satisfies the markedness constraints, and is chosen as optimal.[1]

(3) /bba/ → [abb] (local markedness)

/bba/	*BA	LINEARITY
a. bba	W 1	L
b. <u>bb</u>a	W 1	L 1
c. b<u>ab</u>	W 1	L 1
→ d. <u>abb</u>		2

(4) /bba/ → [abb] (global markedness)

/bba/	*B...A	LINEARITY
a. bba	W 2	L
b. <u>bb</u>a	W 2	L 1
c. b<u>ab</u>	W 1	L 1
→ d. <u>abb</u>		2

[1] To be precise, there are two alphabetized candidates, which input-output correspondence distinguishes: [ab_1b_2] and [ab_2b_1]. The optimal alphabetized candidate is the one that incurs the fewest violations of LINEARITY. For space, these tableaux do not show every logically possible candidate.

In OT, GEN is unrestricted, and produces a candidate set that includes every permutation of the segments in the input. Those candidates whose segments are in alphabetical order satisfy the markedness constraints, and are preferred over candidates that do not. In general then, with either local or global markedness constraints in this ranking, OT always maps inputs onto an alphabetized output.

3.2. Alphabetical sorting with global markedness in Harmonic Serialism

In HS, GEN is restricted to only producing candidates that differ from the input via the application of at most one unfaithful operation; I assume local metathesis, i.e. reordering a substring of length 2, is among those operations (cf. Takahashi (2018)). Tableaux (5-6) show the first two steps of the derivation mapping /bba/ onto [abb] with global markedness. In the first step (5), the fully faithful candidate contains two out-of-order subsequences $b \ldots a$, and loses to the candidate [bab], where the a has moved to the left, removing one $b \ldots a$ subsequence. This candidate serves as the input to the next step of the derivation (6), where it loses to the candidate [abb], where the a moves once again, fully satisfying the markedness constraint. The derivation converges in the next step on the alphabetized output [abb].

(5) /bba/ → [abb]: Step 1 (global markedness)

/bba/	*B...A	LINEARITY
a. bba	W 2	L
b. $\underline{\text{bb}}$a	W 2	1
→ c. b$\underline{\text{ab}}$	1	1

(6) /bba/ → [abb]: Step 2 (global markedness)

/bab/	*B...A	LINEARITY
a. bab	W 1	L
→ b. $\underline{\text{ab}}$b		1
c. b$\underline{\text{ba}}$	W 2	1

As these tableaux illustrate, each application of metathesis improves on *B...A until the derivation converges on the alphabetized output. This is always the case with global markedness. To see why, consider an abstract input /$LYXR$/ mapping onto the output [$L\underline{XY}R$], where YX is out-of-order ($Y > X$), and L and R are some left and right contexts that may be segments or longer strings. The subsequences of /$LYXR$/ are the set $\{L \ldots Y, L \ldots X, L \ldots R, Y \ldots X, Y \ldots R, X \ldots R\}$: L precedes everything, R follows everything, and Y precedes X. Reordering Y and X gives the string [$L\underline{XY}R$], whose subsequences are the set $\{L \ldots Y, L \ldots X, L \ldots R, \underline{X \ldots Y}, Y \ldots R, X \ldots R\}$. In this string, L still precedes everything and R still follows everything; no subsequences involving L or R have changed. The only difference is that X now precedes Y: the subsequence $Y \ldots X$ has been replaced with $X \ldots Y$. Because Y is higher in the alphabet than X, this change removes one out-of-order subsequence, and with it, one violation of *B...A. It follows that reordering out-of-order substrings always improves on *B...A, and faithful candidates with out-of-order substrings will always lose to unfaithful candidates. A series of local swaps is guaranteed to eventually alphabetize the input; this is a known sorting algorithm called Bubble Sort (Knuth, 1973). With global markedness constraints, HS therefore maps all inputs onto alphabetized outputs.

3.3. No alphabetical sorting with local markedness in Harmonic Serialism

Unlike OT, HS is sensitive to the type of markedness constraint used, and cannot model alphabetical sorting with local markedness. Local markedness behaves differently exactly because it evaluates substrings, not subsequences. Consider again the abstract input /$LYXR$/ mapping onto the output [$L\underline{XY}R$], where YX is out-of-order ($Y > X$). While this change only affects one subsequence, replacing $Y \ldots X$ with $X \ldots Y$, it affects multiple substrings. The substrings of length 2 of /$LYXR$/ are the set $\{LY, YX, XR\}$, and the substrings of [$L\underline{XY}R$] are the set $\{\underline{LX}, \underline{XY}, \underline{YR}\}$. Reordering may therefore create new out-of-order substrings, and fail to improve on *BA. This always happens eventually, and the derivation may converge on an output containing out-of-order segments.

Tableaux (7-8) illustrate this point with derivations that converge in the first step. In (7), every candidate contains the out-of-order substring ba, violating *BA once. Only the fully faithful candidate [bba] satisfies LINEARITY, and is chosen as optimal. Similarly, in (8), the fully faithful candidate [cdab] contains the out-of-order substring da, violating *BA once. The losing unfaithful candidates all contain two out-of-order substrings, violating *BA twice, in addition to violating LINEARITY.

(7) /bba/ → [bba]: Step 1 (local markedness)

/bba/	*BA	LINEARITY
→ a. bba	1	
b. $\underline{\text{bb}}$a	1	W 1
c. b$\underline{\text{ab}}$	1	W 1

(8) /cdab/ → [cdab]: Step 1 (local markedness)

/cdab/	*BA	LINEARITY
→ a. cdab	1	
b. $\underline{\text{dc}}$ab	W 2	W 1
c. c$\underline{\text{ad}}$b	W 2	W 1
d. cd$\underline{\text{ba}}$	W 2	W 1

Even when reordering segments is possible, it is impossible for a segment to move more than once. Tableaux (9-10) illustrate a derivation getting stuck with the input /bcba/. In the first step (9), the fully faithful candidate [bcba] contains more out-of-order substrings than the unfaithful candidates [bbca] and [bcba], which tie. The tie is arbitrarily broken by choosing [bbca] as the winner; readers can confirm that choosing [bcab] will yield the same result in the next step. For this string to be alphabetized, the a has to move leftwards, forcing the c to move a second time: bbca → bb$\underline{\text{ac}}$ → b$\underline{\text{ab}}$c → [$\underline{\text{ab}}$bc]. This is not possible. The derivation converges in the second step (10), with the fully faithful candidate harmonically bounding the unfaithful candidates. Although there is not room to fully demonstrate why, this is the case in general. No segment can ever move more than once with local markedness, echoing McCarthy's (2008) argument that ONSET and NOCODA cannot motivate multiple metatheses in HS.

(9) /bcba/ → [bbca]: Step 1 (local markedness)

/bcba/	*BA	LINEARITY
a. bcba	W 2	L
b. $\underline{\text{cb}}$ba	W 2	1
→ c. b$\underline{\text{bc}}$a	1	1
d. bc$\underline{\text{ab}}$	1	1

(10) /bcba/ → [bbca]: Step 2 (local markedness)

/bbca/	*BA	LINEARITY
→ a. bbca	1	
b. $\underline{\text{bb}}$ca	1	W 1
c. b$\underline{\text{cb}}$a	W 2	W 1
d. bb$\underline{\text{ac}}$	1	W 1

With local markedness in HS, metathesis is only possible in a restricted number of environments, and no segment can move more than once. Derivations may reorder a number of non-overlapping substrings, but, unless the underlying representation is almost completely alphabetized, will not converge on an alphabetized output. Depending on how ties are treated, the function this system models is STRICTLY LOCAL (Chandlee, 2014), and can be modeled by an FST reading the input in a window of a fixed length. This illustrates the sensitivity of HS to the different types of markedness constraints in this test case. With global markedness, HS can model alphabetical sorting, but with local markedness, HS is limited to modeling a subregular function.

4. Conclusion

This paper has demonstrated that both OT and HS can model alphabetical sorting. The expressivity of these models therefore dramatically exceeds that of Finite State Transducers, which are sufficient for modeling phonological transformations. While both OT and HS overgenerate, HS displays a sensitivity to CON that OT lacks. Whereas OT can model alphabetical sorting with both local and global markedness constraints, HS can only model sorting with global markedness constraints. This suggests that imposing formal restrictions on both GEN and CON may produce a computationally restrictive model.

In HS, GEN is limited to only producing candidates that differ from the input via the application of at most one unfaithful operation. This limitation has been argued for on empirical grounds, as it appropriately restricts the typological predictions HS makes (McCarthy, 2016). As for CON, Heinz (2018) proposes that all markedness constraints penalize substrings, subsequences, or substrings defined over some tier, defining a Constraint Definition Language (CDL) in the sense of de Lacy (2011). These constraint types correspond to the STRICTLY LOCAL (McNaughton & Papert, 1971), STRICTLY PIECEWISE (Heinz, 2007, 2010; Rogers et al., 2010; Graf, 2017), and TIER-BASED STRICTLY LOCAL (Heinz et al., 2011) formal languages that phonotactic generalizations have been shown to belong to (Heinz, 2018). This is an important hypothesis that defines a testable space for investigation. This paper has demonstrated that markedness constraints defined over subsequences allows HS to model alphabetical sorting. Eisner (1997, 2000) raises a similar point with respect to the Midpoint Pathology, which depends on Alignment constraints defined over subsequences. I advocate for investigating a tighter CDL that excludes markedness constraints defined over subsequences.

The consequences of such a restriction require future research, but defining patterns over subsequences may not be empirically necessary. For example, long-distance segmental processes have been argued to be captured by local constraints defined over tiers (McMullin & Hansson, 2015; McMullin, 2016). Problematically however, HS cannot model iterative harmony with these types of constraints (Pater et al., 2007), and so the overgeneration problem is replaced with an undergeneration problem. In these cases, directionally evaluated constraints à la Eisner (2000) seem to be a promising replacement, and I have ongoing work investigating their use in HS (Lamont, in prep.). It remains to be seen whether this approach fully generalizes, and whether such a version of HS is computationally restrictive.

References

Chandlee, Jane (2014). *Strictly local phonological processes*. Ph.D. thesis, University of Delaware.

Cook, Eung-Do (1979). Palatalization and related rules in Sarcee. Cook, Eung-Do & Jonathan Kaye (eds.), *Linguistic Studies of Native Canada*, University of British Columbia Press, Vancouver, 19–35.

de Lacy, Paul (2011). Markedness and faithfulness constraints. van Oostendorp, Marc, Colin Ewen, Elizabeth Hume & Keren Rice (eds.), *The Blackwell Companion to Phonology*, Wiley-Blackwell, Malden, MA, 1492–1512.

Eisner, Jason (1997). What constraints should OT allow? Paper presented at LSA 71.

Eisner, Jason (2000). Directional constraint evaluation in Optimality Theory. *Proceedings of the 18th International Conference on Computational Linguistics*, 257–263.

Frank, Robert & Giorgia Satta (1998). Optimality Theory and the generative complexity of constraint violability. *Computational Linguistics* 24:2, 307–315.

Garrett, Andrew & Juliette Blevins (2009). Analogical morphophonology. Inkelas, Sharon & Kristin Hanson (eds.), *The Nature of the Word: Essays in Honor of Paul Kiparsky*, MIT Press, Cambridge, 527–546.

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

Hao, Yiding (2017). Harmonic Serialism and finite-state Optimality Theory. *Proceedings of the 13th International Conference on Finite State Methods and Natural Language Processing*, 20–29.

Hao, Yiding (2018). Finite-state models of Harmonic Serialism. Poster presented at PLC 42.

Heinz, Jeffrey (2005). Reconsidering linearity: Evidence from CV metathesis. *Proceedings of the 24th West Coast Conference on Formal Linguistics*, 200–208.

Heinz, Jeffrey (2007). *Inductive Learning of Phonotactic Patterns*. Ph.D. thesis, University of California Los Angeles.

Heinz, Jeffrey (2010). Learning long-distance phonotactics. *Linguistic Inquiry* 41, 623–661.

Heinz, Jeffrey (2018). The computational nature of phonological generalizations. Hyman, Larry M. & Frans Plank (eds.), *Phonological Typology*, Mouton, 126–195.

Heinz, Jeffrey, Chetan Rawal & Herbert Tanner (2011). Tier-based strictly local constraints for phonology. *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics*, 58–64.

Hopcroft, John E., Rajeev Motwani & Jeffrey D. Ullman (2008). *Introduction to Automata Theory, Languages, and Computation*. Pearson, 3 edn.

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

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

Knuth, Donald E. (1973). *The Art of Computer Programming*, vol. 3 / Sorting and Searching. Addison-Wesley Publishing, Reading, MA.

Lamont, Andrew (in prep.). Directional constraint evaluation in Harmonic Serialism. Unpublished manuscript.

McCarthy, John J. (2000). Harmonic serialism and parallelism. Hirotani, Masako, Andries Coetzee, Nancy Hall & Ji-Yung Kim (eds.), *Proceedings of the North East Linguistics Society 30*, GLSA, Amherst, MA, 501–524.

McCarthy, John J. (2008). Restraint of analysis. Blaho, Sylvia & Patrik Bye (eds.), *Freedom of Analysis?*, De Gruyter, 203–231.

McCarthy, John J. (2016). The theory and practice of Harmonic Serialism. McCarthy, John J. & Joe Pater (eds.), *Harmonic Grammar and Harmonic Serialism*, Equinox Publishing, London, 47–87.

McCarthy, John J. & Alan Prince (1995). Faithfulness and reduplicative identity. Beckman, Jill, Suzanne Urbanczyk & Laura Walsh Dickey (eds.), *University of Massachusetts Occasional Papers in Linguistics 18: Papers in Optimality Theory*, GSLA, Amherst, MA, 249–384.

McMullin, Kevin (2016). *Tier-Based Locality in Long-Distance Phonotactics: Learnability and Typology*. Ph.D. thesis, The University of British Columbia.

McMullin, Kevin & Gunnar Ólafur Hansson (2015). Long-distance phonotactics as tier-based strictly 2-local languages. Albright, Adam & Michelle A. Fullwood (eds.), *Proceedings of the 2014 Annual Meeting on Phonology*.

McNaughton, Robert & Seymour Papert (1971). *Counter-Free Automata*. The MIT Press, Cambridge, MA.

Odden, David (1994). Adjacency parameters in phonology. *Language* 70:2, 289–330.

Pater, Joe, Rajesh Bhatt & Christopher Potts (2007). Linguistic optimization. Unpublished manuscript.

Prince, Alan & Paul Smolensky (1993/2004). *Optimality Theory: Constraint Interaction in Generative Grammar*. Blackwell Publishing, Malden, MA.

Rogers, James, Jeffrey Heinz, Gil Bailey, Matt Edlefsen, Molly Visscher, David Wellcome & Sean Wibel (2010). On languages piecewise testable in the strict sense. *The Mathematics of Language* 10/11, 255–265.

Rogers, James, Jeffrey Heinz, Margaret Fero, Jeremy Hurst, Dakotah Lambert & Sean Wibel (2013). Cognitive and sub-regular complexity. Morrill, Glyn & Mark-Jan Nederhof (eds.), *Formal Grammar: 17th and 18th International Conferences*, Springer Berlin Heidelberg, Berlin, Heidelberg, 90–108.

Takahashi, Chikako (2018). No metathesis in Harmonic Serialism. Gallagher, Gillian, Maria Gouskova & Sora Yin (eds.), *Supplemental Proceedings of the 2017 Annual Meeting on Phonology*, Linguistic Society of America, Washington, DC.

Proceedings of the 36th West Coast Conference on Formal Linguistics

edited by Richard Stockwell, Maura O'Leary, Zhongshi Xu, and Z.L. Zhou

Cascadilla Proceedings Project   Somerville, MA   2019

Copyright information

ISBN 978-1-57473-474-4 hardback

Ordering information

Orders for the printed 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

This entire proceedings can also be viewed on the web at www.lingref.com. Each paper has a unique document # which can be added to citations to facilitate access. The document # should not replace the full citation.

This paper can be cited as:

Lamont, Andrew. 2019. Precedence is Pathological: The Problem of Alphabetical Sorting. In *Proceedings of the 36th West Coast Conference on Formal Linguistics*, ed. Richard Stockwell et al., 243-249. Somerville, MA: Cascadilla Proceedings Project. www.lingref.com, document #3468.