Computational Restrictions on Iterative Prosodic Processes

1. Contribution. Here we describe computational restrictions on the computation of iterative phonological processes involving stress, epenthesis, and syllabification. We show that while these processes are fundamentally local, fitting the typology of other computational work, they require additional computational resources. We formulate this description via logical transductions (Courcelle, 1997), and show that using Least Fixed Point operators allows the iterativity to proceed without quantification, a necessary and sufficient restriction characteristic of the computational landscape of local functions (Chandlee and Jardine, 2019).

2. Locality of syllabification. Strother-Garcia (2019) demonstrates that syllabification processes are fundamentally local (in a strict mathematical sense) in Moroccan Arabic (MA) and Imdlawn Tashlhiyt Berber (ITB). To identify syllabic nuclei, for example, it suffices to compare each segment to its predecessor and to the 2 segments that follow it—a 'window' of size 4. Crucially, the logical formulas used for syllabification in ITB and MA lack *quantification* (via $\exists$ or $\forall$) and are said to be *quantifier-free* (QF). They depend only on local information in the input string, corresponding to *input strictly local* functions (ISL; Chandlee, 2014). This contrasts with constraint-interaction accounts (e.g. Prince and Smolensky, 1993) where the entire word must be considered (*global* evaluation), obfuscating the fact that the process is local.

3. QFLFP. While ISL functions suffice for many phonological processes, in some cases computation depends on information in the *output* string. This includes iterative processes (**?**Chandlee et al., 2015). Consider the mapping in (1):

(1) *baaa* $\mapsto$ *bbbb*

All *a*'s following a *b* are outputted as a *b*. Such iterative spreading is not ISL because the trigger for assimilation can be separated from a target by a potentially unbounded number of input elements. Any attempt at a QF definition for this mapping fails because it cannot identify all input positions that could potentially be output as *b* when the process is unbounded. However, the mapping is local in the *output* – an input symbol is rewritten as *b* when it is immediately preceded by a *b* in the output string. Such iterative output-oriented processes can be described by extending QF logic with *least fixed point* operators (LFP; Libkin, 2004). QFLFP logic enables us to write simple recursive definitions of predicates. Rather than present the full formalism, we focus on how these recursive formulas capture a range of phonological patterns while preserving a notion of locality in the output. Thus, we use *implicit definitions* of predicates (Rogers, 1997), whereby a predicate recursively refers to itself. A definition for (1) is as follows:

(2) $b'(x) \stackrel{\text{d}}{=} b(x) \vee b'(p(x))$

Given an input element x, it is mapped to a b in the output when it is a b in the input or it is preceded by a b in the output. Using (1) as an example, the first a will follow a b in the output, and so it is mapped to b. This means that the second a now follows a b, and so it is also mapped to b, and so on. The definition in (2) applies recursively to all elements following a b, but does so in a way that is local to the output – the transduction only need look one position to the left to determine output b labels.

4. QFLFP and Phonological Iterativity Prosodic processes tend to be phonologically iterative. We go through some case studies and show that they are QFLFP.

4.1 Iterative stress: QFLFP logic allows for intuitive definitions of iterative stress assignment. For example, Murinbata (Street and Mollinjin, 1981) applies stress to every other syllable beginning with the initial syllable. It is described by the following transduction:

(3) $\acute{\sigma}(x) \stackrel{\text{d}}{=} first(x) \vee \acute{\sigma}(p(p(x)))$ input-output map: $\sigma\sigma\sigma\sigma\sigma\sigma\sigma \mapsto \acute{\sigma}\sigma\acute{\sigma}\sigma\acute{\sigma}\sigma\acute{\sigma}$

The first element in a string receives stress. Further stresses are placed on those elements that have a stressed syllable two to the left in the output string. This applies recursively.

4.2 Iterative syllabification and epenthesis: A classic case for iteration comes from how different Arabic dialects choose different epenthesis sites for consonant clusters. In a 3-C cluster, a vowel is inserted after C_1 in Iraqi, and C_2 in Cairene. In a 4-C cluster, both dialects insert a vowel after C_2. Itô (1989) analyzes these facts as via directional syllabification. Iraqi syllabifies right-to-left, while Cairene left-to-right. A vowel is added based on a CVC template.

Iraqi	<katab-t-l-u>	<katabt>.lu.	<kata>.bit.lu.	<ka>.ta.bit.lu..	.ka.ta.bit.lu.
(R-to-L)	<katab-t-l-ha>	<katabtl>.ha.	<katab>til.ha.	<ka>tab.til.ha.	.ka.tab.til.ha.
Cairene	<katab-t-l-u>	.kat.<abtlu>	.ka.tab.<tlu>	.ka.tab.til.<u>	.ka.tab.ti.lu
(L-to-R)	<katab-t-l-ha>	.kat.<abtlha>	.ka.tab.<tlha>	.ka.tab.til.<ha>	.ka.tab.til.ha

Computationally, iterative epenthesis is QFLFP. Below, we provide QFLFP transductions for Arabic. The functions $L'(x)$ and $R'(x)$ determine what should be the left- and right-edges of syllables before resyllabification. Resyllabification is only apparent in left-to-right parsing.

R-to-L	$L'(x) \overset{\text{d}}{=}$	$[C(x) \wedge V(s(x))] \vee$	*select a C in $\underline{C}V$ context*
		$[C(x) \wedge C(s(x)) \wedge L(s(s(x)))]$	*select a C in $\underline{C}C[_L$ context*
	$i'(x_2) \overset{\text{d}}{=}$	$C(x) \wedge L(x) \wedge C(s(x))$	*add V in $[_L C\_\ C$*
L-to-R	$R'(x) \overset{\text{d}}{=}$	$[C(x) \wedge V(p(x))] \vee$	*select a C in $V\underline{C}$ context*
		$[C(x) \wedge C(p(x)) \wedge R(p(p(x)))]$	*select a C in $]_R C\underline{C}$ context*
	$i'(x_2) \overset{\text{d}}{=}$	$C(x) \wedge C(s(x)) \wedge R(s(x))$	*add V in $C\_\ C]_R$*

$s(x)$ returns the successor of x, while $p(x)$ returns the predecessor. We illustrate with right-to-left and left-to-right syllabification for /katab-t-l-u/. The former yields the Iraqi form [ka.ta.bit.lu] while the latter yields the Cairene form [ka.tab.ti.lu].

Input	k	a	t	a	b	t	l	u		k	a	t	a	b	t	l	u
L' is true at...									R' is true at...								
Iteration 0	✓		✓				✓					✓		✓			
Iteration 1	✓		✓		✓		✓					✓		✓		✓	
Interim Output:	k_L	a	t_L	a	b_L	t	l_L	u		k	a	t_R	a	b_R	t	l_R	u
$i'(x_2)$					✓										✓		
Output:	k	a	t	a	b i	t	l	u		k	a	t	a	b	t i	l	u

Conclusion. QFLFP provides a necessary, but sufficiently limited extension to previous methods in computational phonology. We speculate that other prosodic processes at different scales of representation also possess such restrictions. Moreover, the fact that iterative processes are describable with QFLFP reinforces the view that locality lies at the heart of phonological processes.

References: **[1]** Chandlee, J. (2014). Strictly Local Phonological Processes. Ph. D. thesis, University of Delaware, Newark, DE. **[2]** Chandlee, J., R. Eyraud, and J. Heinz (2015). Output strictly local functions. In 14th Meeting on the Mathematics of Language, pp. 112–125. **[3]** Chandlee, J. and A. Jardine (2019, July). Quantifier-free least fixed point functions for phonology. In Proceedings of the 16th Meeting on the Mathematics of Language, Toronto, Canada, pp. 50–62. Association for Computational Linguistics. **[4]** Courcelle, B. (1997). The expression of graph properties and graph transformations in monadic second-order logic. In G. Rozenberg (Ed.), Handbook of Graph Grammars and Computing by Graph Transformations, pp. 313–400. World Scientific. **[5]** Itô, J. (1989). A prosodic theory of epenthesis. Natural Language & Linguistic Theory 7(2), 217–259. **[6]** Libkin, L. (2004). Elements of Finite Model Theory. Berlin: Springer-Verlag. **[7]** Prince, A. and P. Smolensky (1993). Optimality Theory: Constraint interaction in Generative Grammar. Malden, Mass: Blackwell. **[8]** Rogers, J. (1997). Strict lt2 : Regular :: Local : Recognizable. In C. Retoré (Ed.), Logical Aspects of Computational Linguistics: First International Conference, LACL '96 Nancy, France, September 23–25, 1996 Selected Papers, pp. 366–385. Berlin, Heidelberg: Springer Berlin Heidelberg. **[9]** Street, C. S. and G. P. Mollinjin (1981). The phonology of murinbata. Australian phonologies: Collected papers, 183–244. **[10]** Strother-Garcia, K. (2019). Using Model Theory in Phonology: A Novel Characterization of Syllable Structure and Syllabification. Ph. D. thesis, University of Delaware.