Yale

Rhythmic Syncope and Strict Locality in Subregular Phonology

Dustin Bowers and Yiding Hao

UNIVERSITY OF ARIZONA · YALE UNIVERSITY

SYNC · STONY BROOK, NY

December 1, 2018

Introduction

Rhythmic Syncope

Rhythmic syncope is the deletion of every second vowel in a word.

Rhythmic Syncope

Rhythmic syncope is the deletion of every second vowel in a word.

(1) 1930s Ojibwe (Bowers, To appear)

a. [gtɪgmɪngɪbnɑː-d] "If he rolls him"

b. [n-gʊtgʊmnʌgbɪnɑː] "I roll him"

Rhythmic Syncope

Rhythmic syncope is the deletion of every second vowel in a word.

(1) 1930s Ojibwe (Bowers, To appear)

a. [gtɪgmɪngɪbnɑː-d] "If he rolls him"

b. [n-gʊtgʊmnʌgbɪnɑː] "I roll him"

Rhythmic Syncope

Rhythmic syncope is the deletion of every second vowel in a word.

(1) 1930s Ojibwe (Bowers, To appear)

a. /gʊtɪgʊmɪnʌgɪbɪnɑː-d/ "If he rolls him"

b. /ni-gʊtɪgʊmɪnʌgɪbɪnɑː/ "I roll him"

Rhythmic Syncope

Definition

Assume an alphabet $\Sigma = C \cup V$, with $C \cap V = \varnothing$. The *rhythmic syncope function* is the function defined by

$$\rho(c_0 v_1 c_1 v_2 c_2 \ldots v_n c_n) = c_0 v_1 c_1 c_2 v_3 c_3 c_4 \ldots c_n$$

where for each i, $c_i \in C^*$ and $v_i \in V$.

Yale

Rhythmic Syncope

Definition

Assume an alphabet $\Sigma = C \cup V$, with $C \cap V = \varnothing$. The *rhythmic syncope function* is the function defined by

$$\rho(c_0 v_1 c_1 v_2 c_2 \ldots v_n c_n) = c_0 v_1 c_1 c_2 v_3 c_3 c_4 \ldots c_n$$

where for each i, $c_i \in C^*$ and $v_i \in V$.

$$\rho(CVCVCVCVCVC) = CVCCVCCVC$$

Outline

- Show that ρ is not *strictly local*.

Yale

Outline

- Show that ρ is not *strictly local*.
- Define a class of functions that includes ρ.

Outline

- Show that ρ is not *strictly local*.
- Define a class of functions that includes ρ.
- Discuss the theoretical consequences.

Strictly Local Functions

What constitutes a possible phonological dependency?

Subregular Phonology

What constitutes a possible phonological dependency?

- *Tier-based strictly local* dependencies (Heinz et al., 2011).

Yale

Subregular Phonology

What constitutes a possible phonological dependency?

- *Tier-based strictly local* dependencies (Heinz et al., 2011).
- *NC̥: *np, *nt, *nk, *mp, *mt, *mk...

Subregular Phonology

What constitutes a possible phonological dependency?

- *Tier-based strictly local* dependencies (Heinz et al., 2011).
- *NC̥: *np, *nt, *nk, *mp, *mt, *mk...
- *CC: *pp, *pt, *pk, *bp, *bt, *bk...

Yale

Subregular Phonology

What constitutes a possible phonological dependency?

- *Tier-based strictly local* dependencies (Heinz et al., 2011).
- *NC̥: *np, *nt, *nk, *mp, *mt, *mk...
- *CC: *pp, *pt, *pk, *bp, *bt, *bk...
- NoCoda: *p#, *t#, *k#, *b#, *d#, *g#...

Subregular Phonology

What constitutes a possible phonological dependency?

- *Tier-based strictly local* dependencies (Heinz et al., 2011).
- *NC̥: *np, *nt, *nk, *mp, *mt, *mk...
- *CC: *pp, *pt, *pk, *bp, *bt, *bk...
- NoCoda: *p#, *t#, *k#, *b#, *d#, *g#...
- Vowel Harmony: *ea, *eı, *eu, *ae, *ai, *aü...

Yale

Tier-Based Strictly Local Languages

A language L is *k-strictly local on tier T* (TSL) if

Tier-Based Strictly Local Languages

A language L is *k-strictly local on tier T* (TSL) if

- there exists $S \subseteq \Sigma^*$ such that

Yale

Tier-Based Strictly Local Languages

A language L is *k-strictly local on tier* T (TSL) if

- there exists $S \subseteq \Sigma^*$ such that
- every string in S has length at most k and

Yale

Tier-Based Strictly Local Languages

A language L is *k-strictly local on tier T* (TSL) if

- there exists $S \subseteq \Sigma^*$ such that
- every string in S has length at most k and
- $x \in L$ if and only if no element of S is a substring of x, ignoring symbols not in T.

Tier-Based Strictly Local Functions

- TSL languages describe *phonotactic dependencies*.

Tier-Based Strictly Local Functions

- TSL languages describe *phonotactic dependencies*.
- Phonological processes are described by *TSL functions*.

Tier-Based Strictly Local Functions

- TSL languages describe *phonotactic dependencies*.
- Phonological processes are described by *TSL functions*.
- Example: $\varnothing \rightarrow \mathrm{u} \ / \ C \ \_ \begin{Bmatrix} C \\ \# \end{Bmatrix}$

Yale

Tier-Based Strictly Local Functions

- TSL languages describe *phonotactic dependencies*.
- Phonological processes are described by *TSL functions*.
- Example: $\varnothing \rightarrow \mathrm{u} \;/\; C \,\_\, \begin{Bmatrix} C \\ \# \end{Bmatrix}$

Japanese /aiskriim/ → [aisukuriimu]

Output:		a								
Input:	#	a	i	s	k	r	i	i	m	#
		↑								

Yale

Tier-Based Strictly Local Functions

- TSL languages describe *phonotactic dependencies*.
- Phonological processes are described by *TSL functions*.
- Example: $\varnothing \rightarrow \mathrm{u} \;/\; C \,\_\, \begin{Bmatrix} C \\ \# \end{Bmatrix}$

Japanese $/\mathrm{aiskriim}/ \rightarrow [\mathrm{aisukuriimu}]$

Output:		a	i							
Input:	#	a	i	s	k	r	i	i	m	#
			↑							

Yale

Tier-Based Strictly Local Functions

- TSL languages describe *phonotactic dependencies*.
- Phonological processes are described by *TSL functions*.
- Example: $\varnothing \to \mathrm{u} \;/\; C\;\_\;\begin{Bmatrix} C \\ \# \end{Bmatrix}$

Japanese /aiskriim/ $\to$ [aisukuriimu]

Output:		a	i	s						
Input:	#	a	i	s	k	r	i	i	m	#
				↑						

Tier-Based Strictly Local Functions

- TSL languages describe *phonotactic dependencies*.
- Phonological processes are described by *TSL functions*.
- Example: $\varnothing \rightarrow \mathrm{u} \;/\; C \;\_\; \begin{Bmatrix} C \\ \# \end{Bmatrix}$

Japanese $/\mathrm{aiskriim}/ \rightarrow [\mathrm{aisukuriimu}]$

Output:		a	i	s	u	k					
Input:	#	a	i	s		k	r	i	i	m	#
						↑					

Yale

Tier-Based Strictly Local Functions

- TSL languages describe *phonotactic dependencies*.
- Phonological processes are described by *TSL functions*.
- Example: $\varnothing \rightarrow \mathrm{u} \,/\, C \,\_ \begin{Bmatrix} C \\ \# \end{Bmatrix}$

Japanese $/\mathrm{aiskriim}/ \rightarrow [\mathrm{aisukuriimu}]$

Output:		a	i	s	u	k	u	r				
Input:	#	a	i	s		k		r	i	i	m	#
								↑				

Tier-Based Strictly Local Functions

- TSL languages describe *phonotactic dependencies*.
- Phonological processes are described by *TSL functions*.
- Example: $\varnothing \rightarrow \mathrm{u} \,/\, C \_ \begin{Bmatrix} C \\ \# \end{Bmatrix}$

Japanese $/\mathrm{aiskriim}/ \rightarrow [\mathrm{aisukuriimu}]$

Output:		a	i	s	u	k	u	r	i			
Input:	#	a	i	s		k		r	i	i	m	#
									↑			

Yale

Tier-Based Strictly Local Functions

- TSL languages describe *phonotactic dependencies*.
- Phonological processes are described by *TSL functions*.
- Example: $\varnothing \rightarrow \text{u} \;/\; C \_ \begin{Bmatrix} C \\ \# \end{Bmatrix}$

Japanese $/\text{aiskriim}/ \rightarrow [\text{aisukuriimu}]$

Output:		a	i	s	u	k	u	r	i	i		
Input:	#	a	i	s		k		r	i	i	m	#
										↑		

Yale

Tier-Based Strictly Local Functions

- TSL languages describe *phonotactic dependencies*.
- Phonological processes are described by *TSL functions*.
- Example: $\varnothing \rightarrow \mathrm{u}\ /\ C\ \_ \begin{Bmatrix} C \\ \# \end{Bmatrix}$

Japanese $/\mathrm{aiskriim}/ \rightarrow [\mathrm{aisukuriimu}]$

Output:		a	i	s	u	k	u	r	i	i	m	
Input:	#	a	i	s		k		r	i	i	m	#
											↑	

Yale

Tier-Based Strictly Local Functions

- TSL languages describe *phonotactic dependencies*.
- Phonological processes are described by *TSL functions*.
- Example: $\varnothing \rightarrow \mathrm{u} \;/\; C \_\_ \begin{Bmatrix} C \\ \# \end{Bmatrix}$

Japanese /aiskriim/ → [aisukuriimu]

Output:		a	i	s	u	k	u	r	i	i	m	u	
Input:	#	a	i	s		k		r	i	i	m		#
													↑

Tier-Based Strictly Local Functions

We make the following assumptions about phonological processes.

Tier-Based Strictly Local Functions

We make the following assumptions about phonological processes.

- Streaming: The process scans the input string from left to right and produces part of the output at each time step.

Tier-Based Strictly Local Functions

We make the following assumptions about phonological processes.

- Streaming: The process scans the input string from left to right and produces part of the output at each time step.
- Determinism: If the process f outputs y after reading x, then y is a *common prefix* of $\{f(xz) | z \in \Sigma^*\}$.

Tier-Based Strictly Local Functions

We make the following assumptions about phonological processes.

- Streaming: The process scans the input string from left to right and produces part of the output at each time step.
- Determinism: If the process f outputs y after reading x, then y is a *common prefix* of $\{f(xz) | z \in \Sigma^*\}$.

Definition

A *common prefix* of a set A is a string p such that every string in A begins with p.

Tier-Based Strictly Local Functions

We make the following assumptions about phonological processes.

- Streaming: The process scans the input string from left to right and produces part of the output at each time step.
- Determinism: If the process f outputs y after reading x, then y is a *common prefix* of $\{f(xz) | z \in \Sigma^*\}$.

Definition

A *common prefix* of a set A is a string p such that every string in A begins with p.

- Parsimony: This common prefix is the longest one possible.

Tier-Based Strictly Local Functions

Definition

For a function $f : \Sigma^* \to \Sigma^*$,

$$f^{\leftarrow}(x) = \text{lcp}(\{f(xz) | z \in \Sigma^*\})$$

and $f^{\rightarrow}(z, xz)$ is the string such that

$$f(xz) = f^{\leftarrow}(x) f^{\rightarrow}(z, xz).$$

Tier-Based Strictly Local Functions

Definition

For a function $f : \Sigma^* \to \Sigma^*$,

$$f^{\leftarrow}(x) = \text{lcp}(\{f(xz) | z \in \Sigma^*\})$$

and $f^{\rightarrow}(z, xz)$ is the string such that

$$f(xz) = f^{\leftarrow}(x) f^{\rightarrow}(z, xz).$$

Definition

For a string $x \in \Sigma^*$, tier $T \subseteq \Sigma$, and number k, $\text{suff}^k_T(x)$ is the last k symbols of x on tier T.

Tier-Based Strictly Local Functions

Definition (Chandlee et al., In prep)

A function $f : \Sigma^* \to \Sigma^*$ is *k-strictly local on tier T* if for every $u, v \in \Sigma^*$, if

$$\text{suff}_T^{k-1}(u) = \text{suff}_T^{k-1}(v)$$

and

$$\text{suff}_T^{k-1}(f^{\leftarrow}(u)) = \text{suff}_T^{k-1}(f^{\leftarrow}(v)),$$

then for all $w \in \Sigma^*$ we have

$$f^{\rightarrow}(w, uw) = f^{\rightarrow}(w, vw).$$

ρ is not TSL

To show that ρ is *not* k-SL on tier T, we must find u, v, w such that

$$\mathsf{suff}_T^{k-1}(u) = \mathsf{suff}_T^{k-1}(v)$$
$$\mathsf{suff}_T^{k-1}(\rho^{\leftarrow}(u)) = \mathsf{suff}_T^{k-1}(\rho^{\leftarrow}(v)),$$

but $\rho^{\rightarrow}(w, uw) \neq \rho^{\rightarrow}(w, vw)$.

ρ is not TSL

To show that ρ is *not* k-SL on tier T, we must find u, v, w such that

$$\mathrm{suff}_T^{k-1}(u) = \mathrm{suff}_T^{k-1}(v)$$
$$\mathrm{suff}_T^{k-1}(\rho^{\leftarrow}(u)) = \mathrm{suff}_T^{k-1}(\rho^{\leftarrow}(v)),$$

but $\rho^{\rightarrow}(w, uw) \neq \rho^{\rightarrow}(w, vw)$.

- Let a be a vowel. $u = a^{4k}$, $v = a^{4k+1}$, $w = a$

Yale

ρ is not TSL

To show that ρ is *not* k-SL on tier T, we must find u, v, w such that

$$\mathsf{suff}_T^{k-1}(u) = \mathsf{suff}_T^{k-1}(v)$$
$$\mathsf{suff}_T^{k-1}(\rho^{\leftarrow}(u)) = \mathsf{suff}_T^{k-1}(\rho^{\leftarrow}(v)),$$

but $\rho^{\rightarrow}(w, uw) \neq \rho^{\rightarrow}(w, vw)$.

- Let a be a vowel. $u = a^{4k}$, $v = a^{4k+1}$, $w = a$
- $\rho^{\leftarrow}(u) = a^{2k}$, $\rho^{\leftarrow}(v) = a^{2k+1}$

Yale

ρ is not TSL

To show that ρ is *not* k-SL on tier T, we must find u, v, w such that

$$\mathsf{suff}_T^{k-1}(u) = \mathsf{suff}_T^{k-1}(v)$$
$$\mathsf{suff}_T^{k-1}(\rho^{\leftarrow}(u)) = \mathsf{suff}_T^{k-1}(\rho^{\leftarrow}(v)),$$

but $\rho^{\rightarrow}(w, uw) \neq \rho^{\rightarrow}(w, vw)$.

- Let a be a vowel. $u = a^{4k}$, $v = a^{4k+1}$, $w = a$
- $\rho^{\leftarrow}(u) = a^{2k}$, $\rho^{\leftarrow}(v) = a^{2k+1}$
- $\rho(uw) = a^{2k+1}$, $\rho(vw) = a^{2k+1}$

Yale

ρ is not TSL

To show that ρ is *not* k-SL on tier T, we must find u, v, w such that

$$\mathsf{suff}_T^{k-1}(u) = \mathsf{suff}_T^{k-1}(v)$$
$$\mathsf{suff}_T^{k-1}(\rho^{\leftarrow}(u)) = \mathsf{suff}_T^{k-1}(\rho^{\leftarrow}(v)),$$

but $\rho^{\rightarrow}(w, uw) \neq \rho^{\rightarrow}(w, vw)$.

- Let a be a vowel. $u = a^{4k}$, $v = a^{4k+1}$, $w = a$
- $\rho^{\leftarrow}(u) = a^{2k}$, $\rho^{\leftarrow}(v) = a^{2k+1}$
- $\rho(uw) = a^{2k+1}$, $\rho(vw) = a^{2k+1}$
- $\rho^{\rightarrow}(w, uw) = a$, $\rho^{\rightarrow}(w, vw) = \varnothing$

Time Alignment

Why is ρ not TSL?

Parsimonious deterministic streaming algorithm for ρ:

Why is ρ not TSL?

Parsimonious deterministic streaming algorithm for ρ:

- Do not delete the first vowel.

Yale

Why is ρ not TSL?

Parsimonious deterministic streaming algorithm for ρ:

- Do not delete the first vowel.
- If the most recent vowel was deleted, do not delete this vowel.

Why is ρ not TSL?

Parsimonious deterministic streaming algorithm for ρ:

- Do not delete the first vowel.
- If the most recent vowel was deleted, do not delete this vowel.
- If the most recent vowel was not deleted, delete this vowel.

Yale

Why is ρ not TSL?

Parsimonious deterministic streaming algorithm for ρ:

- Do not delete the first vowel.
- If the most recent vowel was deleted, do not delete this vowel.
- If the most recent vowel was not deleted, delete this vowel.

Deletion destroys evidence.

Rhythmic Syncope in OT

Existing OT analyses address this problem.

Rhythmic Syncope in OT

Existing OT analyses address this problem.

- Kager (1997): $V \rightarrow \bullet$ instead of $V \rightarrow \varnothing$

Yale

Rhythmic Syncope in OT

Existing OT analyses address this problem.

- Kager (1997): $V \rightarrow \bullet$ instead of $V \rightarrow \varnothing$
- Blumenfeld (2006): IO-markedness constraints

Rhythmic Syncope in OT

Existing OT analyses address this problem.

- Kager (1997): $V \rightarrow \bullet$ instead of $V \rightarrow \varnothing$
- Blumenfeld (2006): IO-markedness constraints
- McCarthy (2008): Harmonic Serialism

Rhythmic Syncope in OT

Existing OT analyses address this problem.

- Kager (1997): $V \rightarrow \bullet$ instead of $V \rightarrow \varnothing$
- Blumenfeld (2006): IO-markedness constraints
- McCarthy (2008): Harmonic Serialism
  - CVCVCVCV

Rhythmic Syncope in OT

Existing OT analyses address this problem.

- Kager (1997): $V \to \bullet$ instead of $V \to \varnothing$
- Blumenfeld (2006): IO-markedness constraints
- McCarthy (2008): Harmonic Serialism
  - CVCVCVCV
  - (CV.CV)(CV.CV)

Rhythmic Syncope in OT

Existing OT analyses address this problem.

- Kager (1997): $V \to \bullet$ instead of $V \to \varnothing$
- Blumenfeld (2006): IO-markedness constraints
- McCarthy (2008): Harmonic Serialism
  - $\mathrm{CVCVCVCV}$
  - $\mathrm{(CV.CV)(CV.CV)}$
  - $\mathrm{(C\acute{V}.CV)(C\acute{V}.CV)}$

Yale

Rhythmic Syncope in OT

Existing OT analyses address this problem.

- Kager (1997): $V \rightarrow \bullet$ instead of $V \rightarrow \varnothing$
- Blumenfeld (2006): IO-markedness constraints
- McCarthy (2008): Harmonic Serialism
  - CVCVCVCV
  - (CV.CV)(CV.CV)
  - (CV́.CV)(CV́.CV)
  - (CV́.C)(CV́.C)

Yale

Time-Aligned TSL Functions

Definition

Let $f : \Sigma^* \to \Sigma^*$ and $x = x_1 x_2 \ldots x_n \in \Sigma^*$. The *ith most recent action of f on x* is the pair $\langle x_{n-i+1}, f_i^{\leftarrow}(x) \rangle$, where $f_i^{\leftarrow}(x)$ is the string such that

$$f^{\leftarrow}(x_1 x_2 \ldots x_{n-i+1}) = f^{\leftarrow}(x_1 x_2 \ldots x_{n-i}) f_i^{\leftarrow}(x).$$

For $T \subseteq \Sigma^*$, the *ith most recent action of f on x on tier T* is the action denoted

$$\langle x_{i,T}, f_{i,T}^{\leftarrow}(x) \rangle := \left\langle x_{n-j+1}, f_{n-j+1}^{\leftarrow}(x) \right\rangle,$$

where j is the ith largest index such that $x_{n-j+1} \in T$ and $f_{n-j+1}^{\leftarrow}(x) \in T^*$.

Yale

Time-Aligned TSL Functions

Definition

Let $f : \Sigma^* \to \Sigma^*$ and $T \subseteq \Sigma$. For $k \in \mathbb{N}$, f is *time-aligned k-strictly local on tier T* if for all $u, v \in \Sigma$, if

$$\left\langle u_{i,T}, f^{\leftarrow}_{i,T}(u) \right\rangle = \left\langle v_{i,T}, f^{\leftarrow}_{i,T}(V) \right\rangle$$

for $1 \leq i \leq k$, then for all $w \in \Sigma^*$,

$$f^{\rightarrow}(w, uw) = f^{\rightarrow}(w, vw).$$

Yale

ρ is TATSL

ρ is 1-TATSL on tier *V*.

ρ is TATSL

ρ is 1-TATSL on tier V.

- $\rho^{\leftarrow}_{1,V}(c_0 v_1 c_1 v_2 c_2 \ldots v_{2m} c_{2m}) = \varnothing$

Yale

ρ is TATSL

ρ is 1-TATSL on tier *V*.

- $\rho^{\leftarrow}_{1,V}(c_0 v_1 c_1 v_2 c_2 \ldots v_{2m} c_{2m}) = \varnothing$
- $\rho^{\leftarrow}_{1,V}(c_0 v_1 c_1 v_2 c_2 \ldots v_{2m+1} c_{2m+1}) = v_{2m+1}$

Yale

Conclusion

Theoretical Consequences

- Chandlee's (2014) goal is to describe *all* phonological processes.

Theoretical Consequences

- Chandlee's (2014) goal is to describe *all* phonological processes.
- This talk suggests that TSL is not the right answer.

Theoretical Consequences

- Chandlee's (2014) goal is to describe *all* phonological processes.
- This talk suggests that TSL is not the right answer.
- However, Bowers (To appear) suggests that rhythmic syncope may not be diachronically robust.

Theoretical Consequences

- Chandlee's (2014) goal is to describe *all* phonological processes.
- This talk suggests that TSL is not the right answer.
- However, Bowers (To appear) suggests that rhythmic syncope may not be diachronically robust.
  - 1930s: /gʊtɪgʊmɪnʌgɪbɪnɑː-d/ → [gtɪgmɪngɪbnɑː-d]

Yale

Theoretical Consequences

- Chandlee's (2014) goal is to describe *all* phonological processes.
- This talk suggests that TSL is not the right answer.
- However, Bowers (To appear) suggests that rhythmic syncope may not be diachronically robust.
  - 1930s: /gʊtɪgʊmɪnʌgɪbɪnɑː-d/ → [gtɪgmɪngɪbnɑː-d]
  - Today: /gtɪgmɪŋɪbnɑː-d/ → [gtɪgmɪŋɪbnɑː-d]

Yale

Theoretical Consequences

- Chandlee's (2014) goal is to describe *all* phonological processes.
- This talk suggests that TSL is not the right answer.
- However, Bowers (To appear) suggests that rhythmic syncope may not be diachronically robust.
  - 1930s: /gʊtɪgʊmɪnʌgɪbɪnaː-d/ → [gtɪgmɪngɪbnaː-d]
  - Today: /gtɪgmɪŋɪbnaː-d/ → [gtɪgmɪŋɪbnaː-d]
- Modern speakers understand the 1930s forms but do not use them.

Theoretical Consequences

- Chandlee's (2014) goal is to describe *all* phonological processes.
- This talk suggests that TSL is not the right answer.
- However, Bowers (To appear) suggests that rhythmic syncope may not be diachronically robust.
  - 1930s: /gʊtɪgʊmɪnʌgɪbɪnɑː-d/ → [gtɪgmɪngɪbnɑː-d]
  - Today: /gtɪgmɪŋɪbnɑː-d/ → [gtɪgmɪŋɪbnɑː-d]
- Modern speakers understand the 1930s forms but do not use them.
- Similar phenomena have been observed in Old Russian (Isacenko, 1970) and Old Irish (McManus, 1983).

Conclusion

- Rhythmic syncope is not TSL.

Conclusion

- Rhythmic syncope is not TSL.
- The time-aligned TSL functions incorporate rhythmic syncope.

Conclusion

- Rhythmic syncope is not TSL.
- The time-aligned TSL functions incorporate rhythmic syncope.
- Rejection of rhythmic syncope by child learners would constitute evidence for the TSL hypothesis.

References

Blumenfeld, Lev A. 2006. Constraints on Phonological Interactions. Stanford, CA: Stanford University PhD Dissertation.

Bowers, Dustin. To appear. The Nishnaabemwin Restructuring Controversy: New Empirical Evidence. Phonology .

Chandlee, Jane. 2014. Strictly Local Phonological Processes. Newark, DE: University of Delaware PhD Dissertation.

Chandlee, Jane, Rémi Eyraud & Jeffrey Heinz. In prep. Input–output strictly local functions and their efficient learnability.

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: Human Language Technologies, 58–64. Portland, OR: Association for Computational Linguistics.

Isacenko, Alexander. 1970. East Slavic morphophonemics and the treatment of the jers in Russian: A revision of Havlík's law. International Journal of Slavic Linguistics and Poetics 13. 73–124.

Kager, René. 1997. Rhythmic vowel deletion in Optimality Theory. In Iggy Roca (ed.), Derivations and Constraints in Phonology, 463–499. Oxford, United Kingdom: Clarendon Press.

McCarthy, John J. 2008. The serial interaction of stress and syncope. Natural Language & Linguistic Theory 26(3). 499–546. doi:10.1007/s11049-008-9051-3.

McManus, Damian. 1983. A Chronology of the Latin Loan-Words in Early Irish. Ériu 34. 21–71.