Rule Application Modes Revisited

Jane Chandlee

Haverford College

17 September 2020
Johns Hopkins University Colloquium Series

Rule application modes

- An SPE-style rule is ambiguous w.r.t. how it is applied to a string:

$[-\text{cons}] \rightarrow [+\text{nasal}] \;/\; \_\_ \; [+\text{nasal}]$

(1) $\text{V}^n\text{N} \mapsto ?$

a. $\tilde{\text{V}}^n\text{N}$

b. $\text{V}^{n-1}\tilde{\text{V}}\text{N}$

Rule application modes

- An SPE-style rule is ambiguous w.r.t. how it is applied to a string:

$$[-\text{cons}] \rightarrow [+\text{nasal}] \ / \ \_\_ \ [+\text{nasal}]$$

(1) $V^nN \mapsto$?

a. $\tilde{V}^nN$ iterative
b. $V^{n-1}\tilde{V}N$ simultaneous

Rule application modes

- An SPE-style rule is ambiguous w.r.t. how it is applied to a string:

$[-\text{cons}] \rightarrow [+\text{nasal}] \ / \ \_\_ \ [+\text{nasal}]$

(1) $\text{V}^n\text{N} \mapsto ?$

a. /ujĩ/ ↦ [ũj̃ĩ] iterative
'praise (n.)' (Mọ́bà Yoruba; Ajíbóyè, 2001)

b. /jɔn/ ↦ [jɔ̃n] simultaneous
'yawn' (English)

Why are we talking about rules at all?

- A rule is an *intensional* description of its *extension*, a map:

  [−cons] → [+nasal] / __ [+nasal]

  iterative:
  { (VN, ṼN), (VV, VV), (NVN, NṼN), (VVVN, ṼṼṼN) ... }

  simultaneous:
  { (VN, ṼN), (VV, VV), (NVN, NṼN), (VVVN, VVṼN) ... }
- The question of how to characterize the difference between these two maps remains, even if we're not using rules.

Simultaneous versus iterative

- Following Chomsky and Halle (1968), various proposals were put forth to replace simultaneous application with iteration (Anderson, 1969, 1974; Johnson, 1972; Howard, 1972; Lightner, 1972; Kenstowicz and Kisseberth, 1979).
- The advent of Optimality Theory (Prince and Smolensky, 1993, 2004), which is inherently iterative, further called into question whether phonology is ever 'noniterative' (e.g. Kaplan, 2008).
- The debate is not whether 'simultaneous' extensions exist, but whether noniterativity is the right way to characterization them.

In terms of computation...

- A computational approach to phonology works with the extension (or map) directly and identifies its computational properties.
- The simultaneous versus iterative distinction becomes one of input- versus output-based computation (Kaplan and Kay, 1994; Chandlee, 2014).
- When dealing with intensions, we can do away with simultaneous application by changing the form of the rule or adding constraints.
- We have less flexibility when dealing with extensions: a map either has a given computational property, or it doesn't.

Big question

- If we don't need simultaneous application, does that mean we don't need input-based computation?

Outline

1. Simultaneous versus iterative application in rule- and constraint-based phonology.
2. Computational background: ISL and OSL
3. Analyses: Do we need both?
4. Open questions/Conclusions

Simultaneous application

(2) ‘To apply a rule, the entire string is first scanned for segments that satisfy the environmental constraints of the rule. After all such segments have been identified in the string, the changes required by the rule are applied simultaneously’ (Chomsky and Halle, 1968, pg. 344).

Simultaneous application

(2) 'To apply a rule, the entire string is first scanned for segments that satisfy the environmental constraints of the rule. After all such segments have been identified in the string, the changes required by the rule are applied simultaneously' (Chomsky and Halle, 1968, pg. 344).

$$[-\text{cons}] \rightarrow [+\text{nasal}] \;/\; [+\text{nasal}] \;\_\_$$

$$\text{NVVV} \mapsto \text{N}\tilde{\text{V}}\text{VV}$$

Simultaneous application

(2) 'To apply a rule, the entire string is first scanned for segments that satisfy the environmental constraints of the rule. After all such segments have been identified in the string, the changes required by the rule are applied simultaneously' (Chomsky and Halle, 1968, pg. 344).

$$[-\text{cons}] \rightarrow [+\text{nasal}] \ / \ [+\text{nasal}] \ \_\_$$

$$\text{NVVV} \mapsto \text{N}\tilde{\text{V}}\text{VV}$$

What about $\text{NVVV} \mapsto \text{N}\tilde{\text{V}}\tilde{\text{V}}\tilde{\text{V}}$

Parenthesis-star notation

$[-\text{cons}] \rightarrow [+\text{nasal}] \;/\; [+\text{nasal}]\ ([-\text{cons}])^{*}\ \_\_$

expands to

$[-\text{cons}] \rightarrow [+\text{nasal}] \;/\; [+\text{nasal}]\ \_\_$
$[-\text{cons}] \rightarrow [+\text{nasal}] \;/\; [+\text{nasal}]\ [-\text{cons}]\ \_\_$
$[-\text{cons}] \rightarrow [+\text{nasal}] \;/\; [+\text{nasal}]\ [-\text{cons}][-\text{cons}]\ \_\_$

$\text{NVVV} \mapsto \text{N}\tilde{\text{V}}\tilde{\text{V}}\tilde{\text{V}}$

Parenthesis-star notation

$$[-\text{cons}] \rightarrow [+\text{nasal}] \; / \; [+\text{nasal}] \; ([-\text{cons}])^{*} \; \_\_$$

Issues:

- Loss of target-trigger adjacency
- Redundancy: specification of target is repeated in the structural description

Directional/linear rules

- Given these issues with parenthesis-star, several proposals followed for doing away with simultaneous application (Anderson, 1969, 1974; Johnson, 1972; Howard, 1972; Lightner, 1972; Kenstowicz and Kisseberth, 1979).

Directional/linear rules

- Johnson (1972): rules apply right-to-left or left-to-right, with the direction specified for each rule

$$[-\text{cons}] \rightarrow [+\text{nasal}] \ / \ [+\text{nasal}] \ \_\_ \ (\text{left-to-right})$$

NVVV

Directional/linear rules

- Johnson (1972): rules apply right-to-left or left-to-right, with the direction specified for each rule

$$[-\text{cons}] \rightarrow [+\text{nasal}] \;/\; [+\text{nasal}] \;\_\_\; (\text{left-to-right})$$

$$\text{N}\tilde{\text{V}}\text{VV}$$

Directional/linear rules

- Johnson (1972): rules apply right-to-left or left-to-right, with the direction specified for each rule

$$[-\text{cons}] \rightarrow [+\text{nasal}] \ / \ [+\text{nasal}] \ \_\_ \ (\text{left-to-right})$$

$$\text{N}\tilde{\text{V}}\tilde{\text{V}}\text{V}$$

Directional/linear rules

- Johnson (1972): rules apply right-to-left or left-to-right, with the direction specified for each rule

$$[-\text{cons}] \rightarrow [+\text{nasal}] \;/\; [+\text{nasal}] \;\_\_ \;\; \text{(left-to-right)}$$

$$\text{N}\tilde{\text{V}}\tilde{\text{V}}\tilde{\text{V}}$$

Directional/linear rules

- Howard (1972): the direction of the rule is determined by its form, applying from the determinant toward the focus

$$[-\text{cons}] \rightarrow [+\text{nasal}] \;/\; [+\text{nasal}] \;\_\_$$

Directional/linear rules

What about NVVV $\mapsto$ N$\tilde{\text{V}}$VV?

Directional/linear rules

What about NVVV ↦ NṼVV?

- Johnson (1972): reverse direction

$$[-\text{cons}] \rightarrow [+\text{nasal}] \;/\; [+\text{nasal}] \_\_ \;(\text{right-to-left})$$

NVVV	NVVV	NVVV
↧	↧	↧
NVVV	NVVV	NṼVV

Directional/linear rules

What about NVVV $\mapsto$ N$\tilde{\text{V}}$VV?

- Howard (1972): simultaneous application (ideally with necessary and sufficient conditions for when this is necessary)

Optimality Theory

- OT is inherently iterative.

/NVVV/	*[+nas][−nas]	IDENT-NAS
[NVVV]	*!	
[NṼVV]	*!	*
[NṼṼV]	*!	**
☞[NṼṼṼ]		***

Optimality Theory

(3) Emergent Noniterativity Hypothesis (Kaplan, 2008): No formal entity in phonological grammars may require noniterativity.

Properties of extensions

- Different rules can be used to represent the same extension.

[−cons] → [+nasal] / [+nasal] ___ (simultaneous, or right-to-left iterative)

[−cons] → [+nasal] / [+cons, +nasal] ___

- The extension itself 'is what it is'.

{ (NVVV, NṼVV), ... }

Properties of extensions

- Relevant computational properties for this talk: input strict locality (ISL) and output strict locality (OSL).
- ISL and OSL functions are part of a subregular hierarchy of functions (see Heinz, 2018).
- These particular function classes are the ones that have been previously tied to the notion of rule application modes.

Input Strict Locality (ISL)

- In an ISL function, the output is determined based on a ‘window’ of the most recent *input*.

a → b / b __

b	a	a	a
b	b		

- Window is of size 2: this function is 2-ISL.

Input Strict Locality (ISL)

- In an ISL function, the output is determined based on a 'window' of the most recent *input*.

a → b / b __

b	a	a	a
b	b	a	

- Window is of size 2: this function is 2-ISL.

Output Strict Locality (OSL)

- In an OSL function, the output is determine based on a ‘window’ of the most recent *output*.

a → b / b __

b	a	a	a
b	b		

- Window is of size 2: this function is 2-OSL.

Output Strict Locality (OSL)

- In an OSL function, the output is determine based on a ‘window’ of the most recent *output*.

a → b / b __

b	a	a	a
b	b	b	

- Window is of size 2: this function is 2-OSL.

Rule application

$$a \rightarrow b \;/\; b \,\_\_$$

ISL = simultaneous application	{(baaa, bbaa) ... }
OSL = iterative application	{(baaa, bbbb) ... }

Rule application

$$a \rightarrow b \;/\; b \;\_\_$$

ISL ≈ simultaneous application	{(baaa, bbaa) ... }
OSL ≈ iterative application	{(baaa, bbbb) ... }

Rule application

$$a \rightarrow b \ / \ b \_\_$$

ISL ≈ simultaneous application {(baaa, bbaa) ... }
OSL ≈ iterative application {(baaa, bbbb) ... }

- Not all maps that appear to be 'simultaneous' are *necessarily* ISL.
- ISL and OSL also have distinct formal properties that affect what kinds of maps they can model.

Directionality

- With OSL but not ISL, direction matters.

Left-OSL					Right-OSL			
$a \to b \ / \ b \ \_\_$					$a \to b \ / \ \_\_ \ b$			
b	a	a	a		a	a	a	b
b	b	b	b		b	b	b	b

Delaying output

- With ISL but not OSL, output can be ‘delayed’.

a → b / a __ a

#	a	a	a	a	#
	a	λ			

Delaying output

- With ISL but not OSL, output can be ‘delayed’.

a → b / a __ a

#	a	a	a	a	#
	a	λ	b		

Delaying output

- With ISL but not OSL, output can be ‘delayed’.

a → b / a ___ a

#	a	a	a	a	#
	a	λ	b	b	

Delaying output

- With ISL but not OSL, output can be ‘delayed’.

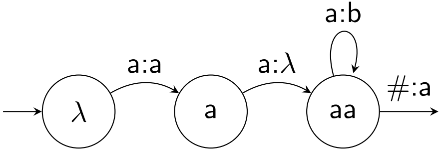
a → b / a __ a

#	a	a	a	a	#
	a	λ	b	b	a

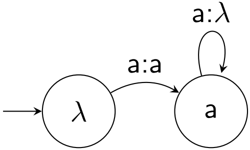
Delaying output

$$a \rightarrow b \; / \; a \_\_ a$$

ISL



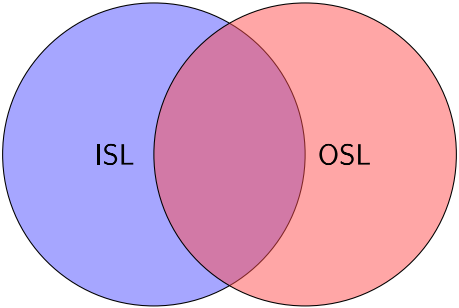
OSL



Do we need both?

- For many (most?) rules, ISL and OSL are extensionally-equivalent.
  - Final obstruent devoicing: {(bad, bat), (hʊnd, hʊnt), ... }
  - Nasal place assimilation: {(anpa, ampa), (anka, aŋka), (anpanpa, ampampa), ... }
- We need to look at rules that create additional triggers (i.e., self-feeding rules).

Do we need both?



Do we need ISL?

- Kaplan (2008, pg. 4): "There is no phenomenon that must be analyzed with a self-feeding rule that is not permitted to apply to its own output".
- Are there any phenomena that are *necessarily* ISL?

Analyses

- Noniterative processes that are still OSL.
  - Nasal assimilation
  - Lango vowel harmony
- Noniterative word-final deletion is necessarily ISL.
- Rules with two-sided contexts are necessarily ISL.
- Appendix: Tone sandhi rule that is necessarily ISL.

Nasal spreading/assimilation

$$[-\text{cons}] \rightarrow [+\text{nasal}] \;/\; [+\text{nasal}] \;\_\_$$

$$/\text{NVVV}/ \mapsto [\text{N}\tilde{\text{V}}\tilde{\text{V}}\tilde{\text{V}}] \quad \text{iterative}$$

$$/\text{NVVV}/ \mapsto [\text{N}\tilde{\text{V}}\text{VV}] \quad \text{simultaneous}$$

Nasal spreading/assimilation

$$[-\text{cons}] \rightarrow [+\text{nasal}] \;/\; [+\text{nasal}] \;\_\_$$

$$/\text{NVVV}/ \mapsto [\text{N}\tilde{\text{V}}\tilde{\text{V}}\tilde{\text{V}}] \quad \text{OSL}$$

$$/\text{NVVV}/ \mapsto [\text{N}\tilde{\text{V}}\text{VV}] \quad \text{ISL}$$

Nasal spreading/assimilation

- The language with the 'simultaneous map' instead has this rule:

  $[-\text{cons}] \rightarrow [+\text{nasal}] \ / \ [+\text{nasal}, +\text{cons}] \ \_\_$

- This rule is both ISL and OSL.

N	V	V	V
N	Ṽ		

Nasal spreading/assimilation

- The language with the ‘simultaneous map’ instead has this rule:

  [−cons] → [+nasal] / [+nasal, +cons] __

- This rule is both ISL and OSL.

N	V	V	V
N	Ṽ	V	

Vowel harmony

Two examples of regressive ATR harmony

(4) Kinande (Niger-Congo; Archangeli and Pulleyblank, 1994; Cole and Kisseberth, 1994)

a. /tU-ka-kI-*huk*-a/ ↦ [tukakihuka], ‘we cook it’

b. /tU-ka-kI-*lɪm*-a/ ↦ [tʊkakɪlɪma], ‘we cultivate it’

(5) Lango (Nilo-Saharan; Noonan, 1992; Kaplan, 2008)

a. /bɔ̀ŋɔ́-ní/ ↦ [bɔ̀ŋó-ní], ‘your (sg.) dress’

b. /cɔ̀ŋɔ̀-ní/ ↦ [cɔ̀ŋò-ní], ‘your (sg.) beer’

Vowel harmony

$V \rightarrow [\alpha\text{ATR}] \;/\; \_\_\; [\alpha\text{ATR}]$

- Kinande: iterative/OSL
- Lango: simultaneous/ISL

Vowel harmony

- Or, Lango doesn't have vowel harmony; this is assimilation (e.g., positional licensing (Kaplan, 2008))

$$V \rightarrow [\alpha\text{ATR}] \;/\; \underline{\quad} = [\alpha\text{ATR}]$$

ɔ	ɔ	=	i
	o	=	i

Vowel harmony

- Or, Lango doesn't have vowel harmony; this is assimilation (e.g., positional licensing (Kaplan, 2008))

$$V \rightarrow [\alpha\text{ATR}] \;/\; \_\_ = [\alpha\text{ATR}]$$

ɔ	ɔ	=	i
ɔ	o	=	i

Word-final deletion

$V \rightarrow \varnothing$ / __ #

(6) Hidatsa (Western Siouan; Harris, 1942; Kenstowicz and Kisseberth, 1979)

a. /cixi/ ↦ [cix], 'jump'
b. /kikua/ ↦ [kiku], 'set a trap'
c. /ikaa/ ↦ [ika], 'look'

Word-final deletion

$$V \rightarrow \varnothing \;/\; \_\_ \; \#$$

k	i	k	u	a	#
				λ	#

Word-final deletion

$$V \rightarrow \varnothing \; / \; \_\_ \; \#$$

k	i	k	u	a	$\#$
			λ	λ	$\#$

/kikua/ $\mapsto$ *[kik]

Rules with two-sided contexts

- Rules with two-sided contexts require delaying output, and so are necessarily ISL.

b	ʌ́	t	ɚ
b	ʌ́	λ	ɾɚ

- The exception is epenthesis.

C	C	V
C	əC	V

Rules with two-sided contexts

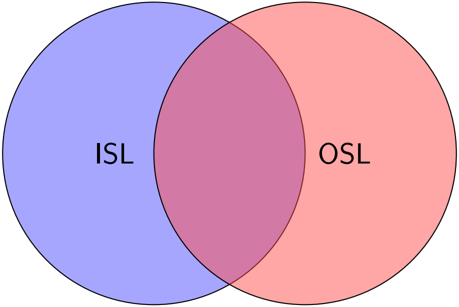
‘There are a number of well-motivated rules in a large number of languages in which the environment is of the form X__Y...So far as I can tell, the direction of iteration in these rules is irrelevant: one can go equally well from right-to-left or from left-to-right’ (Lightner, 1972, pg. 365).

Conclusions

- There is a role for both input- and output-locality in phonology.
- The limitation to iterativity is motivated by having a restrictive theory—with subregular functions we still have computational restrictiveness even if we allow for both ISL and OSL.

Conclusions

- Future work aims to better understand the exact role of ISL in the grammar.
- When phenomena are classified as ISL (e.g., Chandlee et al., 2018; Oakden and Chandlee, 2020; Dolatian and Rawski, 2020), it should be verified that they are *necessarily* ISL.



Open questions

- Can we predict which phenomena will be necessarily ISL?
- Are there rules with two-sided contexts that iterate?

References I

Ajíbóyè, O. (2001). Nasalization in Móbá. In S. Oh, N. Sawai, K. Shiobara, and R. Wojdak (Eds.), *Proceedings of the Northwest Linguistics Conference, University of British Columbia Working Papers in Linguistics 8*, pp. 1–18. Vancouver: University of British Columbia, Department of Linguistics.

Anderson, S. R. (1969). *West Scandinavian vowel systems and the ordering of phonological rules*. Ph. D. thesis, MIT.

Anderson, S. R. (1974). *The Organization of Phonology*. New York: Academic Press.

Archangeli, D. and D. Pulleyblank (1994). *Grounded Phonology*. Cambridge, MA: MIT Press.

Chandlee, J. (2014). *Strictly Local Phonological Processes*. Ph. D. thesis, University of Delaware.

Chandlee, J., J. Heinz, and A. Jardine (2018). Input strictly local opaque maps. *Phonology 35*(2), 171–205.

Chen, M. Y. (1986). The paradox of Tianjin tone sandhi. In *Proceedings of Chicago Linguistics Society 22*, pp. 98–114.

Chen, M. Y. (2000). *Tone sandhi: Patterns across Chinese dialects*. Cambridge: Cambridge UP.

Chomsky, N. and M. Halle (1968). *The Sound Pattern of English*. New York: Harper & Row.

Cole, J. and C. Kisseberth (1994). An optimal domains theory of harmony. *Studies in the Linguistic Sciences 24*(2), 101–114.

Dolatian, H. and J. Rawski (2020). Multi-input strictly local functions for templatic morphology. In *Proceedings of the Society for Computation in Linguistics*, Volume 3, pp. 282–296.

Harris, Z. (1942). Morpheme alternants in linguistic analysis. *Language 18*, 169–180.

Heinz, J. (2018). The computational nature of phonological generalizations. In L. Hyman and F. Plank (Eds.), *Phonological Typology*, Phonetics and Phonology, Chapter 5, pp. 126–195. De Gruyter Mouton.

Howard, I. (1972). *A directional theory of rule application in phonology*. Ph. D. thesis, MIT.

Hung, T. T. (1987). Tianjin tone sandhi: Towards a unified approach. *Journal of Chinese Linguistics 15*(2), 274–305.

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

References II

Kaplan, A. F. (2008). *Noniterativity is an emergent property of grammar*. Ph. D. thesis, University of California Santa Cruz.

Kaplan, R. M. and M. Kay (1994). Regular models of phonological rule systems. *Computational Linguistics* (20), 371–387.

Kenstowicz, M. and C. Kisseberth (1979). *Generative Phonology: Description and Theory*. Academic Press.

Lightner, T. M. (1972). *Problems in the Theory of Phonology Volume 1: Russian Phonology and Turkish Phonology*. Linguistic Research Inc.

Lin, H.-S. (2008). Variable directional applications in Tianjin tone sandhi. *Journal of East Asian Linguistics 17*(3), 181–226.

Noonan, M. (1992). *A Grammar of Lango*. Berlin, New York: Mouton de Gruyter.

Oakden, C. and J. Chandlee (2020). A computational analysis of tone sandhi ordering paradoxes. In *Supplemental proceedings of NELS 50*.

Prince, A. and P. Smolensky (1993). *Optimality Theory: Constraint Interaction in Generative Grammar*. ROA-537.

Prince, A. and P. Smolensky (2004). *Optimality Theory: Constraint Interaction in Generative Grammar*. Blackwell.

Tan, F. (1987). Tone sandhi in the Tianjin dialect. *Journal of Chinese Linguistics 15*(2), 228–246.

Wee, L.-H. (2010). A percolative account of Tianjin tone sandhi. *Language and Linguistics 11*(1), 21–64.

Zhang, Z. S. (1987). The paradox of Tianjin: Another look. *Journal of Chinese Linguistics 15*(2), 247–273.

Tianjin Tone Sandhi

- Three formally identical rules differ in directionality (Chen, 1986; Hung, 1987; Tan, 1987; Zhang, 1987; Chen, 2000; Lin, 2008; Wee, 2010)

(1)
a. $F \rightarrow L$ / __ F right-to-left
b. $L \rightarrow R$ / __ L right-to-left
c. $R \rightarrow H$ / __ R left-to-right

(1a) F [FF] ↧ FLF

(1b) L [LL] ↧ LRL

(1c) [RR] R ↧ H [RR] (1c) ↧ HHR

Tianjin Tone Sandhi

- Three formally identical rules differ in directionality (Chen, 1986; Hung, 1987; Tan, 1987; Zhang, 1987; Chen, 2000; Lin, 2008; Wee, 2010)

(1) a. F → L / __ F ROSL
b. L → R / __ L ROSL
c. R → H / __ R ISL

(1a)	FFF ↧ FLF	(1b)	LLL ↧ LRL	(1c)	RRR ↧ HHR