

# Modulating between input and output locality: A case study on phonological opacity

Jane Chandlee, Haverford College  
NELS 52 | Rutgers

October 29, 2021

# Inputs, outputs, and opacity

- ▶ From a rule-based perspective, opacity is often defined in terms of reference to the input structure or UR.
- ▶ Transparency instead requires reference to output structure, typically via rule ordering.

# Defining opacity

- ▶  $P, Q(x)$  = simultaneous application of rules  $P$  and  $Q$
- ▶  $Q$  is opaque if  $P(Q(x)) = P, Q(x) \neq Q(P(x))$

(Joshi and Kiparsky, 1979, 2005; Baković and Blumenfeld, 2017)

## Example: counterbleeding

Polish

[+back, -low] → [+high] / — [+voi, -nas]

[-son] → [-voi] / — #

	/ʒwob/
Raising	ʒwub
Devoicing	ʒwup
	[ʒwup]
	'crib'

(Bethin, 1978; Kenstowicz and Kisseberth, 1979; Baković, 2011)

## Example: counterbleeding

- ▶  $P, Q(x)$  = simultaneous application of rules P and Q
- ▶ Q is opaque if  $P(Q(x)) = P, Q(x) \neq Q(P(x))$

P = Devoicing

Q = Raising

x = *ʒwob*

$P(Q(x))$	$P, Q(x)$	$Q(P(x))$
<i>ʒwup</i>		* <i>ʒwop</i>
counterbleeding		bleeding

(see also Kenstowicz and Kisseberth, 1977, 1979)

# Inputs, outputs, and subregular functions

- ▶ Input strictly local (Chandlee, 2014)
- ▶ Output strictly local (Chandlee et al., 2015)
- ▶ Input tier-based (input) strictly local (Hao and Andersson, 2019; Burness et al., 2021)
- ▶ Output tier-based strictly local (Burness and McMullin, 2019)
- ▶ Output strictly piecewise (Burness and McMullin, 2020)
- ▶ Input-output strictly local (Chandlee et al., in prep)
- ▶ Input-output tier-based strictly local (Graf and Mayer, 2018; Hao and Bowers, 2019)
- ▶ Autosegmental input strictly local (Chandlee and Jardine, 2019, 2021)
- ▶ Multi-input strictly local (Dolatian and Rawski, 2020; Rawski and Dolatian, 2020)
- ▶ Input strictly local tree transducer (Graf, 2020)

# Inputs, outputs, and subregular functions

	input-oriented	output-oriented
noniterative spreading	ISL	
iterative spreading		OSL
icy targets	ITSL	
consonant harmony		OTSL
tone circles	ISL	
bounded tone spread/shift	AISL	
unbounded tone spread/shift		AOSL*
noniterative stress	ISL	
iterative stress		OSL*
default-to-opposite stress	TISL	
templatic morphology	MISL	
truncation		OSL
reduplication		OSL · OSL
feature assignment	ISL (trees)	

(Chandlee, 2014, 2017; Dolatian and Heinz, 2018; Hao and Andersson, 2019; Koser and Jardine, 2020; Dolatian and Rawski, 2020; Graf, 2020; Burness et al., 2021; Chandlee and Jardine, 2021; Oakden, Ms, ...)

# This talk

1. Does the input=opacity and output=transparency connection translate to these computational notions of input locality and output locality?
2. What modulates between input/output locality in the grammar?



# This talk

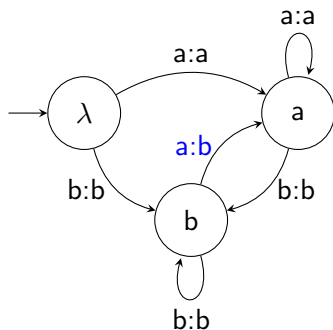
1. Does the input=opacity and output=transparency connection translate to these computational notions of input locality and output locality?
  - ▶ It depends...
2. What modulates between input/output locality in the grammar?
  - ▶ Ordering (via function composition)...

# Input local and output local functions

$a \rightarrow b / b \_$

Input Strictly Local

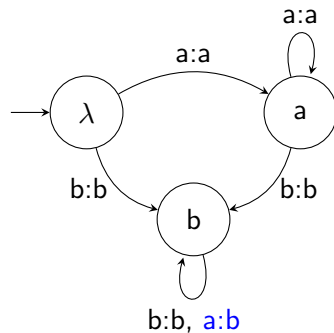
$k = 2$



$baaa \rightarrow bbaa$

Output Strictly Local

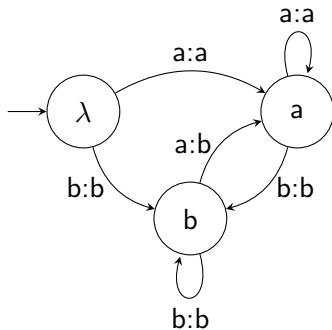
$k = 2$



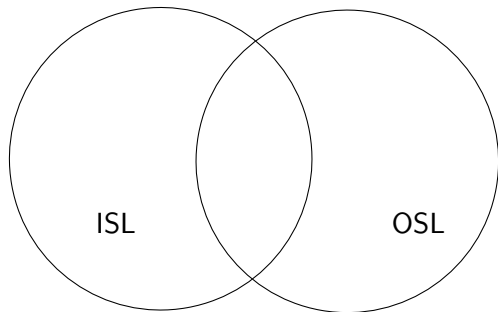
$baaa \rightarrow bbbb$

# Window of locality

Input Strictly Local ( $k = 2$ )



$$\text{ISL} \cap \text{OSL} \neq \emptyset$$



# Counterfeeding on environment

Bedouin Arabic

$a \rightarrow i / \_ \sigma$

$[-\text{cons}, -\text{syl}] \rightarrow [+syl] / C \_ \#$

	/badw/
Raising	—
Vocalization	badu
	[badu]
	'Bedouin'

(McCarthy, 1999; Baković, 2007)

# Counterfeeding on environment

P:  $d \rightarrow c / e \text{ —}$

Q:  $a \rightarrow b / c \text{ —}$

P feeds Q (creates a *trigger*)

## Counterfeeding on environment

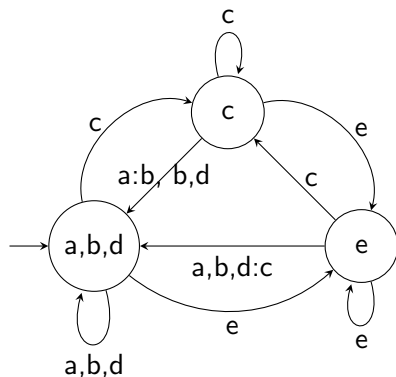
P:  $d \rightarrow c / e \_$

Q:  $a \rightarrow b / c \_$

P feeds Q (creates a *trigger*)

Single function analysis: one FST that combines the extensions of both rules.

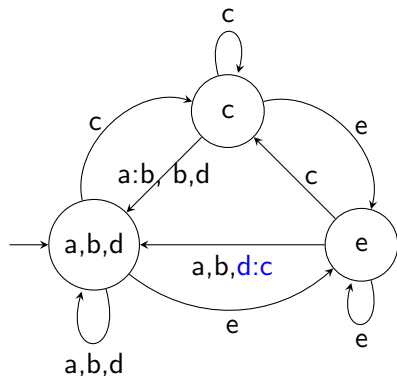
## Note on FSTs



- ▶ Transition labels:  $a = a:a$
- ▶ Minimized FSTs: distinct states for distinct triggers

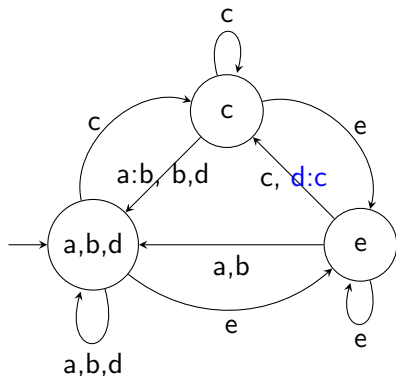


# ISL implements P,Q(x) and counterfeeding



Q: a  $\rightarrow$  b / c           /eda/  
 P: d  $\rightarrow$  c / e           eca  
 [eca]

# OSL implements feeding



$P: d \rightarrow c / e \_ \_$  /eda/  
eca  
 $Q: a \rightarrow b / c \_ \_$  ecb  
[ecb]

So far so good...

- ✓ ISL = simultaneous application/opacity
- ✓ OSL = transparency

# Counterfeeding on focus

Bedouin Arabic

$i \rightarrow \emptyset / \_ \sigma$

$a \rightarrow i / \_ \sigma$

	/katab/
Deletion	—
Raising	kitab
	[kitab]
	'he wrote'

(McCarthy, 1999; Baković, 2007)

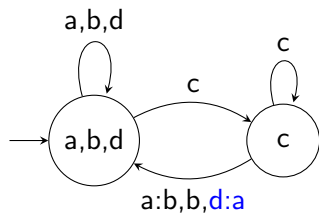
# Counterfeeding on focus

P:  $d \rightarrow a / c \text{ —}$

Q:  $a \rightarrow b / c \text{ —}$

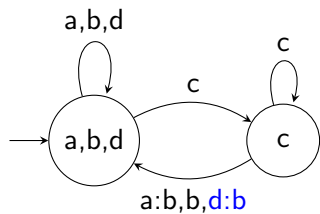
P feeds Q (creates a *target*)

# ISL $\cap$ OSL



Q: a  $\rightarrow$  b / c  $\_$   $\_$  /cd/  
P: d  $\rightarrow$  a / c  $\_$   $\_$  ca  
[ca]

# ISL $\cap$ OSL



P:  $d \rightarrow a / c \_ \_$  /cd/  
ca  
Q:  $a \rightarrow b / c \_ \_$  cb  
[cb]

# Interim summary I

1. Does the input=opacity and output=transparency connection translate to these computational notions of input locality and output locality?
  - ▶ It depends on the type of opacity.
  - ▶ On environment opacity:
    - ▶ Counterfeeding blocks derived triggers; bleeding blocks underlying triggers.
    - ▶ Counterbleeding allows underlying triggers that don't surface.
    - ▶ Feeding allows both underlying and derived triggers.
  - ▶ On focus opacity: triggers are fixed!

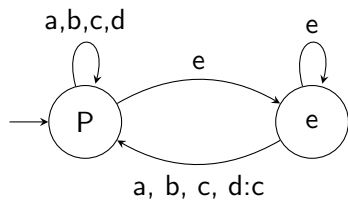
(see also McCarthy, 2007)



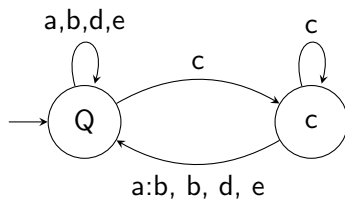
# Next

2. What modulates between input/output locality in the grammar?
  - ▶ Ordering (via composition)...

# FST composition



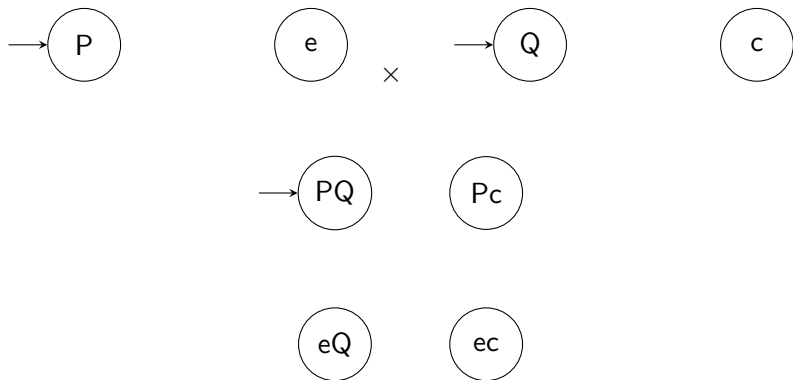
P:  $d \rightarrow c / e \_$



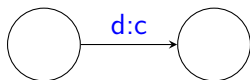
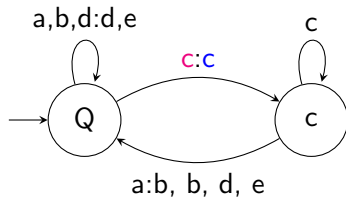
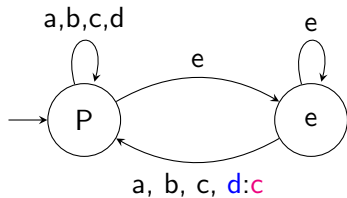
Q:  $a \rightarrow b / c \_$

$P, Q \in \text{ISL} \cap \text{OSL}$

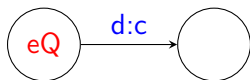
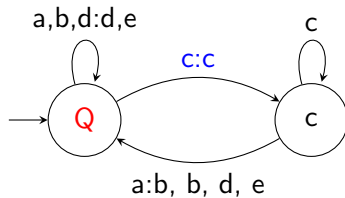
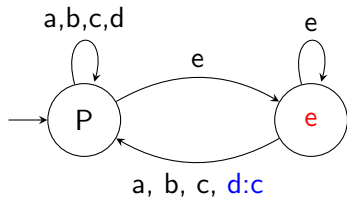
## FST composition: state set



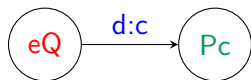
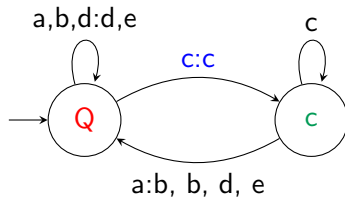
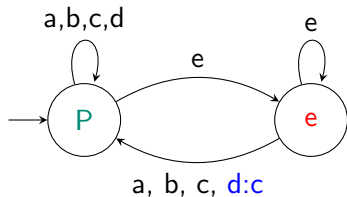
# FST composition: transitions



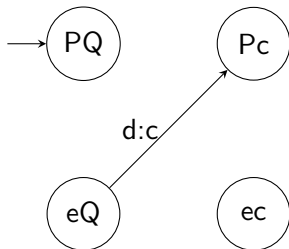
# FST composition: transitions



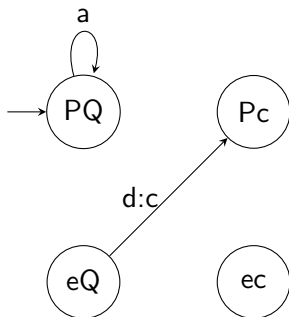
# FST composition: transitions



# FST composition

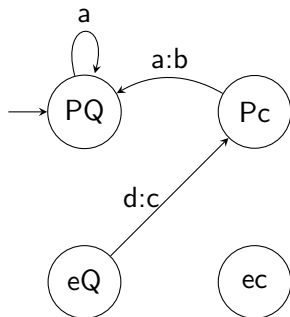


# FST composition

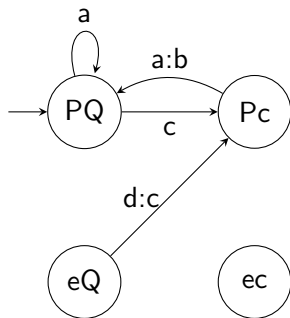




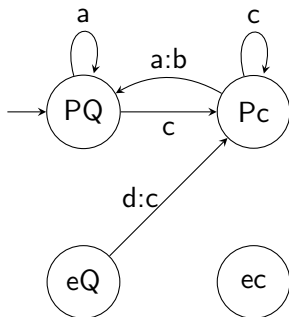
# FST composition



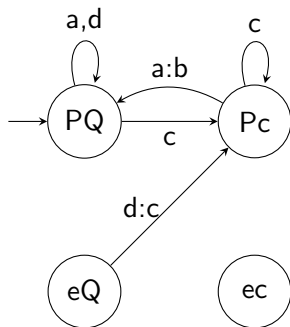
# FST composition



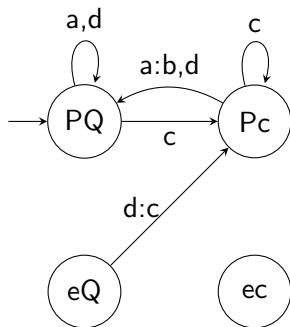
# FST composition



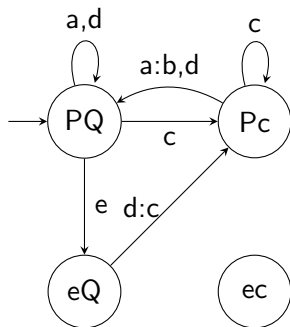
# FST composition



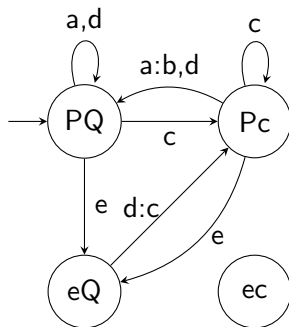
# FST composition



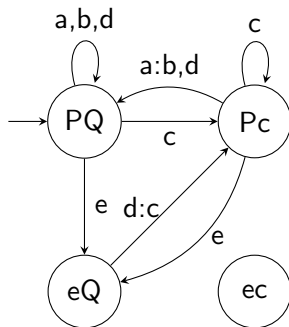
# FST composition



# FST composition

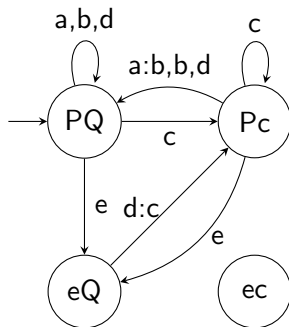


# FST composition

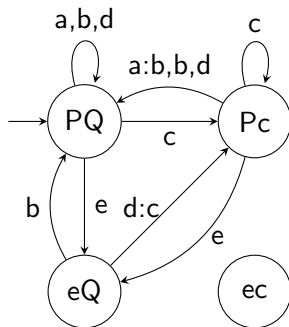




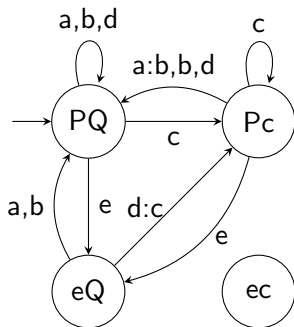
# FST composition



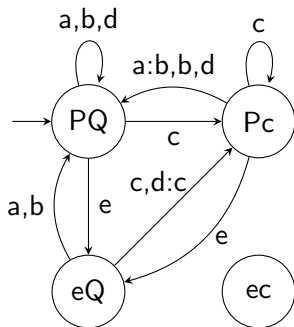
# FST composition



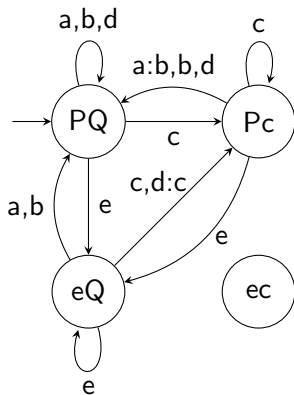
# FST composition



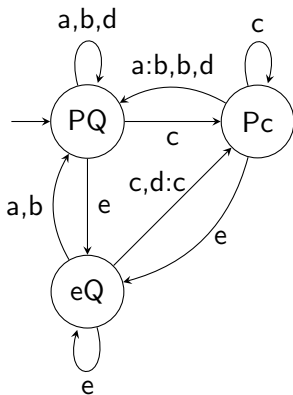
# FST composition



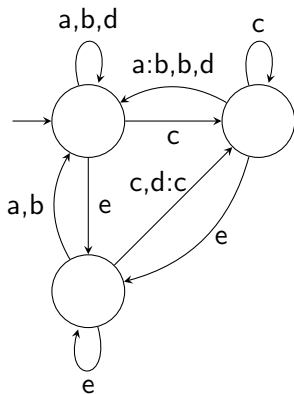
# FST composition



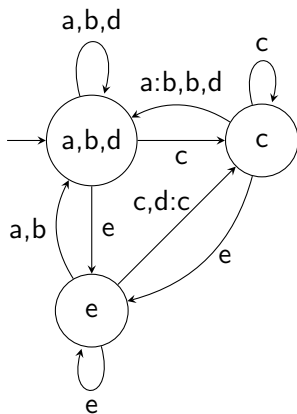
# FST composition



# FST composition

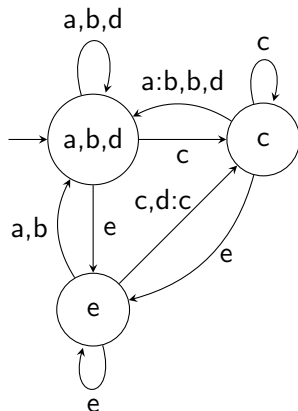


Result of composition is OSL ( $k=2$ )



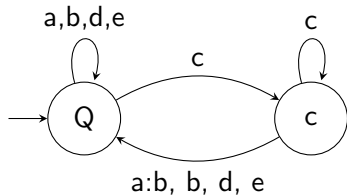


# $Q(P(x))$ generates feeding

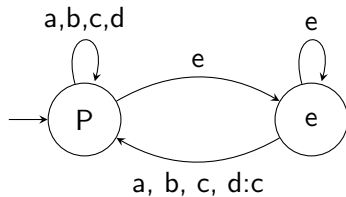


$P: d \rightarrow c / e \_ \quad /eda/$   
 $Q: a \rightarrow b / c \_ \quad eca$   
 $[ecb]$

# What about $P(Q(x))$ ?

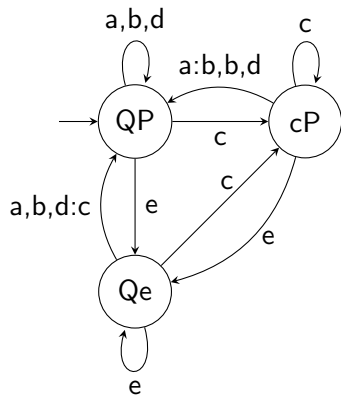


Q:  $a \rightarrow b / c \_$

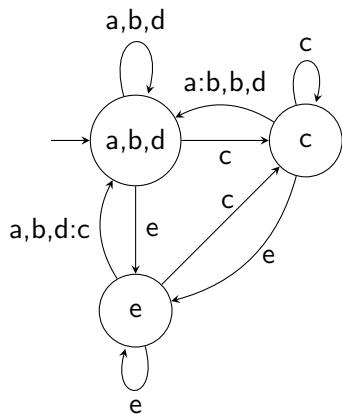


P:  $d \rightarrow c / e \_$

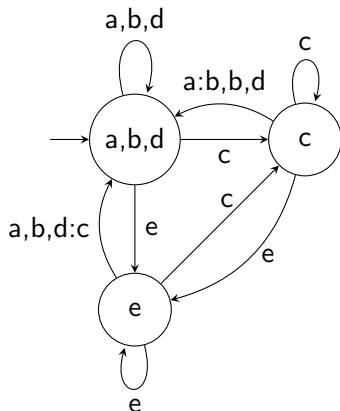
## Result of composition



Result of composition is ISL ( $k=2$ )



# $P(Q(x))$ generates counterfeeding



Q:  $a \rightarrow b / c \text{ — } \text{—}$   $/eda/$   
 P:  $d \rightarrow c / e \text{ — } \text{—}$   $eca$   
 $[eca]$

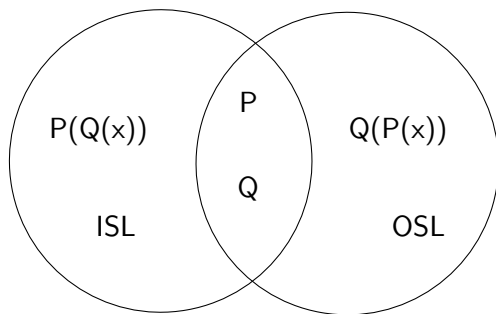
## Opacity and ordering

$P(Q(x))$	$P, Q(x)$	$Q(P(x))$
counterfeeding		feeding
counterbleeding		bleeding
ISL		OSL

## Opacity and ordering

- ▶ The use of input locality for opacity and output locality for transparency doesn't have to be stipulated: it is automatically enforced via the order of composition.

## Composition modulates input/output locality

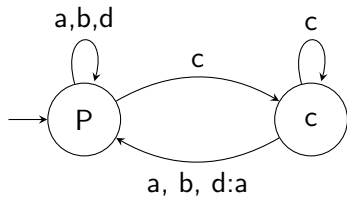




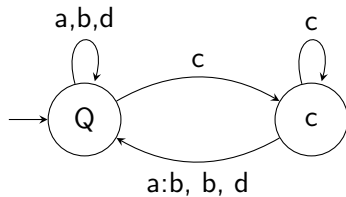
## On focus opacity

- ▶ Prediction for 'on focus' interactions: order of composition doesn't matter w.r.t. input versus output locality.

# (Counter)feeding on focus

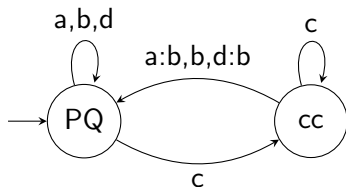


P:  $d \rightarrow a / c \_$

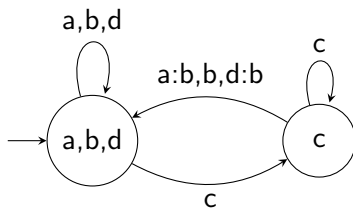


Q:  $a \rightarrow b / c \_$

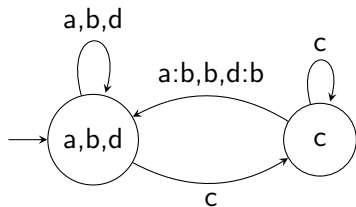
Result of composition:  $Q(P(x))$



## Result of composition: $ISL \cap OSL$

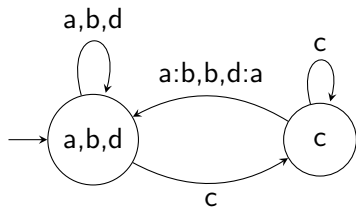


# Q(P(x)) generates feeding



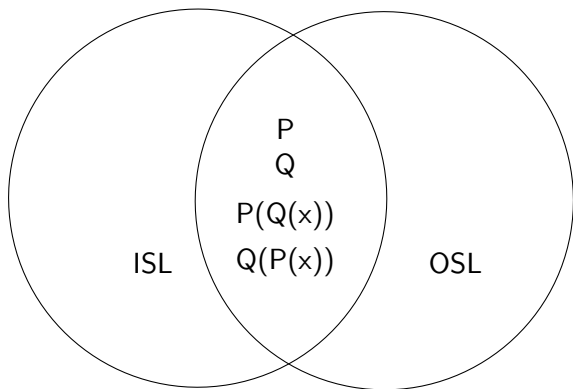
P:  $d \rightarrow a / c \text{ —}$     /cd/  
ca  
Q:  $a \rightarrow b / c \text{ —}$     cb  
[cb]

# $P(Q(x))$ generates counterfeeding



		/cd/
Q: a → b / c	—	—
P: d → a / c	—	ca
		[ca]

## On focus opacity



## Ordering determines input/output locality

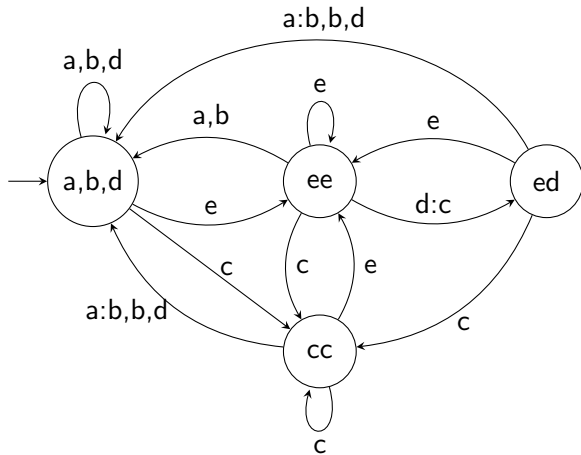
- ▶ Whether an interaction is ISL, OSL, or both is automatically enforced by the operation of composition.



## Size of the window of locality

- ▶ Working assumption:  $k=2$  for a rule of the form  $x \rightarrow y / z \_$
- ▶ If we relax this assumption, feeding is still ISL.

## Size of the window of locality



$d \rightarrow c / e \text{ —}$

$a \rightarrow b / c \text{ —}$

$a \rightarrow b / \{c, ed\} \text{ —}$

$eda \rightarrow ecb$

# Proposal

- ▶ Processes are modeled with  $k$ -local functions, where  $k$  is the length of the corresponding rule's structural description:

$$\alpha \rightarrow \beta / \delta \_ \quad k = |\delta \alpha|$$

$$\alpha \rightarrow \beta / \_ \rho \quad k = |\alpha \rho|$$

- ▶ Input and output locality are enforced via composition in order to maintain this restriction on  $k$ .

# Interim summary II

2. What modulates between input/output locality in the grammar?
  - ▶ Ordering (via composition), provided the window of locality is limited to the length of the structural description.

# Combining opacity and transparency

- ▶ Fed counterfeeding
- ▶ Mutual bleeding (bled counterbleeding)
- ▶ Self-destructive feeding/seeding

(Baković, 2007, 2011; Kavitskaya and Staroverov, 2010; Baković and Blumenfeld, 2019)

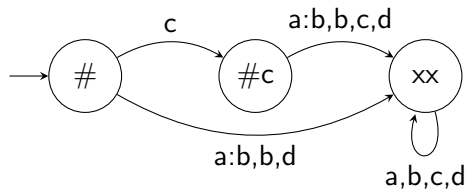
# Fed counterfeeding (on environment)

## Tundra Nenets (Uralic)

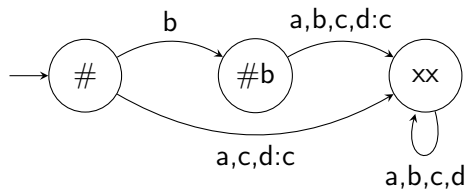
	/tasʌ/	/tʲim-jʌ-s/
C → ʔ / — #	—	tʲimjʌʔ
ʌ → ∅ / — (ʔ) #	tas	tʲimjʔ
	[tas]	[tʲimjʔ]
	‘whole’	‘it rotted’

(Kavitskaya and Staroverov, 2010)

# Fed counterfeeding (on environment)

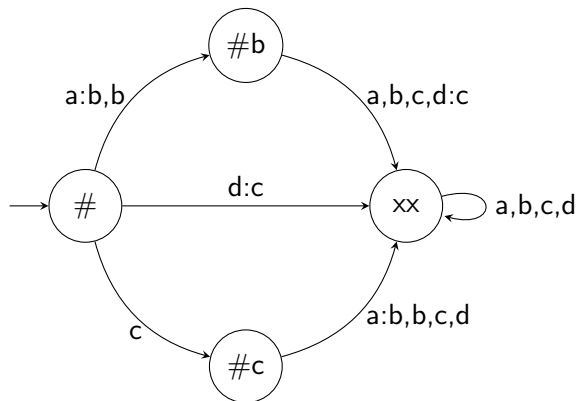


$a \rightarrow b / \# (c) \_$



$d \rightarrow c / \# (b) \_$

## Result of composition



	/da/	/ad/
a → b / # (c) —	—	bd
d → c / # (b) —	ca	bc
	[ca]	[bc]



## Conclusions and next steps

- ▶ Goal of subregular function analyses: understand the computational properties of phonological extensions.
- ▶ Distinguishing ISL, OSL, and  $ISL \cap OSL$  is necessary to achieve this goal.
- ▶ Limiting  $k$  to the length of the rule's structural description is a phonologically-motivated restriction imposed on the existing computational restrictions.
  - ▶ Better captures generalizations.
  - ▶ Enables composition to modulate input/output locality.

## Conclusions and next steps

- ▶ Expansion of proposal to other rule forms:
  - ▶  $\alpha \rightarrow \beta/\delta \text{---} \rho$
  - ▶  $\alpha \rightarrow \emptyset$
  - ▶  $\emptyset \rightarrow \beta$
- ▶  $k$ -ISL/OSL are not closed under composition, but certain subsets might be.
- ▶ Process interactions can help us to identify both the phonological and the formal conditions under which closure under composition holds.

# References I

- Baković, E. (2007). A revised typology of opaque generalisations. *Phonology*, 24(2):217–259.
- Baković, E. (2011). Opacity and ordering. In Goldsmith, J. A., Riggle, J., and Yu, A. C. L., editors, *The Handbook of Phonology Theory*, pages 40–67. 2 edition.
- Baković, E. and Blumenfeld, L. (2017). The interaction of phonological maps: a set-theoretic typology. MOT Phonology Workshop, UQAM.
- Baković, E. and Blumenfeld, L. (2019). Rule interaction conversion operations. *Loquens*, 6.2:e062.
- Bethin, C. (1978). *Phonological rules in the nominative singular and genitive plural of the Slavic substantive declension*. PhD thesis, University of Illinois, Champaign-Urbana.
- Burness, P. and McMullin, K. (2019). Efficient learning of output tier-based strictly 2-local functions. In *Proceedings of the 16th Meeting on the Mathematics of Language*, pages 78–90, Toronto, Canada. Association for Computational Linguistics.
- Burness, P., McMullin, K., and Chandlee, J. (2021). Long-distance phonological processes as tier-based strictly local functions. *Glossa*.
- Burness, P. A. and McMullin, K. (2020). Modelling non-local maps as strictly piecewise functions. In *Proceedings of the Society for Computation in Linguistics*, volume 3. Article 60.
- Chandlee, J. (2014). *Strictly Local Phonological Processes*. PhD thesis, University of Delaware.
- Chandlee, J. (2017). Computational locality in morphological maps. *Morphology*, 27(4):599–641.
- Chandlee, J., Eyraud, R., and Heinz, J. (2015). Output strictly local functions. In *Proceedings of the 14th Meeting on the Mathematics of Language (MOL 2015)*.
- Chandlee, J., Eyraud, R., and Heinz, J. (in prep). Input-output strictly local functions and their efficient learnability. Ms.
- Chandlee, J., Heinz, J., and Jardine, A. (2018). Input strictly local opaque maps. *Phonology*, 35(2):171–205.
- Chandlee, J. and Jardine, A. (2019). Autosegmental input-strictly local functions. *Transactions of the Association for Computational Linguistics*, 7:157–168.
- Chandlee, J. and Jardine, A. (2021). Input and output locality and representation. *Glossa: A Journal of General Linguistics*, 6(43).

# References II

- Dolatian, H. and Heinz, J. (2018). Learning reduplication with 2-way finite-state transducers. In Unold, O., Dyrka, W., and Wieczorek, W., editors, *Proceedings of the 14th International Conference on Grammatical Inference*, volume 93 of *Proceedings of Machine Learning Research*, pages 67–80. PMLR.
- Dolatian, H. and Rawski, J. (2020). Multi-input strictly local functions for templatic morphology. In *Proceedings of the Society for Computation in Linguistics*, volume 3. Association for Computational Linguistics.
- Graf, T. (2020). Curbing feature coding: Strictly local feature assignment. Slides of a talk given at *SciL 2020*, January 2–5, co-located with the LSA 2020.
- Graf, T. and Mayer, C. (2018). Sanskrit n-retroflexion is input-output tier-based strictly local. In *Proceedings of SIGMORPHON 2018*, pages 151–160.
- Hao, Y. and Andersson, S. (2019). Unbounded stress in subregular phonology. In *Proceedings of the 16th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology and Morphology*, pages 135–143, Florence, Italy. Association for Computational Linguistics.
- Hao, Y. and Bowers, D. (2019). Action-sensitive phonological dependencies. In *Proceedings of the 16th Workshop on Computational Research in Phonetics, Phonology and Morphology*, pages 218–228, Florence, Italy. Association for Computational Linguistics.
- Joshi, S. and Kiparsky, P. (1979). Siddha and asiddha in Pāṇinian phonology. In Dinnsen, D., editor, *Current Approaches to Phonological Theory*. Indiana University Press, Bloomington, Indiana.
- Joshi, S. D. and Kiparsky, P. (2005). The extended *siddha*-principle. *Annals of the Bhandarkar Oriental Research Institute*, 86:1–26.
- Kavitskaya, D. and Staroverov, P. (2010). When an interaction is both opaque and transparent: the paradox of fed counterfeeding. *Phonology*, 27:255–288.
- Kenstowicz, M. and Kisseberth, C. (1977). *Topics in Phonological Theory*. Academic Press, New York/San Francisco/London.
- Kenstowicz, M. and Kisseberth, C. (1979). *Generative Phonology: Description and Theory*. Academic Press, New York/San Francisco/London.
- Koser, N. and Jardine, A. (2020). The computational nature of stress assignment. In Baek, H., Takahashi, C., and Yeung, A. H.-L., editors, *Proceedings of the 2019 Meeting on Phonology*, Proceedings of the Annual Meetings on Phonology. Washington, DC: LSA.

# References III

McCarthy, J. J. (1999). Sympathy and phonological opacity. *Phonology*, 16:331–399.

McCarthy, J. J. (2007). *Hidden Generalizations: Phonological Opacity in Optimality Theory*. London: Equinox.

Oakden, C. (Ms). The ticuna nominalizer tone circle and input strict-locality.

Rawski, J. and Dolatian, H. (2020). Multi-input strictly local functions for tonal phonology. In *Proceedings of the Society for Computation in Linguistics*, volume 3. Association for Computational Linguistics.

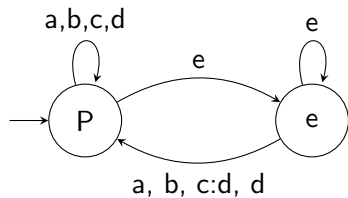
# Acknowledgements

Thank you!

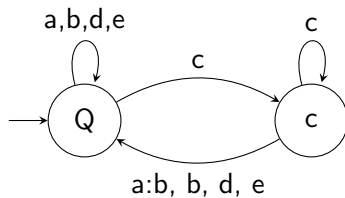
Thanks also to Adam Jardine, Eric Baković, and participants of AMP 2021 for helpful discussion and feedback.

And thanks to the organizers of NELS 52!

# (Counter)bleeding on environment

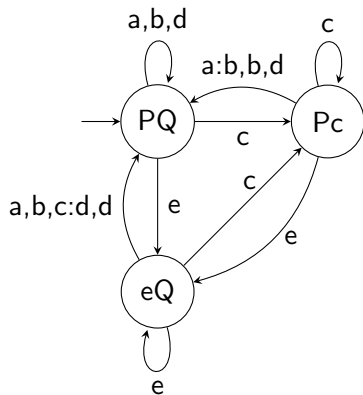


P:  $c \rightarrow d / e \_$



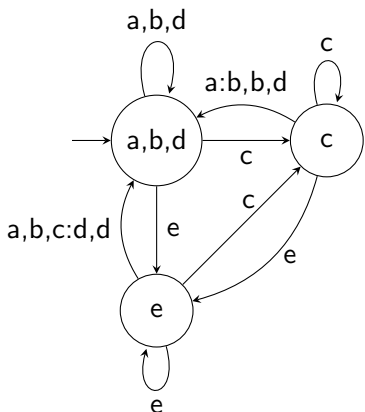
Q:  $a \rightarrow b / c \_$

$Q(P(x))$



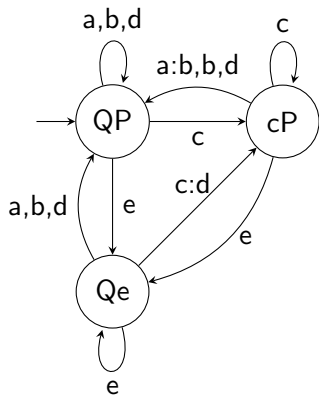


$Q(P(x))$  is 2-OSL, generates bleeding

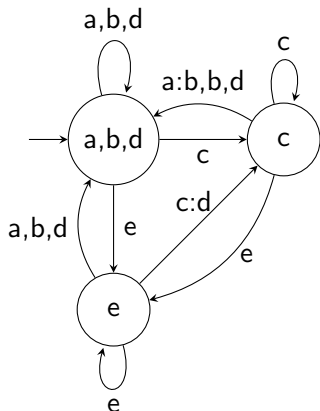


P: $c \rightarrow d$	/	$e$	—	/eca/
Q: $a \rightarrow b$	/	$c$	—	eda
				[eda]

$P(Q(x))$

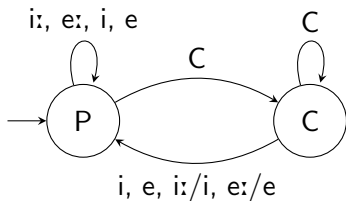


# $P(Q(x))$ is 2-ISL, generates counterbleeding



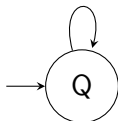
Q:  $a \rightarrow b / c \_ \_$  /eca/  
ecb  
P:  $c \rightarrow d / e \_ \_$  edb  
[eda]

# (Counter)bleeding on focus



P:  $V: \rightarrow V / C \_$

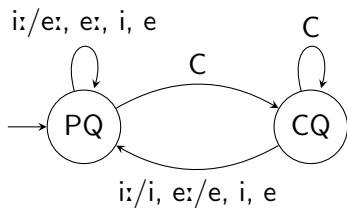
C, i, e, e:, i:/e:



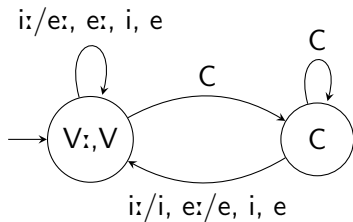
Q:  $i: \rightarrow e:$

Based on Yokuts (McCarthy, 1999)

$Q(P(x))$

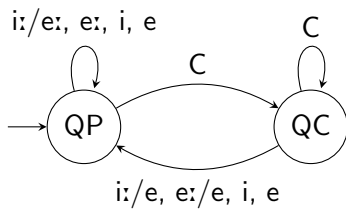


$Q(P(x))$  is 2-ISL and 2-OSL, generates bleeding

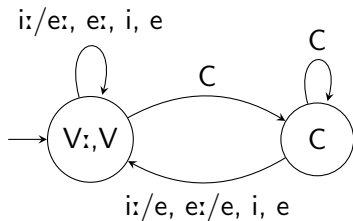


$P: V: \rightarrow V / C \text{ —}$       $/Ci:/$   
 $Q: ix \rightarrow ex$                       $Ci$   
    —  
     $[Ci]$

$P(Q(x))$



$P(Q(x))$  is 2-ISL and 2-OSL, generates counterbleeding



$Q: i: \rightarrow e: /$        $/Ci:/$   
 $P: V: \rightarrow V / C \_$        $Ce:$   
     $Ce$   
     $[Ce]$



## Previous work

- ▶ Chandlee et al. (2018) show 7 examples of opaque maps are all ISL for some  $k$ .

cross-derivational feeding	Lithuanian	$k = 2$
counterbleeding	Yowlumne	$k = 3$
non-gratuitous feeding	Classical Arabic	$k = 3$
fed counterfeeding	Tundra Nenets	$k = 3$
counterfeeding on environment	Bedouin Arabic	$k = 3$
counterfeeding on focus	Bedouin Arabic	$k = 3$
self-destructive feeding	Turkish	$k = 5$

(McCarthy, 1999; Baković, 2007; Kavitskaya and Staroverov, 2010)