

# TYPES OF ORDERINGS AND THEIR IMPLICATIONS FOR PHONOLOGY AND MORPHOLOGY

Jeffrey Heinz



Stony Brook University

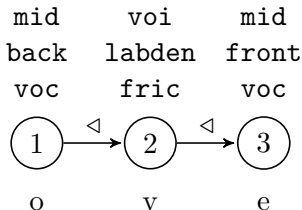
NAPhC 13  
Concordia University  
2025 05 10

# TODAY

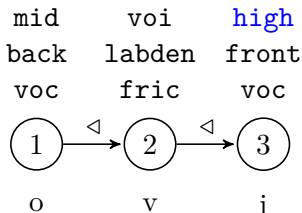
- Informally, by *ordering* I mean a systematic way in which the ordering of the elements in the input are related to the ordering of the elements in the output.
- I discuss two kinds of orderings from a mathematical and computational perspective and argue that phonology is characterized by one, and morphology by the other.
- Proposed universal properties — such as these orderings — make predictions for typology, psycho- and neuro-linguistics, and acquisition/learnability.

# MAPPING INPUTS TO OUTPUTS

## Finnish word-final /e/ raising (Odden 2014)

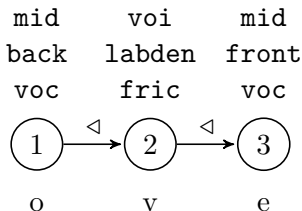


‘door, nom-sg.’



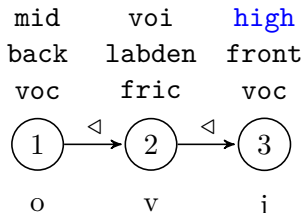
# MAPPING INPUTS TO OUTPUTS

## Finnish word-final /e/ raising (Odden 2014)



‘door, nom-sg.’

- There are 3 positions.
- Positions have properties.
- Positions are ordered according to a relation.
- Input and output positions correspond.



# INTRODUCING ORDER PRESERVATION WITH METATHESIS

**Uab Meto** (Mooney, 2022): /kokise/ → [kōikse] ‘the bread’

**Input:**

k		o		k		i		s		e
①	◁	②	◁	③	◁	④	◁	⑤	◁	⑥

# INTRODUCING ORDER PRESERVATION WITH METATHESIS

**Uab Meto** (Mooney, 2022): /kokise/ → [kōikse] ‘the bread’

**Output:** (transpose positions, not properties of those positions)

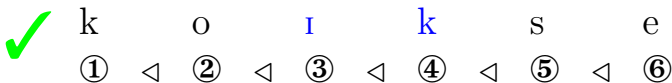


**Not order preserving**

# INTRODUCING ORDER PRESERVATION WITH METATHESIS

**Uab Meto** (Mooney, 2022): /kokise/ → [kōikse] ‘the bread’

**Output:** (change properties, but not the order of the positions)



**Order preserving**

# OUTPUTS DIFFER IN SIZE FROM INPUTS

## **Lardil word-final vowel deletion:**

/yalulu/ → [yalul] ‘flame, uninflected’

## **Malagasy word-final vowel epenthesis:**

/tanana/ → [tanana] ‘hand’

## **Indonesian plural formation:**

/buku/ → [bukubuku] ‘book, plural’

How do the input and output positions correspond?

---

Kenstowicz and Kisseberth 1979, Cohn 1989, Albrow 2005, O’Neill 2015



# LOGICAL TRANSDUCTIONS

- 1 Logical formulas *license* output positions, and unlicensed positions – and all associated relations – delete.
- 2 A transduction comes with a **fixed** *copy size*. This specifies how many copies of each input position is licensable in the output.

---

Courcelle 1994, Courcelle and Engelfriedt 2012

# EXAMPLES

**Lardil:** /yalulu/  $\rightarrow$  [yalul]. 1 copy per position.

Input:

y	a	l	u	l	u
•	•	•	•	•	•

# EXAMPLES

**Lardil:** /yalulu/ → [yalul]. 1 copy per position.

Output:

y	a	l	u	l	u
•	•	•	•	•	•

# EXAMPLES

Malagasy: /tanana/ → [tanana]. 2 copies per position.

Input:

t	a	n	a	n
•	•	•	•	•

# EXAMPLES

Malagasy: /tanana/  $\rightarrow$  [tanana]. 2 copies per position.

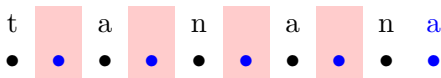
Output:

t        a        n        a        n  
●   ●   ●   ●   ●   ●   ●   ●   ●   ●

# EXAMPLES

Malagasy: /tanana/ → [tanana]. 2 copies per position.

Output:



# EXAMPLES

**Indonesian:** /buku/  $\rightarrow$  [bukubuku]. **2 copies per position.**

Input:

b	u	k	u
•	•	•	•

# EXAMPLES

**Indonesian:** /buku/  $\rightarrow$  [bukubuku]. 2 copies per position.

Output:

b   u   k   u  
•   •   •   •   •   •   •   •



# EXAMPLES

**Indonesian:** /buku/  $\rightarrow$  [bukubuku]. 2 copies per position.

Output:

b	u	k	u	b	u	k	u
●	●	●	●	●	●	●	●

# TWO NATURAL ORDERINGS FOR LINEARIZING THE OUTPUT

Consider 4 positions and a copyset of size 3.

		Position			
		1	2	3	4
Copy	1	●	●	●	●
	2	●	●	●	●
	3	●	●	●	●

position-wise

● ● ● ● ● ● ● ● ● ● ● ●

order by position, then by copy

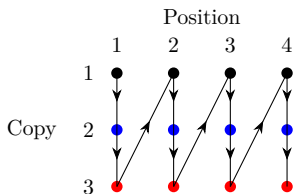
copy-wise

● ● ● ● ● ● ● ● ● ● ● ●

order by copy, then by position

# ANOTHER VIEW

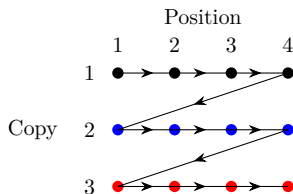
Consider 4 positions and a copyset of size 3.



position-wise



order by position, then by copy



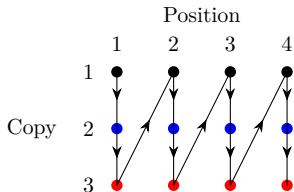
copy-wise



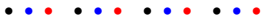
order by copy, then by position

# ANOTHER VIEW

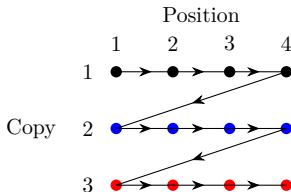
Consider 4 positions and a copyset of size 3.



position-wise



order by position, then by copy



copy-wise



order by copy, then by position

\*In theoretical computer science, only the position-wise ordering is referred to as “order-preserving”.

# TECHNICAL NOTE

- Both the position-wise and copy-wise orderings are easy to define in First Order logic with general precedence:  $\text{FO}(<)$ .
- They are not First Order definable with successor,  $\text{FO}(\triangleleft)$ , but they are definable with Monadic Second Order logic and successor,  $\text{MSO}(\triangleleft)$ .

- **Phonological** processes obey **position-wise** ordering.
- **Morphological** processes obey **copy-wise** ordering.

# MANY PROCESSES ARE BOTH POSITION-WISE AND COPY-WISE

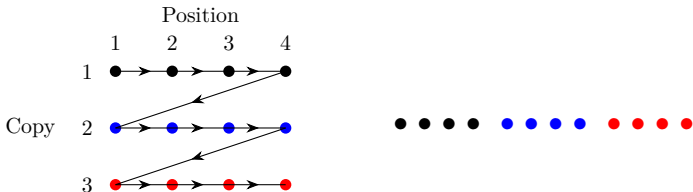
- prefixation, suffixation, affixation, infixation
- deletion
- local assimilation
- consonant harmony
- vowel harmony
- ...

# EVIDENCE FROM TOTAL REDUPLICATION

## Indonesian plural formation:

/buku/ → [bukubuku] ‘book, plural’

copy-wise: order by copy, then by position



Total reduplication is ...

a morphological – not phonological – process, and reflects the copy-wise ordering.

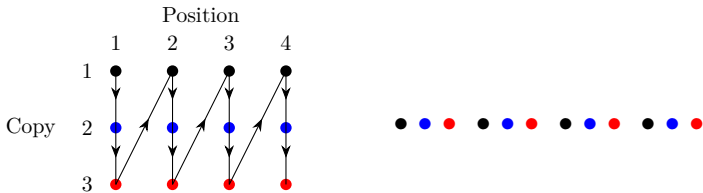


# EVIDENCE FROM TOTAL REDUPLICATION

## Indonesian plural formation:

/buku/  $\rightarrow$  [bukubuku] ‘book, plural’

position-wise: order by position, then by copy



Total reduplication is ...

difficult (impossible?) to do with a position-wise ordering.

# EVIDENCE FROM METATHESIS

**Uab Meto** (Mooney, 2022):

/kokise/ → [kōikse] ‘the bread’

**Kwara’ae** (Heinz, 2005; Takahashi, 2019):

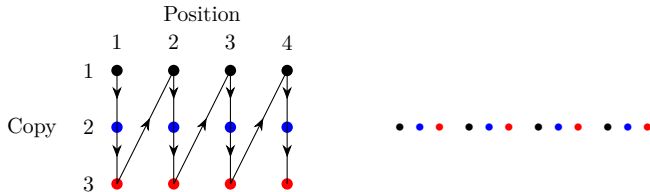
/ketalaku/ → [keatlaʊk] ‘my height’

Takahashi (2019) and Mooney (2022)

Both argue phonology contains no transposition operation, and that metathesis is the outcome of copy and deletion processes (cf. Blevins and Garrett 1998).

# COPY AND DELETE, POSITION-WISE

position-wise: order by position, then by copy



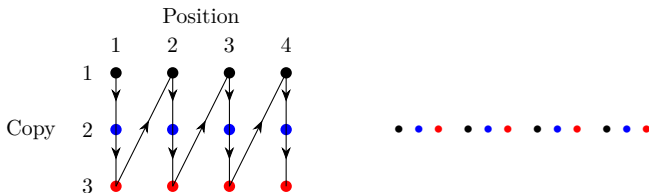
**Uab Meto:** /kokise/  $\rightarrow$  [k $\widehat{o}$ ikse] ‘the bread’

Input:

k   o   k   i   s   e  
●   ●   ●   ●   ●   ●

# COPY AND DELETE, POSITION-WISE

position-wise: order by position, then by copy



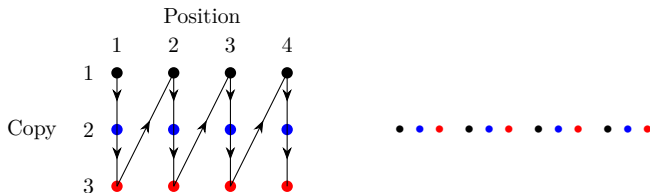
**Uab Meto:** /kokise/  $\rightarrow$  [kōikse] ‘the bread’

Output (make space):

k      o      k      i      s      e  
●   ●   ●   ●   ●   ●   ●   ●   ●   ●   ●   ●

# COPY AND DELETE, POSITION-WISE

position-wise: order by position, then by copy



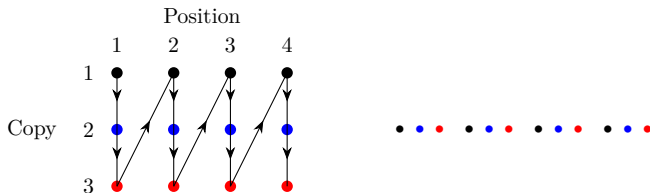
**Uab Meto:** /kokise/  $\rightarrow$  [kōikse] ‘the bread’

Output (copy):

k      o    i    k      i      s      e  
●    ●    ●    ●    ●    ●    ●    ●    ●    ●

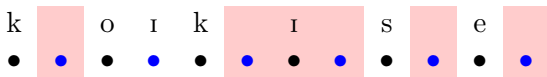
# COPY AND DELETE, POSITION-WISE

position-wise: order by position, then by copy



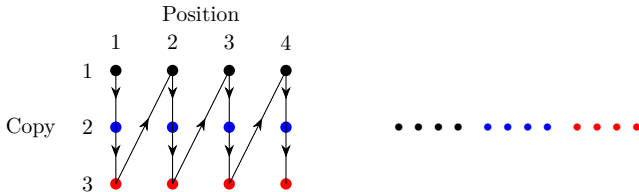
**Uab Meto:** /kokise/  $\rightarrow$  [kōikse] ‘the bread’

Output (copy and delete):



# COPY AND DELETE, COPY-WISE

copy-wise: order by copy, then by position



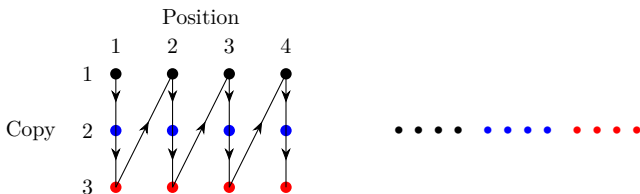
**Uab Meto:** /kokise/  $\rightarrow$  [k $\widehat{o}$ ikse] ‘the bread’

Input:

k o k i s e  
• • • • •

# COPY AND DELETE, COPY-WISE

copy-wise: order by copy, then by position



**Uab Meto:** /kokise/  $\rightarrow$  [k<sup>h</sup>oikse] ‘the bread’

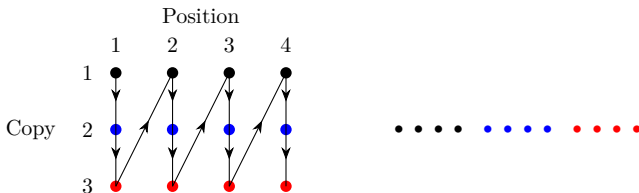
Output (make space):

k o k i s e  
● ● ● ● ● ● ● ● ● ● ● ●



# COPY AND DELETE, COPY-WISE

copy-wise: order by copy, then by position



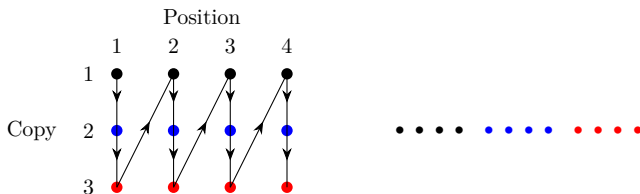
**Uab Meto:** /kokise/  $\rightarrow$  [kōikse] ‘the bread’

Output (copy):

k o k i s e k o k i s e  
● ● ● ● ● ● ● ● ● ● ● ●

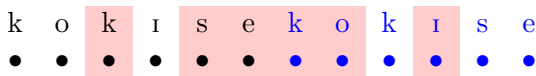
# COPY AND DELETE, COPY-WISE

copy-wise: order by copy, then by position



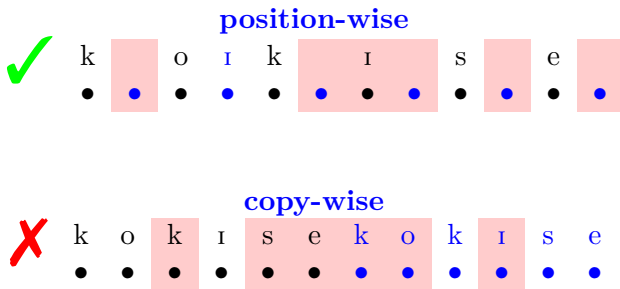
**Uab Meto:** /kokise/  $\rightarrow$  [kōikse] ‘the bread’

Output (delete):



# OBSERVATION #1

The position-wise ordered function resembles the copy-and-delete analyses of Takahasi (2019) and Mooney (2022), but the copy-wise ordering function does not.



## OBSERVATION #2

This copy-wise ordering necessarily *bounds* the number of instances of metathesis, regardless of the size of the input, because

- 1 the size of the copy is fixed in advance, and
- 2 copy-wise ordered functions requires a copy set of size  $n + 1$  for  $n$  instances of metathesis.

**Kwara'ae:** /ketalaku/  $\rightarrow$  [keatla<sub>̩</sub>uk] 'my height'

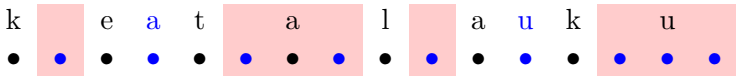
k	e	t	a	l	a	k	u	k	e	t	a	l	a	k	u
•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•

Need another copy to linearize [uk]!

## OBSERVATION #3

In contrast, For position-wise ordered functions with a copy set of size 2, it's the size of the input which bounds the number of instances of metathesis.

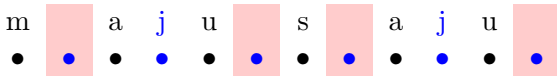
**Kwara'ae:** /ketalaku/ → [keatlaŋk] 'my height'



## OBSERVATION #4

This difference between position-wise and copy-wise ordering is less about metathesis, and more about “making space for insertion.”

**Hypothetical:**  $\emptyset \longrightarrow j / a \text{ \_\_ } u$   
/mausau/  $\rightarrow$  [majusaju] ‘logic’



## OBSERVATION #5

More generally:

- 1 With position-wise orderings, it's easy to describe functions where the number of possible insertion sites scales with the input size. However, the inserted material itself is bounded in size. A copy set of size  $n$  allows a max  $n - 1$  positions to be inserted between input segments.
- 2 For copy-wise ordered functions, the number of possible insertion sites is fixed by the size of the copy set (so 2 for reduplication, 3 for triplication), but the size of the inserted material scales with the size of the input.

## OBSERVATION #6

### Caveat

- I have not proved that it is **impossible** to model multiple metathesis/epenthesis with copy-wise ordered functions.
- A proof will depend on the logical language. I conjecture it is impossible with  $\text{FO}(<)$ . I am unsure about  $\text{MSO}(<)$ .



# CONCLUSION

- 1 Logical transductions in the style of Courcelle admit two kinds of orderings: position-wise and copy-wise.
- 2 While many processes can be modeled either way, certain phonological processes (epenthesis, metathesis) are more easily modeled position-wise, and certain morphological processes are more easily modeled copy-wise.
- 3 This supports the hypothesis that
  - **phonological** processes obey **position-wise** ordering.
  - **morphological** processes obey **copy-wise** ordering.
- 4 This analysis holds for other logical languages, such as Boolean Recursive Monadic Schemes (Chandlee and Jardine 2021).

# ACKNOWLEDGMENTS

I'd like to thank those who participated in the following at Stony Brook University:

- the Fall 2024 computational phonology seminar
- the Spring 2025 PhoRUM reading group
- the Spring 2025 Math Ling Recreation Group

**Thank you too!**

# REFERENCES I

- Albro, Dan. 2005. A large-scale, LPM-OT analysis of Malagasy. Doctoral dissertation, University of California, Los Angeles.
- Chandlee, Jane, and Adam Jardine. 2021. Computational universals in linguistic theory: Using recursive programs for phonological analysis. *Language* 93:485–519.
- Cohn, Abigail. 1989. Stress in Indonesian and bracketing paradoxes. *Natural language & linguistic theory* 7:167–216.
- Courcelle, Bruno. 1994. Monadic second-order definable graph transductions: a survey. *Theoretical Computer Science* 126:53–75.
- Courcelle, Bruno, and Joost Engelfriet. 2012. *Graph Structure and Monadic Second-Order Logic, a Language Theoretic Approach*. Cambridge University Press.

## REFERENCES II

- Heinz, Jeffrey. 2005. Description and analysis of surface patterns in Kwara'ae. In *Working Papers in Phonology*, 57–92. UCLA Working Papers.
- Kenstowicz, Michael, and Charles Kisseberth. 1979. *Generative Phonology*. Academic Press, Inc.
- Mooney, Kate. 2022. Phonology cannot transpose: evidence from Meto. *Phonology* 39:293–343.
- Odden, David. 2014. *Introducing Phonology*. 2nd ed. Cambridge University Press.
- O'Neill, Timothy. 2015. The phonology of betsimisaraka malagasy. Doctoral dissertation, University of Delaware.
- Takahashi, Chikako. 2019. No transposition in harmonic serialism. *Phonology* 36:695–726.