

Rule Interaction Conversion Operations

Eric Baković¹ and Lev Blumenfeld²

¹UC San Diego

²Carleton University

ebakovic@ucsd.edu ORCID: <https://orcid.org/0000-0002-2048-5135>

lev.blumenfeld@carleton.ca ORCID: <https://orcid.org/0000-0002-3335-7503>

Submitted: 30/06/2019; Accepted: 08/09/2019; Published online: 07/04/2020

Citation / Cómo citar este artículo: Eric Baković and Lev Blumenfeld (2019). *Rule interaction conversion operations*. *Loquens*, 6(2), e062, <https://doi.org/10.3989/loquens.2019.062>

ABSTRACT: Different types of interactions between pairs of phonological rules can be converted into one another using three formal operations that we discuss in this article. One of these conversion operations, rule re-ordering (here called swapping), is well-known; another, flipping, is a more recent finding (Hein et al., 2014). We introduce a third conversion operation that we call cropping. Formal relationships among the members of the set of rule interactions, expanded by cropping beyond the classical four (feeding, bleeding, counterfeeding, and counterbleeding) to include four more (mutual bleeding, seeding, counterseeding, and merger), are identified and clarified. We show that these conversion operations exhaustively delimit the set of possible pairwise rule interactions predicted by conjunctive rule ordering (Chomsky & Halle, 1968), and that each interaction is related to each of the others by the application of at most two conversion operations.

Keywords: theoretical phonology; rule ordering; conversion operations.

RESUMEN: Operaciones de conversión de interacciones de reglas.— En este artículo discutimos tres operaciones formales que pueden utilizarse para subsumir en uno solo diversos tipos de interacciones entre pares de reglas fonológicas. Una de estas operaciones de conversión, la reordenación de reglas (aquí denominada *swapping*), es bien conocida; otra, llamada aquí *flipping*, es un hallazgo más reciente (Hein et al., 2014). Introducimos una tercera operación de conversión a la que le hemos dado el nombre de *cropping*. Se identifican y clarifican las relaciones formales entre los tipos integrantes del conjunto de interacciones de reglas, que mediante el *cropping* van más allá de los cuatro clásicos (*feeding*, *bleeding*, *counterfeeding* y *counterbleeding*) y pasan a incluir otros cuatro tipos (*mutual bleeding*, *seeding*, *counterseeding* y *merger*). Mostramos cómo estas operaciones de conversión delimitan de manera exhaustiva el grupo de posibles interacciones entre pares de reglas que el orden conjuntivo de reglas predice (Chomsky & Halle, 1968) y que cada interacción se relaciona con cada una de las restantes mediante la aplicación de, como mucho, dos operaciones de conversión.

Palabras clave: fonología teórica; ordenación de las reglas; operaciones de conversión.

1. INTRODUCTION

What is the set of possible pairwise phonological rule interactions, and how are they related? By construing rule interaction relationships in terms of formal conversion operations, we find that the total number of pairwise interactions is eight. Conversion from any one of these

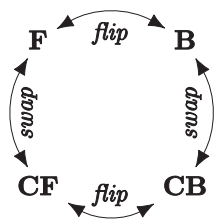
interactions to another can be accomplished by applying at most two of the three conversion operations, revealing a great deal about the relationships among the different types of interactions predicted by conjunctive rule ordering (Chomsky & Halle, 1968).

To start, we already know what it takes to formally convert one of the four ‘classical’ rule interactions —feeding,

bleeding, counterfeeding, and counterbleeding—into one another. Most familiarly, feeding and bleeding can be converted into their counter-counterparts, and vice-versa, via rule re-ordering, relabeled here as a conversion operation we call swapping (Chafe, 1968; Kiparsky, 1968; Koutsoudas et al., 1974). If P precedes and feeds, Q then swapping the order of P and Q such that Q precedes P means that P will counterfeed Q , and vice-versa. Likewise, if P precedes and bleeds R , then swapping the order of P and R such that R precedes P means that P will counterbleed R , and vice-versa.

More recently, Hein et al. (2014) have shown how feeding and bleeding, or counterfeeding and counterbleeding, can be converted into one another via an operation that they call flipping. Given a rule P that precedes and feeds (or follows and counterfeeds) another rule Q , flipping the target and the change of P means that P will bleed (or counterbleed) Q , and vice-versa¹. Combining these two operations completes the circle: feeding (F) and bleeding (B) become counterbleeding (CB) and counterfeeding (CF), respectively and vice-versa, via both flipping (*flip*) and swapping (*swap*), applied to each interaction in either order.

Figure 1: Flipping and swapping among classical rule interactions.



We discuss these two rule interaction conversion operations in more detail below, and then introduce a new conversion operation that we dub cropping. The set of possible pairwise rule interactions is thereby expanded to include distinctions beyond the classical, and conversion relations among all interactions are identified and clarified.

Given the formal nature of the topic at hand, the discussion will rather necessarily abstract away from the empirical basis for the various rule interactions. We endeavor to temper this abstraction throughout with relevant examples; these are simplified for expository purposes, and possibly overly so in some cases. In order to make clear that we are (over)simplifying, we write that each example is ‘based on’ the relevant language example.

Our deep dive into the predicted consequences of conjunctive rule ordering is inspired by Prince (2007), who notes (p. 33) that “a theory is the totality of its consequences ... Once formulated, a theory has broken definitively with intuition and belief. We are stuck with

its consequences whether we like them or not, anticipate them or not, and we must develop techniques to find them.” The techniques that we use in this article to identify and relate rule interactions elucidate key predictions of conjunctive rule ordering, well beyond and more precisely than what has been achieved since discussions of this theory began.

2. INTERACTIONS AND RELATIONS

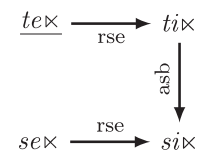
2.1. Feeding and provision

Consider the following example of feeding, based on a Finnish example (Kiparsky, 1973, 1993). The raising rule in (1a) changes word-final e to i , and the assibilation rule in (1b) changes ti sequences to si sequences². The feeding interaction is apparent in (1c) only raising is applicable to *vete*, resulting in *veti*; assibilation is now newly applicable, yielding *vesi*.

- (1) Raising feeds assibilation
 - a. **Raising** (‘rse’): $e \rightarrow i / _ \#$
 $kiele \xrightarrow{\text{rse}} kieli$ ‘tongue’
 - b. **Assibilation** (‘asb’): $t \rightarrow s / _ i$
 $tilati \xrightarrow{\text{asb}} tilasi$ ‘ordered’
 - c. **Feeding interaction:**
 $vete \xrightarrow{\text{rse}} veti \xrightarrow{\text{asb}} vesi$ ‘water’

In previous work (Baković & Blumenfeld, 2017, 2018a), we show how a classical feeding interaction like this one is composed of two more basic relations between the two rules. This decomposition is illustrated in Figure 2, with attention limited³ to a set of minimal substrings standing in for the equivalence classes required to reveal the more basic relations⁴.

Figure 2: rse feeds asb



Because these types of figures and the relations they illustrate are unfamiliar, we will take some time to explain Figure 2 in such a way that later figures may be read more easily.

¹ Note that the formal operation of flipping is distinct from the historical phenomenon of RULE INVERSION (Vennemann, 1972), which involves “reversal of the input and output of a rule and complementation of the environment” (McCarthy, 1991, p. 194, emphasis added).

² In the actual Finnish example, Kiparsky (1973, 1993) argues, the application of assibilation is limited to derived environments, effectively meaning that it applies only when its context is created (a) by the concatenation of morphemes or (b) by another phonological rule (like raising). This detail is set aside here.

³ This limitation abstracts away from more complex situations involving several loci of application of a rule in a single string, but these complications are not germane to the present discussion.

⁴ In string representations we adopt the convention of e.g. Heinz (2018) of distinguishing word-final boundaries (‘X’) from word-initial boundaries (‘X’), both represented as ‘#’ in rules.

The earlier-ordered rule involved in the interaction — in this case, the feeding raising rule ‘rse’ — is displayed horizontally (input $te\ltimes$ on the left, output $ti\ltimes$ on the right, and a ‘rse’-labeled arrow between them), and the later-ordered rule — the fed assibilation rule ‘asb’ — is displayed vertically (input $ti\ltimes$ above, output $si\ltimes$ below, and an ‘asb’-labeled arrow between them). To more fully represent the extensions of the rules involved in the interaction, we explicitly indicate that $si\ltimes$ is also an independently-predicted nonvacuous output of raising by adding the relevant input $se\ltimes$ to its left and a ‘rse’-labeled arrow between them. Finally, to highlight the starting point of the path through the figure that intuitively corresponds to the feeding interaction itself, the initial input of raising is underlined: $te\ltimes$.

The first, perhaps more obvious relation is that raising provides inputs to assibilation (‘rse i-provides asb’), because for at least one nonvacuous ⟨input, output⟩ pair of raising, ⟨ $te\ltimes$, $ti\ltimes$ ⟩, the output, $ti\ltimes$, is a possible input to assibilation, but the input, $te\ltimes$, is not. The second relation is that raising also provides outputs to assibilation (‘rse o-provides asb’), because for at least one nonvacuous ⟨input, output⟩ pair of raising, ⟨ $se\ltimes$, $si\ltimes$ ⟩, the output, $si\ltimes$, is a possible output of assibilation, but the input, $se\ltimes$, is not.

This output-provision relation is less obvious, but crucial. It ensures that the application of assibilation to $ti\ltimes$ does not change the fact that the resulting form, $si\ltimes$, is also a possible direct result of raising, albeit from a different input. If this output-provision relation did not hold, the application of assibilation to the output of raising would result in a form that does not look like raising should have applied to it. In such a scenario, the application of raising would not be surface apparent, a well-known property of counterbleeding interactions (McCarthy, 1999) — and also of self-destructive feeding interactions (Baković, 2007), of which this hypothetical example would be one, as discussed in §5.2 below.

Another way to understand these two relations is in terms of the intersections of the string sets that constitute the two rules’ inputs and outputs (Baković, 2013b). Let’s call the set of input strings to raising rse_i , the set of output strings of raising rse_o , the set of input strings to assibilation asb_i , and the set of output strings of assibilation asb_o . Limiting our attention again to minimal relevant substrings, we have the string set intersections in (2).

- (2) String set intersections of rse_x and asb_y

$$\begin{aligned} rse_i \cap asb_i &= \left\{ \begin{array}{l} te\ltimes \\ se\ltimes \end{array} \right\} \cap \{ti\ltimes\} = \emptyset \\ rse_i \cap asb_o &= \left\{ \begin{array}{l} te\ltimes \\ se\ltimes \end{array} \right\} \cap \{si\ltimes\} = \emptyset \\ rse_o \cap asb_i &= \left\{ \begin{array}{l} ti\ltimes \\ si\ltimes \end{array} \right\} \cap \{ti\ltimes\} = \{ti\ltimes\} \\ rse_o \cap asb_o &= \left\{ \begin{array}{l} ti\ltimes \\ si\ltimes \end{array} \right\} \cap \{si\ltimes\} = \{si\ltimes\} \end{aligned}$$

Raising thus nonvacuously produces both outputs that are inputs to assibilation and outputs that are outputs of assibilation; that is, raising both i-provides and o-provides assibilation.

2.2. Bleeding and removal

Consider now the following example of bleeding, based on a Lamba example (Doke, 1938; Kenstowicz & Kisseberth, 1979)⁵. Both lowering and palatalization are applicable to *kosika*, but once lowering applies, yielding *koseka*, palatalization is no longer applicable.

- (3) Lowering bleeds palatalization

- a. **Lowering** (‘low’):

$$V \rightarrow [-\text{high}] / \left[\begin{array}{l} -\text{high} \\ -\text{low} \end{array} \right] C_0 _ \#$$

ponika $\xrightarrow{\text{low}}$ *poneka* ‘it falls’

- b. **Palatalization** (‘pal’): $s \rightarrow \acute{o} / _ i$

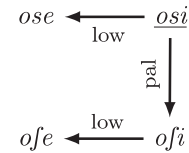
$$fisika \xrightarrow{\text{pal}} f\acute{i}sika \text{ ‘it hides’}$$

- c. **Interaction**: ‘it is strong’

$$kosika \xrightarrow{\text{low}} koseka \xrightarrow{\text{pal}} koseka$$

A classical bleeding interaction like this one is also composed of two more basic relations between the two rules (Baković & Blumenfeld, 2017, 2018a). This is illustrated in Figure 3, again with attention focused on just the minimal substrings relevant to the interaction.

FIGURE 3: low bleeds pal.



Again, the earlier-ordered rule involved in the interaction — the bleeding lowering rule ‘low’ — is displayed horizontally (input *osi* on the right, output *ose* on the left, ‘low’-labeled arrow between them), and the later-ordered rule — the bled palatalization rule ‘pal’ — is displayed vertically (input *osi* above, output *ofi* below, ‘pal’-labeled arrow between them). And again, to more fully represent the extensions of the rules involved in the interaction, we explicitly indicate that *ofi* is a predicted nonvacuous input to lowering by adding the relevant output *ofe* to its left and a ‘low’-labeled arrow

⁵ In the actual Lamba example, palatalization also changes *k* to *tʃ*. Charles Reiss (p.c.) suggests that there must be two rules, given that {*s*, *k*} do not form a natural class. Regardless of whether there is one rule or two, lowering also bleeds palatalization of *k*: *sekika* $\xrightarrow{\text{low}}$ *sekeka* $\xrightarrow{\text{pal}}$ *sekeka* ‘it laughs at’.

between them⁶. And finally, the starting point of the path through the figure that intuitively corresponds to the bleeding interaction itself is highlighted by underlining the initial input of lowering: *osi*.

The first, more obvious relation is that lowering removes inputs from palatalization ('low i-removes pal'), because for at least one nonvacuous ⟨input, output⟩ pair of lowering, ⟨*osi*, *ose*⟩, the input, *osi*, is a possible input to palatalization but the output, *ose*, is not. The second relation is that lowering also removes outputs from palatalization ('low o-removes pal'), because for at least one nonvacuous ⟨input, output⟩ pair of lowering, ⟨*ofi*, *ofe*⟩, the input, *ofi*, is a possible output of palatalization but the output, *ofe*, is not.

Like the output-provision relation of feeding, the less-than-obvious output-removal relation of bleeding is crucial. Its presence ensures that the application of palatalization to *osi* does not change the fact that lowering is still applicable to the resulting form, *ofi*. If this output-removal relation did not hold, then the output of palatalization would no longer be eligible for lowering, and in that case palatalization would bleed lowering just as lowering bleeds palatalization. This is mutual bleeding, as discussed in §5.1 below.

And again, from the point of view of string set intersection:

- (4) String set intersections of rse_x and pal_y

$$low_i \cap pal_i = \left\{ \frac{osi \times}{ofi \times} \right\} \cap \{osi \times\} = \{osi \times\},$$

$$low_i \cap pal_o = \left\{ \frac{osi \times}{ofi \times} \right\} \cap \{ofi \times\} = \{ofi \times\}$$

$$low_o \cap pal_i = \left\{ \frac{ose \times}{ofe \times} \right\} \cap \{osi \times\} = \emptyset$$

$$low_o \cap pal_o = \left\{ \frac{ose \times}{ofe \times} \right\} \cap \{ofi \times\} = \emptyset$$

Lowering thus nonvacuously changes both inputs that are inputs to palatalization and inputs that are outputs of palatalization to outputs that are neither inputs nor outputs of palatalization; that is, lowering both i-removes and o-removes palatalization.

2.3. Summary

The classical feeding and bleeding rule interactions can each be decomposed into two more basic relations between rules, provision and removal of inputs and outputs. Provision and removal are respectively defined more explicitly in (5) and (6).

⁶ In our terms, an input-output mapping ⟨in, out⟩ is 'predicted' by a rule X if $X(in) = out$, and is not predicted otherwise. Given that lowering does not specify anything about the consonants that intervene between its trigger and target, ⟨*ofi*, *ofe*⟩ is predicted by lowering just as ⟨*osi*, *ose*⟩ is; it matters not at all if the *s*-*j* alternation in Lambda is allophonic or the neutralization of an underlying contrast.

- (5) Provision of inputs and outputs (feeding)
- P **input-provides** Q if \exists forms a, b s.t. $P(a) = b$ and Q applies to b but not a .
 - P **output-provides** Q if \exists forms a, b s.t. $P(a) = b$ and \exists a form c s.t. $Q(c) = b$ but \nexists a form d s.t. $Q(d) = a$.
- (6) Removal of inputs and outputs (bleeding)
- P **input-removes** R if \exists forms a, b s.t. $P(a) = b$ and R applies to a but not b .
 - P **output-removes** R if \exists forms a, b s.t. $P(a) = b$ and \exists a form c s.t. $R(c) = a$ but \nexists a form d s.t. $R(d) = b$.

In the case of feeding, the earlier-ordered feeding rule both i-provides and o-provides the later-ordered fed rule; in the case of bleeding, the earlier-ordered bleeding rule both i-removes and o-removes the later-ordered bled rule.

Generalizing somewhat from the relatively specific (albeit simplified) examples discussed above, let us imagine three rules of the following schematic and possibly familiar form.

- (7) a. $P = A \rightarrow B/C_D$
 b. $Q = E \rightarrow f/BD_$
 c. $R = E \rightarrow f/ AD_$

Given a starter input cade, P feeds Q and bleeds R , as shown in Figure 4 and Figure 5.

FIGURE 4: P feeds Q .

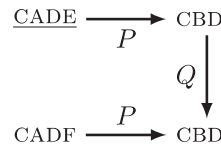
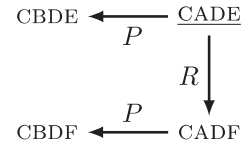


FIGURE 5: P bleeds R .



These figures are intended as schematic visual representations of crucial formal properties of each interaction. Some properties depend on rule order: the distinction between feeding and counterfeeding depends on the order of P and Q , and the distinction between bleeding and counterbleeding depends on the order of P and R . Order-dependent properties are illustrated by the dimensional orientation of the arrows in the figures: the earlier-ordered rule is displayed horizontally, while the later-ordered rule is displayed vertically. This dimensional distinction will be manipulated in §3 below to represent the result of the swapping operation.

Other properties are independent of order, and are encoded in the topology of the arrows. The structure $\rightarrow. \rightarrow.$ corresponds to both feeding and counterfeeding (e.g. $CADE \rightarrow CBDE \rightarrow CBDF \leftarrow CADF$), and the structure $\leftarrow. \rightarrow.$ corresponds to both bleeding and counterbleeding (e.g. $CBDE \leftarrow CADE \rightarrow CADF \rightarrow CBDF$). Mnemonically, when the two outermost arrows point inward, we have in-flow (feeding and counterfeeding); when those two arrows point outward, we have out-flow (bleeding and counterbleeding).

3. SWAPPING AS (CLOCKWISE) ROTATION

Given the convention in our figures of illustrating the application of the earlier-ordered rule horizontally and the application of the later-ordered rule vertically, we can rotate the figures 90 degrees, semi-arbitrarily clockwise, to represent swapping, converting feeding (Figure 4) into counterfeeding (Figure 6) and bleeding (Figure 5) into counterbleeding (Figure 7). As noted in the summary in §2.3 further above, rotating the figures preserves the topology of the arrows and thus the order-independent properties of these rule interactions.

FIGURE 6: P cntr-feeds Q .

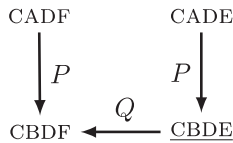
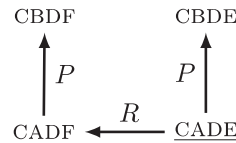


FIGURE 7: P cntr-bleeds R .



Note that we continue to follow the convention of underlining the starting point of the path through the figure that intuitively corresponds to the type of interaction in each case. Counterfeeding is when P fails to feed Q , meaning that Q maps $cbde$ to $cbdf$ before P gets a chance to create more $cbde$ outputs to also be inputs to Q . This makes the application of Q non-surface-true, a characteristic of counterfeeding (McCarthy, 1999) as well as of other blocking interactions that are not predicted by conjunctive rule ordering (non-derived environment blocking, do-something-except-when blocking, and disjunctive blocking; see Baković, 2011, 2013a). Counterbleeding is when P fails to bleed R , meaning that R maps $cade$ to $cadf$ in time for P to then map $cadf$ to $cbdf$. This makes the application of R non-surface-apparent, a characteristic of counterbleeding (McCarthy, 1999) and of self-destructive feeding (Baković, 2007), on which see §5.2 below.

4. FLIPPING AS ARROW REVERSAL

Recall that Hein et al.'s (2014) flipping operation involves taking the rule we've been referring to as P — the earlier-ordered rule in a feeding or bleeding interaction, and the later-ordered rule in a counterfeeding or counterbleeding interaction — and exchanging P 's target for P 's change and vice-versa. So, given that P is $A \rightarrow B / C _ _ D$, changing cad to cbd , then $flip(P)$ is $B \rightarrow A / C _ _ D$, which now changes cbd to cad . With Q and R still applicable and inapplicable to the same string sets in each case, as shown in (8), feeding becomes bleeding and bleeding becomes feeding, and likewise for their counter-counterparts.

- (8) a. $flip(P) = B \rightarrow A / C _ _ D$ cf. (7a)
 b. $Q = E \rightarrow F / BD _ _$ = (7b)
 c. $R = E \rightarrow F / AD _ _$ = (7c)

This is illustrated with simple reversal of P 's arrows in the illustrative figures presented thus far, converting Figure 4, Figure 5, Figure 6 and Figure 7 into Figure 8, Figure 9, Figure 10, and Figure 11, pairwise respectively.

FIGURE 8: $flip(P)$ bleeds Q .

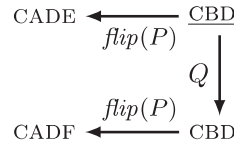


FIGURE 9: $flip(P)$ feeds R .

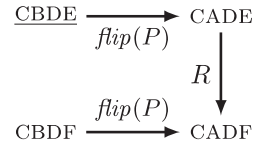


FIGURE 10: $flip(P)$ cntr-bleeds Q .

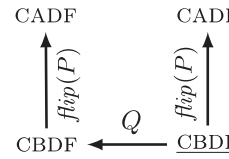
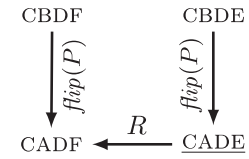


FIGURE 11: $flip(P)$ cntr-feeds R .



Note that flipping changes the topology of the arrows. Flipping P amounts to changing the direction of the two outer arrows, which converts $\rightarrow . \rightarrow . \leftarrow$ into $\leftarrow . \rightarrow . \rightarrow$ changing in-flow to out-flow, and vice-versa. On the other hand, even though e.g. Figure 4 (P feeds Q) and Figure 9 ($flip(P)$ feeds R) have different strings in the underlined starting point position, they both have the same arrow topology — in other words, the *de novo* feeding interaction in Figure 4 and the feeding interaction derived via flipping in Figure 9 are formally identical in terms of input and output provision, or how the rules' input and output string sets intersect.

Note also that flipping the inner Q -/ R - arrow preserves the essential arrow topology, and thus has no formal effect on the interaction. Consider first feeding and counterfeeding: $\rightarrow . \rightarrow . \leftarrow$, when Q -flipped to $\rightarrow . \leftarrow . \leftarrow$ is essentially unchanged; in-flow remains in-flow. In the Finnish-based feeding example in (1), assimilation ($Q = t \rightarrow s / _ _ i$) can be flipped to become 'desibilization' ($flip(Q) = s \rightarrow t / _ _ i$); raising ($=P$) still i -provides and o -provides the i in the context of this flipped rule. Now consider bleeding and counterbleeding: $\leftarrow . \rightarrow . \rightarrow$ when R -flipped to $\leftarrow . \leftarrow . \rightarrow$, is again essentially unchanged; out-flow remains out-flow. In the Lamba-based bleeding example in (3), palatalization ($R = s \rightarrow \int / _ _ i$) can be flipped to become depalatalization ($flip(R) = \int \rightarrow s / _ _ i$); lowering ($=P$) still i -removes and o -removes the i in the context of this flipped rule.

5. CROPPING AS ARROW EXCISION/ADDITION

Explicit recognition of the provision and removal of outputs is critical for distinguishing the classical from other pairwise rule interaction types (Baković & Blumenfeld, 2017, 2018a), to which we now turn.

5.1. Mutual bleeding

Consider first the case of mutual bleeding in (9), based on a Russian example (Kenstowicz & Kisseberth, 1979). Bleeding is mutual here because if l -drop were to apply

first, it would be the rule that bleeds and deletion would be the rule that is bled: $metl \xrightarrow{l-d} met \xrightarrow{del} *met$.

(9) Mutual bleeding of deletion and *l*-drop

a. **Deletion** ('del'):

$$\begin{bmatrix} +cor \\ -son \\ -cont \end{bmatrix} \rightarrow \emptyset / _ l$$

$$metla \xrightarrow{del} mela \text{ 'she swept'}$$

b. ***l*-drop** ('l-d'): $l \rightarrow \emptyset / C _ \#$

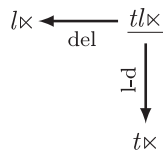
$$nesl \xrightarrow{l-d} nes \text{ 'he carried'}$$

c. **Interaction:**

$$metl \xrightarrow{del} mel \xrightarrow{l-d} mel \text{ 'he swept'}$$

The illustrative figure for this case of mutual bleeding thus looks like Figure 12, with deletion and *l*-drop mutually removing inputs from each other, and with no output-removal.

FIGURE 12: Mutual bleeding of del and l-d.



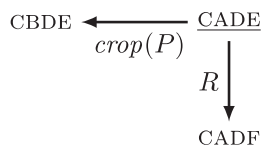
It is crucial that there is no deletion arrow originating in *l*⊗, and no *l*-drop arrow originating in *t*⊗: the absence of an arrow in this figure is as significant as its presence. The topology of Figure 12 is thus $\leftarrow \cdot \rightarrow$, without a third arrow.

Generalizing somewhat to our schematic rules and interactions, imagine that *P* is further specified such that it more specifically changes cade to cbde, and call this new rule *crop(P)*.⁷ (*R* is repeated for ease of reference below.)

- (10) a. $crop(P) = A \rightarrow B / C _ DE$ cf. (7a)
b. $R = E \rightarrow F / AD _$ = (7c)

This constitutes a 'cropping' of *P* in that the extension of *crop(P)* is a narrower, proper subset of *P*'s extension. Most critically, *crop(P)* doesn't apply to the output cadf of *R*, which the original *P* does apply to. The effect of narrowing *P*'s extension in this way is to excise the rightmost arrow of the (counter)bleeding topology $\leftarrow \cdot \rightarrow$, converting it to mutual bleeding. $\leftarrow \cdot \rightarrow$ The generalized figure for mutual bleeding is thus Figure 13: just like *P* bleeds *R* in Figure 5, except that the output-removal arrow of *P* has been excised.

FIGURE 13: Mutual bleeding of *crop(P)* & *R*.

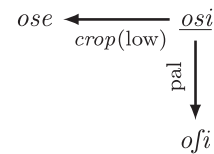


⁷ Many thanks to Karen Shelby (p.c.) for suggesting the term 'cropping'.

The cropping operation is intended to encompass both excision and addition of an arrow, thus mimicking the inherent symmetry of the swapping and flipping operations.⁸ Thus classical bleeding can be converted via cropping to mutual bleeding by excision of the rightmost, o-removal arrow of the topology, and mutual bleeding can likewise be converted via cropping to classical bleeding by addition of this o-removal arrow.

To see how cropping works with more substantive examples, let's first see what it takes to convert the example based on Lamba from bleeding to mutual bleeding. Excising the o-removal arrow of the lowering rule from Figure 3 results in Figure 14.

FIGURE 14: Mutual bleeding of crop(low) & pal.



In order to achieve this result, the context of lowering must be changed such that the last of the zero or more consonants intervening between the trigger mid vowel and the target *i* is not *j*.⁹ (Recall now the relevance of the discussion in fn. 6 about whether the *s*~*j* alternation in Lamba is allophonic or the neutralization of an underlying contrast.) This condition excludes rather than includes a natural class, and so must be stated negatively:

(11) Mutual bleeding of *crop(low)* & pal

a. **Cropped lowering** ('crop(low)'):

$$V \rightarrow [-high] / \begin{bmatrix} -high \\ -low \end{bmatrix} C_0 _ \#;$$

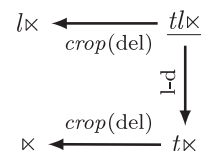
$$last(C_0) \neq j$$

b. **Palatalization** ('pal'): $s \rightarrow \int / _ i$

Cropped lowering and palatalization mutually bleed each other: if cropped lowering applies first to *osi*, yielding *ose*, then palatalization is no longer applicable, and if palatalization applies first, yielding *ofi*, then cropped lowering is no longer applicable.

Now let's see how to convert the example based on Russian from mutual bleeding to bleeding. Adding an o-removal arrow to the deletion rule in Figure 12 results in Figure 15.

FIGURE 15: *crop(del)* bleeds l-d.



⁸ We refer to both sides of this operation as 'cropping' although the analogy is strained by arrow addition, which is technically 'un-cropping'.

⁹ The formulation in 11a assumes a function *last* that returns the last member of a string. In this case the string is $C_0 = C_1C_2 \dots C_{n-1}C_n$, so $last(C_0) = C_n$, and so the condition on the rule is that $C_n \neq j$.

The minimal change necessary to achieve this result is for the context of deletion to become a disjunctive one, such that coronal stops delete before *l* or at the end of the word:

- (12) Cropped deletion bleeds *l*-drop
a. **Cropped deletion** ('*crop*(del)'):

$$\left[\begin{array}{c} +\text{cor} \\ -\text{son} \\ -\text{cont} \end{array} \right] \rightarrow \emptyset / _ \left\{ \begin{array}{c} l \\ \# \end{array} \right\}$$

- b. ***l*-drop** ('*l*-d'): $l \rightarrow \emptyset / C_ \#$

Cropped deletion bleeds *l*-drop, but not vice-versa: if cropped deletion applies first to *tl*×, yielding *l*×, then *l*-drop is no longer applicable, but if *l*-drop applies first, yielding *t*×, then cropped deletion is still applicable, yielding the maximally deleted ×.

The contextual disjunction technically means that *l*-drop 'transfuses' cropped deletion, in the sense defined in an unpublished paper by Don Churma entitled 'Rule interactions' and cited by Zwicky (1987, p. 93)¹⁰ "[I]f one rule transfuses another, the string to which the second rule applies is different from what it would be if the first rule didn't apply — either because the first rule removes some material to which the second could apply but also supplies new places for the second rule to apply in, or because the first rule changes one string to which the second is applicable into a different string to which the second is applicable[.]" We discuss transfusion and similar complex interactions in Baković & Blumenfeld, in prep.

5.2. Self-destructive feeding ('seeding')

Consider now the case of self-destructive feeding (Baković, 2007) shown in (13), based on a Turkish example. The feeding is 'self-destructive' because applying deletion to the output of epenthesis results in a form that is itself not a possible output of epenthesis. Following Baković & Blumenfeld (2018b), we henceforth refer to this interaction type as seeding.

- (13) Epenthesis seeds deletion

- a. **Epenthesis** ('*epn*'):

$$\emptyset \rightarrow i / C_ \#$$

$$ipn \xrightarrow{\text{ep}} ipin \text{ 'your rope'}$$

- b. **Deletion** ('*del*'): $k \rightarrow \emptyset / V_ V$

$$bebeki \xrightarrow{\text{del}} bebei \text{ 'baby (acc)'}$$

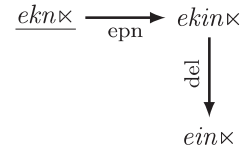
- c. **Interaction**: 'your baby'

$$bebekn \xrightarrow{\text{enn}} bebekin \xrightarrow{\text{del}} bebein$$

Epenthesis *i*-provides deletion: deletion is inapplicable to the input to epenthesis, *bebekn*, but is applicable to its output, *bebekin*. Unlike classical feeding, however, epenthesis does not also *o*-provide deletion: there is no

application of epenthesis that leads directly to the output *bebein*. The illustrative figure for this case of seeding is thus Figure 16, with epenthesis *i*-providing but not *o*-providing deletion. Furthermore, since the application of deletion results in a form that is not in the set of possible outputs of epenthesis, deletion in turn *o*-removes epenthesis. (Seeding shares this feature in common with counterbleeding; see §6.)

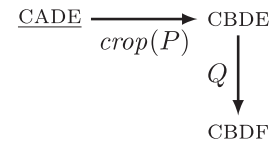
FIGURE 16: epn seeds del.



The topology of seeding, $\rightarrow . \rightarrow$, is distinct from other topologies we've seen thus far. But much like mutual bleeding, there are only two arrows; the absence of a deletion arrow originating at *ekn*× and of an epenthesis arrow ending at *ein*× in Figure 16 is crucial.

Generalizing somewhat again to our schematic rules and interactions, recall *crop*(*P*) in (10a), $A \rightarrow B / C _ _$ de, compared to the original *P* in (7a), $A \rightarrow B / C _ _ D$. In this case, what's relevant about the difference between *P* and *crop*(*P*) is that *crop*(*P*) doesn't apply to an input that would result in the output *c*bd of *Q*. The effect of narrowing *P*'s extension in this way is to excise the rightmost arrow of the (counter)feeding topology, $\rightarrow . \rightarrow . \leftarrow$ converting it into seeding $\rightarrow . \rightarrow$. The generalized figure for seeding is thus Figure 17: just like *P* feeds *Q* in Figure 4, except that the *o*-provision arrow of *P* has been excised.

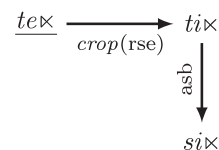
FIGURE 17: *crop*(*P*) seeds *Q*.



Again, the cropping operation is intended to be symmetrical, encompassing both excision and addition of an arrow. Thus classical feeding can be converted by cropping to seeding by excision of the rightmost, *o*-provision arrow of the topology, and seeding can likewise be converted by cropping to classical feeding by addition of this *o*-provision arrow.

To see how cropping in the feeding context works with more substantive examples, let's first see what it takes to convert the example based on Finnish from feeding to seeding. Excising the *o*-provision arrow of the raising rule from Figure 2 results in Figure 18.

FIGURE 18: *crop*(*rse*) seeds asb.



¹⁰ Many thanks to Arnold Zwicky for drawing our attention to this work, unfindable though it may be.

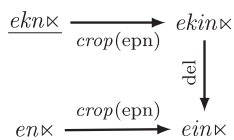
In order to achieve this result, the context of raising must be changed such that there not be an *s* to the left of the raising target *e*. Because this condition excludes rather than includes a natural class, it must be stated negatively:

- (14) Cropped raising seeds assibilation
 a. **Cropped raising** ('crop(rse)'): $e \rightarrow i / C _ \#; C \neq s$
 b. **Assibilation** ('asb'): $t \rightarrow s / _ i$

Cropped raising seeds rather than feeds assibilation, because assibilation destroys the conditions that were necessary for the application of cropped raising in the first place.

Now let's see how to convert the example based on Turkish from seeding to feeding. Adding an o-provision arrow to the epenthesis rule in Figure 16 results in Figure 19.

FIGURE 19: *crop(epn)* feeds *del*.



This result can be achieved by changing the context of epenthesis such that *i* is inserted before any word-final consonant, without regard for what may appear to its left:

- (15) Cropped epenthesis feeds deletion
 a. **Cropped epenthesis** ('crop(ep)'): $\emptyset \rightarrow i / _ C \#$
 b. **Deletion** ('del'): $k \rightarrow \emptyset / V _ V$

Cropped epenthesis feeds deletion, but deletion does not in turn destroy the context that enabled the application of cropped epenthesis: the form resulting from the feeding interaction is itself a possible direct output of cropped epenthesis.

5.3. More swapping

Just as with classical feeding and bleeding, the rules involved in mutual bleeding and seeding interactions can be swapped — in other words, Figure 13 and Figure 17 can be rotated (clockwise) 90 degrees. The results of these rotations are shown in Figure 20 and Figure 21.

FIGURE 20: Mut. bld. of *R* & *crop(P)*.

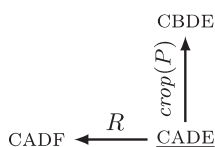
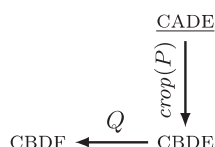


FIGURE 21: *crop(P)* counterseeds *Q*.



In the case of Figure 20, the result is simply that *R* is now the bleeding rule and *P* is now the bled rule. In the case of Figure 21, on the other hand, the result is a counterseeding interaction. Such an interaction is shown in (16), based on an example in Lomongo. If deletion were to apply first, deletion would seed gliding: $obina \xrightarrow{\text{del}} oina \xrightarrow{\text{gld}} *wina$.

- (16) Deletion counterseeds gliding:
 a. **Gliding** ('gld'): $[-\text{cons}] \rightarrow [-\text{syll}] / _ V$

$oisa \xrightarrow{\text{gld}} wisa$ 'you hide'

- b. **Deletion** ('del'):

$$\begin{bmatrix} -\text{son} \\ +\text{voi} \end{bmatrix} \rightarrow \emptyset / V _$$

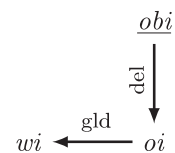
$babina \xrightarrow{\text{del}} baina$ 'they dance'

- c. **Interaction:**

$obina \not\xrightarrow{\text{gld}} obina \xrightarrow{\text{del}} oina$ 'you dance'

Deletion i-provides but does not also o-provide gliding: there is no input to which deletion can apply that leads directly to the output *wina* that would result from deletion seeding gliding. The illustrative figure for this case of counterseeding is thus Figure 22, with deletion i-providing but not o-providing gliding, and with gliding consequently o-removing deletion.

FIGURE 22: *del* counterseeds *gld*.

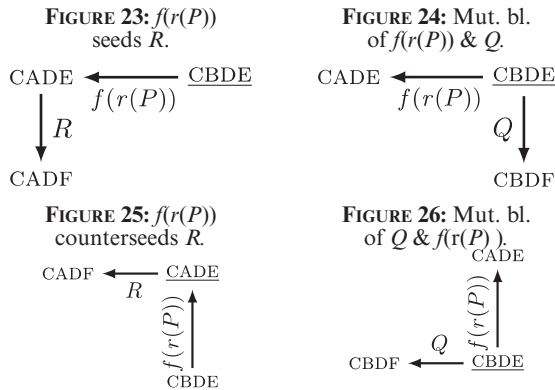


The interaction in Lomongo has been described simply as counterfeeding in e.g. Baković (2011, 2013a), but proper attention to the classically-ignored output relations reveals that it is formally distinct from counterfeeding. In order for this interaction to be converted to counterfeeding, an o-provision *crop(del)*-arrow would need to be added from input *wbi* to output *wi* in Figure 22 above, thus generalizing deletion such that it applies when the preceding segment is a vowel or a glide ($=[-\text{Cons}]$), not just a vowel ($=V/[+\text{syll}]$).

Finally, note once again that our illustrative figures highlight the difference between order-dependent and order-independent properties. Swapping (as figure rotation) converts between seeding and counterseeding as order-dependent categories; cropping (as arrow excision/addition) converts between (counter) feeding and (counter)seeding as order-independent categories.

5.4. More flipping

In the same way that flipping P converts between (counter)feeding and (counter)bleeding, flipping $crop(P)$ converts between mutual bleeding and (counter)seeding. Figure 13, Figure 17, Figure 20 and Figure 21 thus become Figure 23, Figure 24, Figure 25 and Figure 26, pairwise respectively.

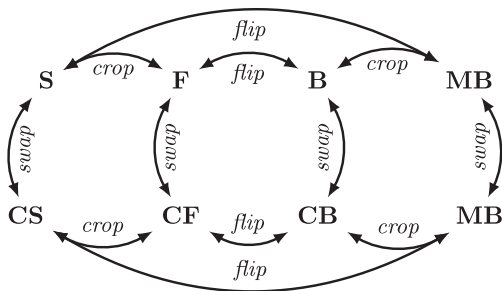


The orientations of these figures differ from those of their unflipped originals, but the results are the same three interaction types: seeding, mutual bleeding, and counterseeding.

6. INTERIM SUMMARY

By way of summary we offer the following diagram, building on the initial diagram offered in Figure 1. The classical interactions are in the center circle, related to each other via *flip* and *swap*; *crop* relates these classical interactions to mutual bleeding and (counter)seeding, which are also related to each other via *flip* and *swap*.

FIGURE 27: Flipping, swapping and cropping among order-dependent interactions.



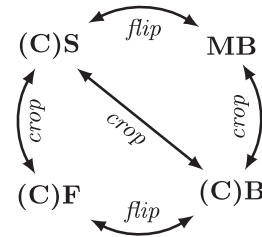
Note that (counter)bleeding and (counter)seeding, when considered in order-independent topological form (17), are directly relatable to each other via *crop* by excising the leftmost out-flow arrow of (counter)bleeding or by adding such an arrow to (counter)seeding.¹¹

¹¹ Further consequences of this relationship between counterbleeding and seeding are explored in more detail in Baković & Blumenfeld (2018b).

- (17) a. (Counter)bleeding: $\leftarrow . \rightarrow . \rightarrow$
b. (Counter)seeding: $\rightarrow . \rightarrow$

Focusing on just order-independent properties of interactions, then, conversions via *flip* and *crop* alone can be summarized as follows.

FIGURE 28: Flipping and cropping among order-independent interactions.



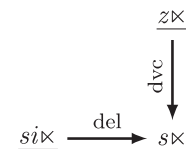
7. MERGER

The final topological arrow configuration is $\rightarrow . \leftarrow .$. This represents merger. Consider the following hypothetical pair of rules:

- (18) a. **Devoicing:**
[- son] @ [- voi] / $__ \#$
b. **Deletion:**
 $i \rightarrow \emptyset$ / [- voi] $__$

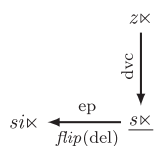
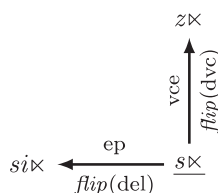
Neither feeds or bleeds the other, classically, mutually, self-destructively, or otherwise. But in our terms, the rules mutually output-provide each other: an output ending in $s\bowtie$ could result from the application of either rule, from input $z\bowtie$ by devoicing or from input $si\bowtie$ by deletion.

FIGURE 29: Merger.

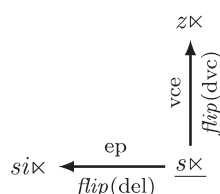


Our expectation is that this configuration should be relatable to other types of interaction via the *flip* and *crop* operations, and indeed it is.

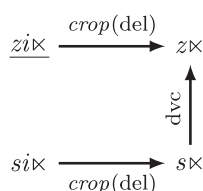
Flipping either of the arrows results in one of two seeding configurations. If deletion is flipped to epenthesis, then devoicing counterseeds epenthesis (Figure 30): epenthesis causes the insertion of a vowel, which makes the consonant non-final, thus destroying devoicing's potential triggering environment. Conversely, if devoicing is flipped to voicing, then deletion seeds voicing (Figure 31): deletion causes voicing, but voicing makes the consonant voiced, thus destroying deletion's triggering environment.

FIGURE 30:
Merger \Rightarrow cntr-seeding.**FIGURE 31:**
Merger \Rightarrow seeding.

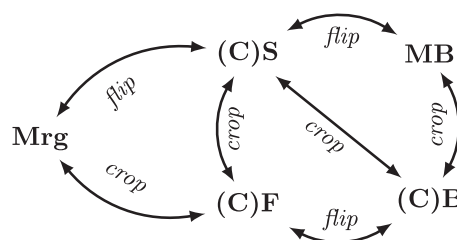
Flipping both arrows results in mutual bleeding. Devoicing becomes final voicing, and deletion becomes *i*-epenthesis after final voiceless segments. Both of these flipped rules are applicable to *asX*, but not to each other's outputs.

FIGURE 32: Merger \Rightarrow mutual bleeding.

Merger is also relatable to (counter)feeding by the *crop* operation: an arrow can be added to the $\rightarrow . \leftarrow$ topology of merger to turn it into the $\rightarrow . \rightarrow . \leftarrow$ topology of (counter)feeding. This can be straightforwardly accomplished by extending deletion to apply after all consonants, not just voiceless ones ($crop(del) = i \rightarrow \emptyset / C _ \#$); the resulting added arrow creates the feeding configuration, and cropped deletion feeds devoicing: $ziX \xrightarrow{crop(del)} zX \xrightarrow{dvc} sX$.

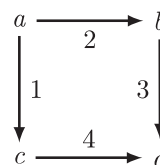
FIGURE 33: Merger \Rightarrow feeding.

Finally, consider the relation between merger and (counter)bleeding. Merger's topology, $\rightarrow . \leftarrow$, includes two arrows pointing at the same node, while the (counter)bleeding topology, $\leftarrow . \rightarrow . \rightarrow$, lacks a node with two arrows pointing at it. Thus, to convert merger to (counter)bleeding or vice-versa, both *flip* and *crop* are required. Merger can be converted via flipping to seeding (or to mutual bleeding, applying *flip* twice), and the resulting interaction can in turn be converted to (counter)bleeding via cropping. Or, merger can first be cropped into (counter)feeding, which can then be flipped into (counter)bleeding. All of the relationships among these order-independent relations are summarized in Figure 34.

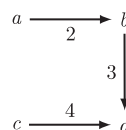
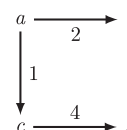
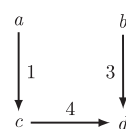
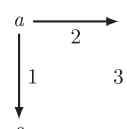
FIGURE 34: Flipping and cropping among order-independent interactions, with merger.

8. TYPOLOGY AND TOPOLOGY

As we've already seen along the way, the entire typology of map interactions can be visualized as the topology of arrow configurations. With two arrows, there are three possibilities: merger, $\rightarrow . \leftarrow$ mutual bleeding $\leftarrow . \rightarrow$, and seeding $\rightarrow . \rightarrow$ (equivalently, $\leftarrow . \leftarrow$). With three arrows, there are only two possibilities, (counter)feeding $\rightarrow . \rightarrow . \leftarrow$ (equivalently, $\rightarrow . \leftarrow . \leftarrow$) and (counter)bleeding $\leftarrow . \rightarrow . \rightarrow$ (equivalently, $\leftarrow . \leftarrow . \rightarrow$). To see that these are the only interactional possibilities, consider the 'full square' illustrated in Figure 35.

FIGURE 35: The full square.

Each of the four three-arrow configurations — the two topological possibilities, under each of two rule orders — can be obtained by excising one of the arrows from the square. Excising one of the arrows originating at *a* results in (counter)feeding: feeding if arrow 1 is excised (Figure 36) and counterfeeding if arrow 2 is excised (Figure 38). Excising one of the arrows pointing to *d* results in (counter)bleeding: bleeding if arrow 3 is excised (Figure 37), and counterbleeding if arrow 4 is excised (Figure 39).

FIGURE 36: F**FIGURE 37:** B**FIGURE 38:** CF**FIGURE 39:** CB

The two-arrow configurations can in turn be obtained in various ways from each of these three-arrow configurations. Excising arrow 2 from feeding (Figure 36) or

arrow 1 from counterfeeding (Figure 37) results in merger (Figure 40). Excising arrow 4 from feeding (Figure 36) or arrow 1 from counterbleeding (Figure 39) results in seeding (Figure 41). Excising arrow 3 from counterfeeding (Figure 37) or arrow 2 from bleeding (Figure 38) results in counterseeding (Figure 42). Finally, excising arrow 4 from bleeding (Figure 38) or arrow 3 from counterbleeding (Figure 39) results in mutual bleeding (Figure 43).

FIGURE 40: Mrg

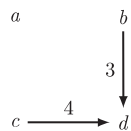


FIGURE 42: CS

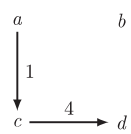


FIGURE 41: S

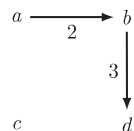
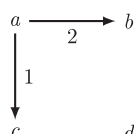


FIGURE 43: MB



9. CONCLUDING REMARKS

We have explored here the typological and topological space of pairwise rule interactions, based on a formal characterization of those interactions in terms of more basic relations: input-provision, output-provision, input-removal, and output-removal. These relations can be represented as arrows toward or away from (sets of) inputs and outputs of the rules, which can be flipped, excised, or added — the latter two forming complementary suboperations of a broader cropping operation. Representing conjunctive rule ordering two-dimensionally, with the earlier-ordered rule represented horizontally and the later-ordered rule represented vertically, allows rule re-ordering to be obtained via 90-degree rotation. The formal consequences and further development of the structure of this space are topics of ongoing research.

REFERENCES

- Baković, E. (2007). A revised typology of opaque generalisations. *Phonology*, 24, 217–259. ROA-850, Rutgers Optimality Archive, <http://roa.rutgers.edu>. <http://dx.doi.org/10.1017/S0952675707001194>
- Baković, E. (2011). Opacity and ordering. In J. A. Goldsmith, J. Riggle, & A. C. L. Yu (Eds.), *The Handbook of Phonological Theory* (pp. 40–67). Oxford: Wiley-Blackwell, 2nd edition. <http://dx.doi.org/10.1002/9781444343069.ch2>
- Baković, E. (2013a). *Blocking and Complementarity in Phonological Theory*. London: Equinox.
- Baković, E. (2013b). Defining phonological interactions with string set intersection. Handout, *WOpiG, Workshop on Opacity in Grammar*, Leipzig.
- Baković, E., & Blumenfeld, L. (2017). The interaction of phonological maps: a set-theoretic typology. Presented at the *Montréal-Ottawa-Toronto (MOT) Phonology Workshop*, UQAM.
- Baković, E., & Blumenfeld, L. (2018a). A formal typology of map interactions. Presented at the *Sound Workshop*, University of Massachusetts Linguistics Department, March 2018.
- Baković, E., & Blumenfeld, L. (2018b). Overapplication conversion. In R. Bennett, A. Angeles, A. Brasoveanu, D. Buckley, N. Kalivoda, S. Kawahara, G. McGuire, & J. Padgett, J. (Eds.), *Hana-Bana (花々): A Festschrift for Junko Ito and Armin Mester* (pp. 1–11). Department of Linguistics, University of California Santa Cruz.
- Baković, E., & Blumenfeld, L. (2019). A typology of map interactions. Manuscript in preparation, UC San Diego and Carleton University.
- Chafe, W. L. (1968). The ordering of phonological rules. *International Journal of American Linguistics*, 34, 115–136. <http://dx.doi.org/10.1086/465004>
- Chomsky, N., & Halle, M. (1968). *The Sound Pattern of English*. New York: Harper and Row. <https://dx.doi.org/10.1086/465290>
- Doke, C. M. (1938). *Textbook of Lamba Grammar*. Johannesburg: Witwatersrand University Press. <https://dx.doi.org/10.1017/S0041977X00088972>
- Hein, J., Murphy, A., & Zaleska, J. (2014). Rule flipping and the feeding-bleeding relationship. *Linguistische Arbeitsberichte*, 92, 1–31.
- Heinz, J. (2016). The computational nature of phonological generalizations. In L. Hyman & F. Plank, F. (Eds.), *Phonological Typology, Phonetics and Phonology* (pp. 126–195). The Hague: Mouton. <https://dx.doi.org/10.1515/9783110451931-005>
- Kenstowicz, M. J., & Kisseberth, C. W. (1979). *Generative Phonology: Description and Theory*. San Diego, CA: Academic Press. <https://dx.doi.org/10.2307/413703>
- Kiparsky, P. (1968). Linguistic universals and linguistic change. In E. Bach & R. T. Harms (Eds.), *Universals in Linguistic Theory* (pp. 170–202). New York: Holt, Reinhart, and Winston. Reprinted in *Explanation in Phonology* (pp. 3–55). Dordrecht: Foris, 1982. <https://dx.doi.org/10.1515/9783111666242.13>
- Kiparsky, P. (1973). How abstract is phonology? In O. Fujimura (Ed.), *Three Dimensions of Linguistic Theory. Part I: Phonological Representations* (pp. 5–56). Tokyo: TEC. Reprinted in *Explanation in Phonology* (pp. 119–163), Dordrecht: Foris, 1982.
- Kiparsky, P. (1993). Blocking in nonderived environments. In S. Hargus & E. M. Kaisse (Eds.), *Studies in Lexical Phonology* (pp. 277–313). San Diego, CA: Academic Press. <http://dx.doi.org/10.1016/B978-0-12-325071-1.50016-9>
- Koutsoudas, A., Sanders, G., & Noll, C. (1974). On the application of phonological rules. *Language*, 50, 1–28. <http://dx.doi.org/10.2307/412007>
- McCarthy, J. J. (1991). Synchronic rule inversion. In L. A. Sutton, C. Johnson & R. Shields (Eds.), *Seventeenth Annual Meeting of the Berkeley Linguistics Society*, pp. 192–207. <http://dx.doi.org/10.3765/bls.v17i0.1628>
- McCarthy, J. J. (1999). Sympathy and phonological opacity. *Phonology*, 16, 331–399. <http://dx.doi.org/10.1017/S0952675799003784>
- Prince, A. (2007). In pursuit of theory. In de Lacy, P. (Ed.), *The Cambridge Handbook of Phonology* (pp. 33–60). Cambridge: Cambridge University Press. <http://dx.doi.org/10.1017/CBO9780511486371.003>
- Vennemann, T. (1972). Rule inversion. *Lingua*, 29, 209–242. [https://doi.org/10.1016/0024-3841\(72\)90025-3](https://doi.org/10.1016/0024-3841(72)90025-3)
- Zwicky, A. (1987). Rule interactions: another gloss on K&K. *Innovations in Linguistics Education*, 5(1), 91–111.