A formal typology of process interactions

Eric Baković[a] & Lev Blumenfeld[b*]

[a]University of California, San Diego – ebakovic@ucsd.edu
[b]Carleton University – lev.blumenfeld@carleton.ca

Some phonologically significant generalizations result from processes, often formalized as rewrite rules, while others result from interactions among independently motivated processes, often formalized in terms of serial ordering. We adopt these general formalizations of processes and interactions to address two questions. One is the interaction question: what are all the possible forms of interaction between two processes? The other is the opacity question: what makes an interaction between two processes opaque? We show that these questions are best addressed with a rigorous algebraic formalization of processes and their pairwise interactions, describing the complete formal typology of process interactions and identifying the formal properties of those interactions that lead to different types of opacity.

Keywords: ordering, opacity, rules, interaction, phonology

1 Introduction

An abiding assumption amongst phonologists is that some phonologically significant generalizations result from PROCESSES, which take members of a defined set of representations as input and map them to members of a different (or modified) set of representations as output, while other phonologically significant generalizations result from INTERACTIONS among independently motivated processes. In the earliest days of generative phonology (Halle 1962; Chomsky 1967; Chomsky & Halle 1968), and still evident in much work today, processes have been formalized as REWRITE RULES and interactions among them have been analyzed in terms of their SERIAL ORDERING. In the context of this general formalization of processes and interactions, we pose the following question.

(1) **The interaction question**

What are all the possible forms of interaction between two processes?

Our aim is to show that this question is best addressed with a rigorous algebraic formalization of processes and their pairwise interactions. In some ways, this is well-trodden ground. The basic concepts behind what we will refer to as the 'classical' typology of pairwise rule interaction – FEEDING, BLEEDING, COUNTERFEEDING, and COUNTERBLEEDING – are familiar to phonologists from a large body of work.[1] But some of this work has, at different times and in different ways, exposed gaps in the classical typology.

[1] We must limit our references here to a few relevant handbook chapters – Iverson (1995), McCarthy (2007a), Baković (2011), Bye (2011), and Mascaró (2011) – and point the reader to the references therein.

Most significantly, additional pairwise rule interaction types have been identified – MUTUAL BLEEDING (Kiparsky 1971), DUKE-OF-YORK DERIVATIONS (Pullum 1976; McCarthy 1999, 2003), TRANSFUSION or SHIFTING (Zwicky 1987; Kiparsky 2015; Rasin 2022; Pruitt 2023), OPAQUE FEEDING (Lee 1999, 2007; Baković 2007, 2011), FED COUNTERFEEDING (Kavitskaya & Staroverov 2010) – demonstrating that the set of possible pairwise process interactions is broader than the classical typology admits and begging the interaction question in (1).

The algebraic approach that we take in this paper yields a formal typology in which process interaction types like feeding, bleeding, and others result from various combinations of more elementary formal properties. Throughout the paper we adopt a chemistry metaphor: the observed process interaction types are like *molecules*, consisting of various combinations of more basic formal components, or *atoms*. Much of the paper is devoted to exploring the set of possible atoms and possible molecules. For example, one of the atoms of an interaction whereby one process P feeds another process Q is that P *provides inputs to Q*, and one of the atoms of an interaction whereby P bleeds Q is that P *removes inputs from Q*. We show in later sections that feeding and bleeding also contain other atoms, and that other interaction types contain some of the same atoms that feeding and bleeding do. We formalize and explain the set of possible atoms, showing how they combine to form different molecular interaction types familiar from the literature, as well as some that are unfamiliar. Indeed, we show that all pairwise interaction types that are made possible by rules and ordering are molecules that can be broken down into distinct sets of these more basic atoms.

Our answer to the interaction question turns out to shine a bright light on an answer to a second question, one of perennial significance to phonological theory, concerning OPACITY.

(2)   **The opacity question**

   What makes an interaction between two processes opaque (conversely, transparent)?

Discussion of opacity abounds in previous work (see again the work cited in fn. 1), and is intimately tied to interaction and in particular to serial rule ordering. In a nutshell, an opaque order between two rules results in outputs where the generalization about expected surface forms stated by the first rule is obscured in some way by the subsequent application of the second rule. Opacity is thus a property of the output of a serial derivation relative to the rules that have applied to it. Kiparsky (1973b:79) offers the following diagnostic characterization of two types of opacity, centered on the rule R that is rendered opaque.

(3)   A phonological rule R of the form $a \longrightarrow b \mathbin{/} c \underline{} d$ is OPAQUE if there are surface structures with either of the following characteristics:

   a.   instances of a in the environment $c \underline{} d$, or

   b.   instances of b derived by R that occur in environments other than $c \underline{} d$.

The first type of opacity in (3a) refers to derivations ending in outputs to which R was expected to have applied but has not in fact applied, and is now commonly referred to as UNDERAPPLICATION opacity (McCarthy 1999). To establish underapplication, all that is needed is an inventory of rules and surface structures. The second type of opacity in (3b) refers to derivations ending in outputs to which R has applied unexpectedly. This type of OVERAPPLICATION opacity is more complex than underapplication, obtaining when surface structures are derived by a rule R but the conditions for R's application are not present in the output. To establish overapplication, in addition to rules and surface structures, one must also know which parts of those surface structures were derived by which rules.

While intuitively graspable, opacity has proven difficult to understand formally. The classical view is that counterfeeding and counterbleeding interactions are opaque while feeding and bleeding interactions are transparent (Kiparsky 1973b; McCarthy 1999), but Baković (2007, 2011) has shown the limitations of this view; see also Kiparsky (2015), Prillop (2018), and Rasin (2022). This is where our algebraic approach

most obviously proves its usefulness: we demonstrate that the same formalization of process interactions that answers the interaction question in (1) leads to a satisfying answer to the opacity question in (2).

The paper unfolds as follows. In §2, we clarify what we mean by processes and pairwise process interaction, elaborate on the classical typology of pairwise interaction, discuss its limitations, and chart the path beyond it. In §3, we introduce the algebraic formalism that we will use to describe the complete, atomized formal typology of process interactions in §5, answering the interaction question. Along the way, in §4, we will see that this approach provides the key to answering the opacity question. In §6 we discuss a way to characterize the typology in terms of a set of formal relationships among the various molecular interaction types in the typology. We conclude the paper in §7 with a summary and prospect.

2   Interactions

2.1   Processes and pairwise interactions

As noted in §1, we adopt for our purposes here the formalization of processes as rewrite rules. We thus use the terms 'process' and 'rule' interchangeably. We also use, synonymously, a more neutral term 'map' to refer to any function from a set of inputs to a set of outputs. That said, nothing in our formal machinery relies on a process, or map, being statable in any particular format. For the sake of concreteness and consistency with the existing literature, however, our exposition is mainly exemplified by rewrite rules statable in the familiar form $a \longrightarrow b \mathbin{/} c \underline{\quad} d$, where a is the FOCUS of the rule, a natural class description of phonological segments; b is the CHANGE of the rule, indicating phonological feature values that are substituted for conflicting values in a; c and d are the CONTEXT of the rule, and are either natural class descriptions, boundaries (word, syllable, etc.), or null. Also, a or b (but not both) may be null: if a is null, the rule inserts a segment with the feature values specified in b; if b is null, the rule deletes a segment with the feature values specified in a. Phonological forms are assumed to be strings of segments defined in terms of binary features, with appropriately-placed boundaries. The rule $a \longrightarrow b \mathbin{/} c \underline{\quad} d$ applies to all such forms containing the substring *cad*, as many times as the substring is present in the form.

Although most of our examples are stated in this general formalism, we are not beholden to it, such as in the statement of the (disjunctive) stress rule in Palestinian Arabic discussed in §2.3. Indeed, any well-defined total function mapping the members of a set of input forms to members of a set of output forms can in principle constitute what we call a 'process' here. Our focus, after all, is on the *interaction* between such broadly-defined processes.

The scope of our approach has some potential limitations. First, our focus on *pairwise* interactions appears at first to ignore interactions among three or more processes (e.g. 'rule sandwiching'; Bye 2002). But given the serial ordering model, any such case can be recursively reduced to a pairwise interaction between the first- or last-ordered of those processes and a composed input-output map representing the overall effect of the remaining processes. Of course, composed maps of this type may affect strings in complex and/or distributed ways that isolated processes as we've generally circumscribed them above typically do not, but we assert that the formal landscape of interaction is the same.

Second, interaction types that have been identified in the literature but that are not statable with serially ordered rules *tout court* are simply outside of our scope: disjunctive application (Chomsky & Halle 1968; Kiparsky 1973c; Baković 2013), nonderived environment blocking (Kiparsky 1968, 1993; Łubowicz 2002; Burzio 2000, 2011; Kula 2008; Rasin 2016; Chandlee 2021), cross-derivational feeding (Baković 2005, 2007), counterfeeding from the past (Wilson 2006; Wolf 2011; Andersson 2020) and from the future (Sayeed 2016), saltation (White 2013, 2014, 2017; Hayes & White 2015; Tesar 2014; Smith 2022), etc. However, functions that effectively instantiate the overall map defined by some such interaction types may be invoked as individual processes involved in a pairwise interaction within our scope, again such as the case of the disjunctive stress rule in Palestinian Arabic discussed in §2.3.

One reviewer's critique of our approach is that it is not able to describe mutual counterfeeding or

mutual counterbleeding (Wolf 2011), two unattested interaction types whereby two processes counterfeed or counterbleed each other because both necessarily apply *simultaneously* to the same input representation. We are not concerned with such interaction types, and neither is any approach which models phonology as serial process application. To put it another way, our purpose in this paper is to formally characterize the types of interaction that are describable in a theory that assumes the serial ordering of processes. Phenomena described by other possible theories are thus outside our scope.

2.2   The classical typology

Taking as background the informal notion of ordering given in the introduction, we will incrementally improve its precision, starting with a view that approximates the classical typology in the extensive literature alluded to in the introduction.

In what follows, we will generally refer to the two rules involved in an interaction as P and Q, and to a (hypothetical) form serving as a potential input to P or Q as x. $P(x)$ denotes the result of applying P to x and $Q(P(x))$ denotes the result of applying Q to the result of applying P to x, and similarly for $Q(x)$ and $P(Q(x))$. Minimally, an INTERACTION between P and Q for some input x obtains if the order of application of P and Q makes a difference; that is, iff $P(Q(x)) \neq Q(P(x))$. If the application of P to x *causes* an application of Q, it means that Q would not have applied to x had P not applied first. In other words, $P(x) \neq x$, and $Q(x) = x \neq Q(P(x))$. Such a situation is what is traditionally called FEEDING. Conversely, if the application of P to x *prevents* an application of Q, it means that Q would have been applicable to x had P not applied first. In other words, $P(x) \neq x$, and $Q(P(x)) = P(x) \neq Q(x)$. Such a situation is what is traditionally called BLEEDING.

Feeding and bleeding as just defined are properties of a particular derivation: we can inspect the fate of a given input x in the course of application of rules P and Q and determine whether the derivation is an instance of feeding, bleeding, or neither. In practice, the same terms are also extended to relations between rules as a whole, because rules typically proposed in analyses of real phonologies tend to have similar interactions with each other across many derivations, and usually do not display feeding for some inputs and bleeding for others (though this is not always true; see e.g. fn. 9 in §3.2, and also §5.6).

Additionally, either of the two orders $P > Q$ and $Q > P$ is available in any given grammar, where '$X > Y$' means 'X precedes Y in the order'. This gives rise to a terminological ambiguity: the terms *feeding* and *bleeding* can be used not only as labels for abstract relations but also more specifically as indications of a particular grammar, one where $P > Q$. The reverse order $Q > P$ results in derivations traditionally called COUNTERFEEDING and COUNTERBLEEDING.

The full classical typology of interactions is given in (4), cross-classified in terms of application causation vs. prevention on the one hand and the order of the two rules on the other. In addition, feeding and bleeding are traditionally classified as transparent while their counter-orders are classified as opaque, with counterfeeding involving underapplication and counterbleeding involving overapplication.

(4)

	$P > Q$	$Q > P$	
P causes application of Q	feeding	counterfeeding	Q underapplies
P prevents application of Q	bleeding	counterbleeding	Q overapplies
	transparent	opaque	

Our aim is to expand upon this classical typology and to flesh out its formal structure, but for background we offer brief examples of the interactions in (4).

Feeding and counterfeeding are illustrated by two rules based on examples from Bedouin Arabic (Al-Mozainy 1981; McCarthy 1999), glide vocalization (P) and raising in nonfinal open syllables (Q), shown

in (5). Vocalization potentially feeds raising: if vocalization applies first, then vocalization feeds raising; otherwise, vocalization counterfeeds raising, as in actual Bedouin Arabic. The mapping of /badw/ to [badu] 'Bedouin' shows underapplication (3a): raising could have applied to this form but did not. While both hypothetical orders and their effects are shown in (5), here and in other examples the observed order of application in the language that the example is based on is indicated by a checkmark ($\checkmark$) next to that rule order. Note that the choice of labels P and Q for the rules in each example language is arbitrary; the actual rule order may be $P > Q$ in some cases and $Q > P$ in others.

(5)　　P: [−cons] $\longrightarrow$ [+syll] / C ___ #　　　　　　　　　　　　　　　　　　(vocalization)

　　　　Q: a $\longrightarrow$ i / ___ CV　　　　　　　　　　　　　　　　　　　　　　　　　(raising)

　　　　　($P > Q$): /badw/ $\xrightarrow{P}$ badu $\xrightarrow{Q}$ bidu　　　　　　　　　　　　(feeding, transparent)

　　　　$\checkmark$($Q > P$): /badw/ $\xrightarrow{Q}$ badw $\xrightarrow{P}$ badu 'Bedouin'　　　　　(counterfeeding, Q underapplies)

Bleeding and counterbleeding are illustrated by two rules based on examples from Polish (Bethin 1978; Kenstowicz & Kisseberth 1979), final devoicing (P) and raising before final voiced nonnasals (Q), shown in (6). Devoicing potentially bleeds raising: if devoicing applies first, then devoicing bleeds raising; otherwise, devoicing counterbleeds raising, as in actual Polish. The mapping of /ʒwob/ to [ʒwup] 'crib' shows overapplication: raising appears on the surface as though it should not have applied, but it did.

(6)　　P: [−son] $\longrightarrow$ [−voi] / ___ #　　　　　　　　　　　　　　　　　　　　(devoicing)

　　　　Q: o $\longrightarrow$ u / ___ $\begin{bmatrix} +\text{voi} \\ -\text{nas} \end{bmatrix}$ #　　　　　　　　　　　　　　　　　　　　(raising)

　　　　　($P > Q$): /ʒwob/ $\xrightarrow{P}$ ʒwop $\xrightarrow{Q}$ ʒwop　　　　　　　　　　　(bleeding, transparent)

　　　　$\checkmark$($Q > P$): /ʒwob/ $\xrightarrow{Q}$ ʒwub $\xrightarrow{P}$ ʒwup 'crib'　　　　　　(counterbleeding, Q overapplies)

The notion of opacity appears coherent, even if informally so: derivations are opaque when their outputs are somehow inconsistent with the processes that produce those outputs. A natural question arises whether opacity can be formally characterized as a unified phenomenon, rather than as the disjunction of underapplication and overapplication seen in (3). We will see in §4 that this disjunction is inevitable, but in the following subsection we will consider a previous attempt at a unified characterization of opacity.

2.3　Misapplication and the Simultaneous Application Condition

The first step beyond the classical typology in (4) is the observation that some process interactions that appear opaque are not obviously classifiable as underapplication or overapplication, nor indeed as feeding or bleeding. This happens in cases where the application of one rule shifts or otherwise alters the result of the application of the other. In such cases, opacity is neither underapplication nor overapplication but might instead be called MISAPPLICATION, or application that appears to be 'wrong' in some way.

Consider the case in (7), based on examples from Palestinian Arabic (Hayes 1995:125; Kiparsky 2000; Watson 2011). An epenthesis rule breaks up final consonant clusters (P), and the relevant part of the stress rule is disjunctively quantity-sensitive (Q). (Although the stress rule is stated disjunctively, we view it not as a disjunctive set of multiple maps but rather as a single input-output map, and consider here its interaction with another single map, namely epenthesis). For input /katabt/ 'I wrote', both rules are applicable. The actual output is [katábit], with unexpected stress on a light penult, because stress applies prior to epenthesis. The reverse order would have produced *[kátabit].

(7) P: $\varnothing \longrightarrow$ i / C ___ C#   (epenthesis)

   Q: Stress superheavy final, else heavy penult, else antepenult   (stress)

   $(P > Q)$: /katabt/ $\xrightarrow{P}$ katabit $\xrightarrow{Q}$ kátabit   (transparent)

   ✓$(Q > P)$: /katabt/ $\xrightarrow{Q}$ katábt $\xrightarrow{P}$ katábit 'I wrote'   (Q misapplies)

A different type of case is the interaction in (8) between long vowel lowering (P) and vowel copy epenthesis (Q), based on examples from Yokuts (Kuroda 1967; Steriade 1986). For the input stem /sduːk-/ 'remove', both rules are applicable. The actual output is [sudoːk-], with an unexpected imperfect copy vowel [u], because vowel copy epenthesis applies prior to long vowel lowering. The reverse order would have produced *[sodoːk-].

(8) P: [+long] $\longrightarrow$ [−high]   (lowering)

   Q: $\varnothing \longrightarrow \alpha$V / #C ___ C αV   (epenthesis)

   $(P > Q)$: /sduːk-/ $\xrightarrow{P}$ sdoːk- $\xrightarrow{Q}$ sodoːk-   (transparent)

   ✓$(Q > P)$: /sduːk-/ $\xrightarrow{Q}$ suduːk- $\xrightarrow{P}$ sudoːk- 'remove'   (Q misapplies)

First note that neither of these examples can be categorized as feeding, bleeding, or their counter-orders, disrupting the classical typology in (4). In both cases, the rule labeled P has an influence on *how Q* applies, whereas the four types of interaction in the classical typology are concerned with *whether Q* applies. Both P and Q apply under both orders in (7) and (8), but the result of Q's application depends on its order with respect to P. This obviously differs from the feeding/counterfeeding and bleeding/counterbleeding cases. In the feeding/counterfeeding case in (5), Q only applies when $P > Q$, and in the bleeding/counterbleeding case in (6), Q only applies when $Q > P$. We choose to use the term TRANSFUSION to refer to this nonclassical type of interaction, borrowing the term from unpublished work by Don Churma cited by Zwicky (1987).[2]

Intuitively, the correct outcomes in both (7) and (8) are opaque. In (7), the location of stress in [katábit] is inconsistent with the stress rule: stress has shifted to a light penult from what would have been a superheavy ultima, whereas the transparent expectation is that stress should fall on the antepenult. In (8), the vowel inserted in [sudoːk-] is inconsistent with copy epenthesis: the vowel is [u], copying the following underlying /uː/, whereas the transparent expectation is that the vowel should be [o], copying the following surface [oː].

At the same time, it is difficult to interpret these derivations in terms of underapplication or overapplication. The apparent opacity in (7) is in the unexpected penultimate location of stress, but has antepenultimate stress underapplied despite the fact that its conditions are met on the surface, or has penultimate stress overapplied despite the fact that its conditions are *not* met on the surface? Likewise, the apparent opacity in (8) is in the unexpected inserted [u], but has copy epenthesis of [o] underapplied despite the fact that its conditions are met on the surface, or has copy epenthesis of [u] overapplied despite the fact that its conditions are *not* met on the surface? The answer in both cases is neither, showing that (3) is incomplete as a diagnostic for opacity because it does not cover misapplication.

A more general diagnostic for opacity is offered by Joshi & Kiparsky (1979, 2005) and Kiparsky (2015), following Pāṇini. Let $P, Q(x)$ denote the result of SIMULTANEOUS APPLICATION of rules P and Q to x (as in the 'direct mapping hypothesis' of Kenstowicz & Kisseberth 1979:291*ff*), which contains mappings required by P and Q based only on conditions present in the input x. (It is undefined when the two mappings lead to incompatible results.) Opacity can then be given a unified characterization, as follows.

---

[2] Rasin (2022) refers to the type of interaction as (COUNTER)SHIFTING (see also Pruitt 2023), which we find is more apt in the stress case in (7) than in the vowel copy epenthesis case in (8). See also fn. 21 below.

(9)   **The Simultaneous Application Condition** on opacity

     Q is opaque if there exists an input x such that $P(Q(x)) = P, Q(x) \neq Q(P(x))$.

Unpacking this condition a bit: a rule Q is opaque with respect to another rule P if (a) for at least one input x, the output of x depends on the order of application between Q and $P - P(Q(x)) \neq Q(P(x))$ – and (b) simultaneous application gives the same output as applying Q to x first, and applying P to the result of that – $P(Q(x)) = P, Q(x)$. Simultaneous application of P and Q effectively eliminates intermediate representations and thus clarifies how each rule is independently expected to apply to the input x. To the extent that $P(Q(x)) = P, Q(x)$, the three forms of opacity are unified: $P, Q(x)$ ignores all effects of P on Q, whether they involve P newly providing Q's conditions (= underapplication of Q), removing them (= overapplication of Q), or affecting the result of Q's application (= misapplication of Q).[3] This is illustrated below using the examples discussed in (5)–(8) above, for which the Simultaneous Application Condition correctly identifies Q as opaque when applied before P.

First, the counterfeeding/underapplication example: given the Bedouin Arabic input /badw/, applying raising before vocalization yields the same result as applying both simultaneously, and a different result than applying vocalization first, as shown in (10). This is because raising is not applicable until after vocalization.

(10) P: vocalization; Q: raising   cf. Bedouin Arabic, (5)

     $P(Q(\text{badw})) = P, Q(\text{badw}) \neq Q(P(\text{badw}))$
       badu   =   badu $\neq$ bidu

Next, the counterbleeding/overapplication example: given the Polish input /ʒwob/, applying raising before devoicing yields the same result as applying both simultaneously, and a different result than applying devoicing first, as shown in (11). This is because raising is no longer applicable after devoicing.

(11) P: devoicing; Q: raising   cf. Polish, (6)

     $P(Q(\text{ʒwob})) = P, Q(\text{ʒwob}) \neq Q(P(\text{ʒwob}))$
       ʒwup   =   ʒwup $\neq$ ʒwop

Now, the first transfusion/misapplication example: given the Palestinian Arabic input /katabt/, applying stress before epenthesis yields the same result as applying both simultaneously, and a different result than applying epenthesis first, as shown in (12). Stress is applicable before and after epenthesis, but because epenthesis disrupts both the composition and relative position of syllables in the form, stress falls on a different syllable depending on the order between stress and epenthesis.

(12) P: epenthesis; Q: stress   cf. Palestinian Arabic, (7)

     $P(Q(\text{katabt})) = P, Q(\text{katabt}) \neq Q(P(\text{katabt}))$
       katábit   =   katábit $\neq$ kátabit

Finally, the other transfusion/misapplication example: given the Yokuts input /sduːk-/, applying epenthesis before lowering yields the same result as applying both simultaneously, and a different result than applying lowering first, as shown in (13). Epenthesis is applicable before and after lowering, but because lowering changes the vowel to be copied by epenthesis, the quality of the epenthetic vowel differs if epenthesis is applied before lowering than if applied afterwards.

---

[3] To be clear, simultaneous application is used here merely as a diagnostic of opaque interaction. No claim is made about the possibility of *crucial* simultaneous application, for example as invoked in the analysis of the (unattested) 'mutual counterfeeding' and 'mutual counterbleeding' interactions discussed by Wolf (2011).

(13) P: lowering; Q: epenthesis   cf. Yokuts, (8)

$$P(Q(\text{sduːk-})) = P, Q(\text{sduːk-}) \neq Q(P(\text{sduːk-}))$$
$$\text{sudoːk-} \quad = \quad \text{sudoːk-} \quad \neq \quad \text{sodoːk-}$$

As we will see in the next subsection, there are further opaque interaction types beyond the classical typology and beyond the reach of the Simultaneous Application Condition.

2.4 Further beyond the classical typology

The classical view summarized in (4) suggests a four-way ordering typology that correlates with opacity: feeding and bleeding orders are transparent, while their counter-orders are opaque. There are also examples where opacity appears in the form of misapplication. A well-defined criterion based on simultaneous application unifies all of these types of opacity.

And yet, this typology is still incomplete. A class of OPAQUE FEEDING interactions outside of its boundaries is identified by Lee (1999, 2007) and Baković (2007, 2011), which we will refer to more succinctly here as SEEDING.[4] A case based on examples from Turkish illustrates (Sprouse 1997; Baković 2007, 2011). Here, epenthesis into final clusters (P) potentially interacts with intervocalic velar deletion (Q). Let us start with the second order listed in (14), where Q is ordered first. For inputs like /bebekn/ there appears to be ordinary counterfeeding with underapplication: the environment of deletion is not met in the input, and epenthesis produces *[bebekin] – which would then be eligible for deletion, for which it is now too late to apply. The $Q > P$ order has the signature of counterfeeding with underapplication; it looks like *bebekin* should have undergone deletion, but it did not.

(14) P: ∅ ⟶ i / C ⎯ C#   (epenthesis)
      Q: k ⟶ ∅ / V ⎯ V   (deletion)

   ✓($P > Q$): /bebekn/ $\xrightarrow{P}$ bebekin $\xrightarrow{Q}$ bebein 'baby.ACC'   (seeding, P overapplies)
   ($Q > P$): /bebekn/ $\xrightarrow{Q}$ bebekn $\xrightarrow{P}$ bebekin   (counterseeding, Q underapplies)

The order $P > Q$ leading to the result actually found in Turkish presents a challenge for the classical view summarized in (4). On the one hand, P appears to feed Q: epenthesis into the final cluster places the velar in an intervocalic position, where it is newly eligible for deletion, and deletion applies because it is ordered second. This is expected given the apparent counterfeeding nature of the $Q > P$ order. However, the output [bebein] has the character of *overapplication* opacity: it looks like epenthesis should not have applied, but it did nevertheless. The destruction of the environment of epenthesis, namely the deletion of the first consonant of the input cluster, was only possible because of the application of a rule that is itself fed by epenthesis. This is why this type of interaction was originally dubbed SELF-DESTRUCTIVE FEEDING: the application of epenthesis "sows the seeds" of its own overapplication by causing another rule to apply that destroys the context that motivated epenthesis in the first place (Baković 2007:227). As we discuss below, such cases are not only outside of the scope of the typology in (4), but are also not amenable to the Simultaneous Application Condition, which fails to diagnose one of the opaque interactions correctly.

Another case of seeding, based on examples from Javanese (Lee 1999, 2007), is illustrated in (15). (Many thanks to Kie Zuraw for bringing this case and these citations to our attention.) Just as in Turkish, the incorrect $Q > P$ order results in what looks like counterfeeding with underapplication, but the correct $P > Q$ order involves overapplication.

---

[4] This term was introduced in Baković & Blumenfeld (2018, 2019) to clearly distinguish seeding from feeding, given the critical differences between these two classes of interactions elaborated upon in the text.

(15) P: n $\longrightarrow \varnothing$ / C $\underline{\quad}$ (*n*-deletion)

   Q: h $\longrightarrow \varnothing$ / V $\underline{\quad}$ V   (*h*-deletion)

   $\checkmark$ ($P > Q$): /omahne/ $\xrightarrow{P}$ omahe $\xrightarrow{Q}$ omae 'house'   (seeding, P overapplies)

   ($Q > P$): /omahne/ $\xrightarrow{Q}$ omahne $\xrightarrow{P}$ omahe   (counterseeding, Q underapplies)

Seeding is a classically unexpected case of opacity, given that it shares with feeding the property that one rule creates inputs to another rule. What we see from the examples in (14) and (15) is that seeding also shares the property of overapplication with counterbleeding.[5]

Another seeding interaction is a case that Baković (2007) refers to as NONGRATUITOUS FEEDING, so named because it is an instance of overapplication opacity that is technically amenable to analysis in Optimality Theory given that it does not involve a 'gratuitous' violation of faithfulness (McCarthy 1999). In other respects, however, nongratuitous feeding is indistinguishable from seeding. A case based on examples from Classical Arabic illustrates (McCarthy 2007b).

(16) P: $\varnothing \longrightarrow \alpha$V / # $\underline{\quad}$ CC αV   (vowel copy epenthesis)

   Q: $\varnothing \longrightarrow$ ʔ / # $\underline{\quad}$ V   (ʔ-insertion)

   $\checkmark$ ($P > Q$): /ktub/ $\xrightarrow{P}$ uktub $\xrightarrow{Q}$ ʔuktub 'write.IMP.MSG'   (seeding, P overapplies)

   ($Q > P$): /ktub/ $\xrightarrow{Q}$ ktub $\xrightarrow{P}$ uktub   (counterseeding, Q underapplies)

In this case, vowel copy epenthesis before initial clusters (P) potentially interacts with glottal stop insertion before initial vowels (Q). If Q is ordered first, then for inputs like /ktub/ there again appears to be classical counterfeeding with underapplication: the environment of ʔ-insertion is not met in the input, and vowel copy epenthesis produces *[uktub] – which would then be eligible for ʔ-insertion, for which it is now too late to apply. The $Q > P$ order has the signature of classical counterfeeding with underapplication; it looks like *[uktub] should have undergone ʔ-insertion, but it did not. The reverse order $P > Q$ – the order leading to the result actually found in Classical Arabic – is again a challenge for the classical typology in (4). On the one hand, P appears to feed Q: vowel copy epenthesis before an initial cluster places a vowel in initial position, where it is newly eligible for ʔ-insertion, and ʔ-insertion applies because it is ordered second. This apparent classical feeding interaction is again expected given the apparent counterfeeding character of the $Q > P$ order. However, the output [ʔuktub] again exhibits overapplication opacity: it looks like vowel copy epenthesis should not have applied, but it did. The destruction of the environment of vowel copy epenthesis, namely the insertion of a glottal stop before the inserted vowel, is only possible because of the application of a rule that is itself fed by vowel copy epenthesis.

Importantly, *both* orderings of P and Q are opaque in all of these seeding interactions, one with underapplication and the other with overapplication. Because the Simultaneous Application Condition is asymmetrical, it can unsurprisingly only diagnose one of them as opaque. According to this condition, only the derivations with what we have above called COUNTERSEEDING are correctly diagnosed as opaque, as shown in (17). The $P(Q(x))$ forms on the left are the correctly-diagnosed counterseeding results; the $Q(P(x))$ forms on the right are the incorrectly-diagnosed seeding results.

---

[5] Indeed, both Moreton & Smolensky (2002:315) and Potts & Pullum (2002:384–385) misclassify the correct Turkish derivation in (14) as an instance of counterbleeding, as noted by Baković (2007:228).

(17)　a.　P: epenthesis; Q: deletion　　　　　　　　　　　　　　　　cf. Turkish, (14)

$$P(Q(\text{bebekn})) \quad = \quad P, Q(\text{bebekn}) \quad \neq \quad Q(P(\text{bebekn}))$$
$$\text{bebekin} \qquad = \qquad \text{bebekin} \qquad \neq \qquad \text{bebein}$$

　　b.　P: *n*-deletion; Q: *h*-deletion　　　　　　　　　　　　　　cf. Javanese, (15)

$$P(Q(\text{omahne})) \quad = \quad P, Q(\text{omahne}) \quad \neq \quad Q(P(\text{omahne}))$$
$$\text{omahe} \qquad = \qquad \text{omahe} \qquad \neq \qquad \text{omae}$$

　　c.　P: vowel copy epenthesis; Q: ʔ-insertion　　　　　　　cf. Classical Arabic, (16)

$$P(Q(\text{ktub})) \quad = \quad P, Q(\text{ktub}) \quad \neq \quad Q(P(\text{ktub}))$$
$$\text{uktub} \qquad = \qquad \text{uktub} \qquad \neq \qquad \text{ʔuktub}$$

There is no mystery as to why the (correct) seeding derivations with overapplication are not detected by the Simultaneous Application Condition. Generally, simultaneous application ignores the potential effects of one rule on another. In our seeding examples, the effect of Q on P is to destroy the environment required by P, but the problem is that Q is itself not applicable to the input, which is the level at which the Simultaneous Application Condition is evaluated. Instead, Q only gets to apply due to the effects of P.[6]

In sum, we have demonstrated above that the classical typology of feeding and bleeding in (4), even with the more accurate diagnostic tool of the Simultaneous Application Condition on opacity (9), fails to properly categorize seeding because this type of interaction shares properties with feeding (P causes application of Q) and with counterbleeding (P overapplies). In the remainder of the paper we flesh out a formal typology that solves this problem, demonstrating precisely what seeding shares with feeding and what it shares with counterbleeding. In doing so, we will go far beyond the interaction types identified thus far, and characterize the formal space of all possible pairwise process interactions. As a preliminary step to developing the typology, we first develop in the next section a formal and precise understanding of how one rule's application can be caused or prevented by another.

3　Algebra

3.1　Formal basics

Let Σ be an alphabet, and Σ^* be the set of finite strings over Σ. A PHONOLOGICAL MAP, or MAP for short, is a total function on Σ^*, enhanced with a correspondence relation. Thus, for each element of an input string, there are zero or more elements of the output string that correspond to it, and vice versa.[7] We will use the term MAPPING and the interchangeable notations $x \xrightarrow{P} y$, $P(x) = y$, and (x, y) for individual input-output pairs. A VACUOUS mapping is a mapping where the input and output are segmentally identical, and the

---

[6] More carefully: because the Simultaneous Application Condition requires the *existence* of a form with the necessary property, to make the claim that it fails to identify P as opaque we need to show that no form x exists such that $P(Q(x)) \neq P, Q(x) = Q(P(x))$. In the case of Turkish (14)/(17a), for example, there must be no form where the outcome of simultaneous application differs from the outcome of applying epenthesis (= P) first. Indeed, if such a form could exist, the application of epenthesis would depend on whether or not deletion (= Q) has applied. But epenthesis requires a word-final consonant cluster, and intervocalic deletion is incapable of destroying such a cluster (because neither of the Cs is intervocalic) or of creating one (because no Cs outside of the V*k*V sequence are affected). The applicability of this reasoning to the Javanese (15)/(17b) and Classical Arabic (16)/(17c) cases is left as an exercise.

[7] Correspondence is only introduced here for formal completeness; correspondence relations in our examples are straightforward. Formally: a string of length n can be modeled as a function $f : N \longrightarrow \Sigma$, where $N = \{i \mid 1 \leqslant i \leqslant n\}$. So, a string is a set of pairs (i, s_i), where i is a position and s_i is a segment. A correspondence relation between strings S and T is a subset of $S \times T$. For example, strings *foo* and *bar* standing in one-to-one correspondence are modeled as follows: the strings are $\{(1, f), (2, o), (3, o)\}$ and $\{(1, b), (2, a), (3, r)\}$, and the correspondence relation is $\{((1, f), (1, b)), ((2, o), (2, a)), ((3, o), (3, r))\}$.

correspondence relation between the input and output is one-to-one and preserves precedence.[8] We may use the term *apply* to mean 'apply nonvacuously' (as in, "velar palatalization applies to *ki* but does not apply to *či* or *ku*"), but we understand that formally vacuous input-output pairs like (*či*, *či*) and (*ku*, *ku*) are parts of maps along with nonvacuous pairs like (*ki*, *či*).

In this paper, the notion of mapping is only formally defined for linear representations, i.e. strings of segments. While a full formalization for autosegmental and prosodic representations is left for another day, in an informal way we will refer to stress assignment rules and stress representations, folding them into the linear view where a stressed or unstressed vowel is simply one of the segments in a string.

Let us call the INPUT SET of map P, written $In(P)$, the subset of Σ^* to which P applies nonvacuously. Likewise, the OUTPUT SET of map P, written $Out(P)$, is the subset of Σ^* which have nonvacuous inputs for map P. In other words, $In(P)$ are all the forms to which P potentially applies, while $Out(P)$ are all the forms to which P has potentially applied.

3.2   Feeding and bleeding preliminaries

Whether a rule P feeds or bleeds another rule Q, informally, has to do with whether P creates or destroys inputs of Q. A reasonable formal analog of this intuitive notion might be stated in terms of set membership: we could ask, does P make new elements of $In(Q)$, or does it remove existing elements from $In(Q)$? We will see later on that this naive formulation is inadequate, but its inadequacy will light the path to a better understanding of the phenomena at issue. These preliminary definitions are given in (18).

(18)   Feeding and bleeding: provisional definitions

   a. P feeds Q if $\exists x : x \in In(P), x \notin In(Q), P(x) \in In(Q)$.

   b. P bleeds Q if $\exists x : x \in In(P), x \in In(Q), P(x) \notin In(Q)$.

While the notions as defined in (18) refer to relations between entire maps, those relations are evidenced by specific strings that fit the requirements of (18). In common practice, the terms *feeding* and *bleeding* are ambiguous in that they can either be used globally to refer to map relations, or locally and specifically to refer to individual mappings.[9] We will continue this ambiguous usage. In particular, there is an expectation that the definitions in (18) work in reverse, as it were: it should be true that any sequence $x \xrightarrow{P} y \xrightarrow{Q} z$ that we understand informally to involve feeding or bleeding should fit the set membership definition in (18). As we will see, this is where the naive set-theoretic idea in (18) goes wrong.

To be sure, in most situations these definitions seem to characterize the interactions adequately. This is true, for example, of both the Bedouin Arabic and Polish cases in (5) and (6). With $x = badw$ in Bedouin Arabic, P (vocalization) feeds Q (raising), because *badw* is not a member of $In(Q)$, but $P(badw) = badu$ is. With $x = ʒwob$ in Polish, P (devoicing) bleeds Q (raising), because *ʒwob* is a member of $In(Q)$, but $P(ʒwob) = ʒwop$ is not. Creation and destruction of set elements seems a reasonable model of these interactions.

There is also promise in the observation that these definitions do *not* apply to the transfusion interaction between stress and epenthesis in Palestinian Arabic (7): since every unstressed string is a member of the input set of a stress rule, epenthesis into a final cluster neither creates nor destroys members of that set. Likewise for the transfusion interaction between vowel copy epenthesis and long vowel lowering in Yokuts (8): since every #CCV string is a member of the input set of vowel copy epenthesis, lowering of the vowel in such a

---

[8] Formally: let x be a string of length n, consisting of pairs (i, s_i) for $1 \leqslant i \leqslant n$ (see fn. 7). $P(x)$ is vacuous if $P(x) = x$, and the correspondence relation consists of $((i, s_i), (i, s_i))$ for all i between 1 and n.

[9] For example, given $P = V \longrightarrow \varnothing \,/ \underline{\quad} V$ and $Q = s \longrightarrow ʃ \,/ \underline{\quad} i$, P feeds Q in the derivation *sai* $\xrightarrow{P}$ *si* $\xrightarrow{Q}$ *ʃi* but P bleeds Q in the derivation *sia* $\xrightarrow{P}$ *sa* $\xrightarrow{Q}$ *sa* (Baković 2011:42). An example of this type is found in Karok (Bright 1957; Kenstowicz & Kisseberth 1979). See also discussion in §5.6.

string neither creates nor destroys members of that set. This reinforces our contention in §2.3 that feeding and bleeding are not appropriate to characterize those cases.

There are further situations where the definitions in (18) are insufficient. The clearest case involves interactions of the feeding/bleeding type where the rule that is fed or bled has more than one potential locus of application in a string. For example, consider the following hypothetical pair of processes, based very loosely on examples from Finnish (see (47) in §6).

(19) P: e $\longrightarrow$ i / __ #   (word-final raising)

   Q: k $\longrightarrow$ č / __ i   (velar palatalization)

P raises word-final mid vowels, and Q palatalizes velars before high vowels. Clearly, there is feeding here, and there are well-behaved examples, like *take* $\xrightarrow{P}$ *taki* $\xrightarrow{Q}$ *tači*. Raising in this example takes a string *take* which is not in $In(Q)$ and makes a string *taki* which is in $In(Q)$. We should also be able to say that a sequence like *kike* $\xrightarrow{P}$ *kiki* $\xrightarrow{Q}$ *čiči* also instantiates feeding: had it not been for raising, velar palatalization of the second *k* would not have applied. But this example doesn't work in the set-theoretic terms of (18): the raising rule, P, does not create a new member of $In(Q)$, since both *kike* and *kiki* are eligible for palatalization. What it does create is a new *locus of application* for Q, but this is not captured by (18). We will solve this issue formally in the following section, where we develop a way to count how many times a map has applied to a string, and then define feeding and bleeding as increases and reductions in those counts, respectively.[10]

3.3   String input-rank

Counting how many times a rule applies to a string might seem easy: you simply have to count how many corresponding input-output segment pairs are different. However, this will not work. There are many rules whose single application changes more than one segment, for example harmony or foot assignment that can affect an entire domain. Whether a rule does "one thing" to a string or "more than one thing" is a more subtle notion than mere counting of segmental changes, but still a notion that is amenable to formal analysis.

Informally, a map applies to a string more than once if the string can be broken into pieces and the map will still apply to each of those pieces in exactly the same way. This section formalizes this intuition. First we define the notion of a nonvacuous break (NVB).

(20)   Let P be a map, and x a string in $In(P)$. Then strings $a_1, a_2, \ldots, a_n$ are called a NONVACUOUS
   BREAK (NVB) of rank n of string x if:

   a. $x = a_1 a_2 \ldots a_n$,

   b. $P(x) = P(a_1)P(a_2)\ldots P(a_n)$, and

   c.   each $a_i \in In(P)$.

In other words, an NVB breaks up a string into separate nonvacuous applications of P that are independent of each other.[11] For velar palatalization, the string *taki* has only a trivial NVB of rank 1, *taki*. The string *kiki*

---

[10] Another strategy would be to try to keep track of the loci of application of rules, to be able to compare applications of one map across strings and thus make it possible to determine whether P causes a new instance of application of Q. This approach is challenging to make coherent for maps involving deletion and insertion, however, where the locus is not instantiated by a string position in both the input and the output.

[11] One formal complication that we note but do not resolve here is that NVB substrings must be allowed to overlap when the left context of one locus is the right context of another. For example, consider a rule that affects a consonant in intervocalic position, and the string VCVCV. According to our definition, there is no NVB of rank higher than 1, but intuitively, the rule would apply in two loci in this string. Allowing NVB substring overlap would solve this problem.

has a trivial NVB of rank 1, *kiki*, and a nontrivial NVB of rank 2, *ki-ki*. The string *kikiki* has one NVB of rank 1 (*kikiki*), two NVBs of rank 2 (*kiki-ki*, *ki-kiki*), and one of rank 3 (*ki-ki-ki*). Any string will have an upper limit on the rank of its NVBs, and this upper limit is the measure of how many times a map applies to that string. Let us call this measure a string's INPUT-RANK.

(21)   Let P be a map and x a string. Then:

   a.   the input-rank of x relative to P, written $\mathsf{IR}(P, x)$, is 0 if $x \notin In(P)$;

   b.   otherwise, $\mathsf{IR}(P, x)$ is the highest possible NVB rank of x for map P.

Input-rank thus partitions the set Σ^* into nonoverlapping subsets: those strings to which the rule doesn't apply at all, those to which the rule applies once, to which it applies twice, etc., for example, as shown in (22) for velar palatalization. For this map, and in fact for most ordinary segmental maps, there is in principle no upper limit for input-rank. Indeed, for a map specified by the rule $a \longrightarrow b \: / \: c \underline{\quad} d$, strings with arbitrarily large input-ranks can be constructed by concatenating *cad* arbitrarily many times.

(22)

string examples	input-rank
pa	
ku	0
banana	
ki	
taki	1
kibulipami	
kiki	
kipaki	2
takiki	
. . .	
kikikikikiki	6

On the other hand, there are maps with an upper limit on string input-rank. A clear example is the final devoicing map: there is no string with more than one application of final devoicing, and thus all strings have an input-rank of either 0 or 1. Note that strings like *adad* still have input-rank of 1. It might seem at first that *ad-ad* is an NVB of rank 2 because final devoicing applies to both of these substrings. However, condition (20b) of being an NVB is not satisfied: $P(ad)P(ad) = atat \neq P(adad) = adat$. The only NVB here is the trivial rank-1 NVB *adad*, and thus the string has an input-rank of 1.

   Likewise, for stress rules that assign only one privileged or primary stress per word, there is no string with input-rank higher than 1. If a string x were to have an NVB of rank higher than 1 for such a stress rule, say $x = a_1 a_2$, then each of a_1 and a_2 would have a primary stress, but x would have only one primary stress. The intuitive upshot is that a stress rule either doesn't apply to a string (e.g. because it already has stress), or applies once. More generally, this upper limit on input-rank holds of any map that targets a single, privileged element of a string. In practice, such rules reference unique positions such as *leftmost/initial* or *rightmost/final*. This is the case for rules like final devoicing and stress. Somewhat further afield, consider a palatalization rule that only applies to the *rightmost* eligible element of the string: /kipa/ $\longrightarrow$ *čipa*, but /kiki/ $\longrightarrow$ *kiči* (e.g. 'featural affixation'; Akinlabi 1996). This rule also has no strings with input-rank greater than 1.

   Alternatively (or in addition), of course, word boundaries can be considered to be elements occupying the initial and final positions of the original string, such that e.g. the relevant string under discussion in the final devoicing case is $\#adad\#$. There is then clearly no NVB of this string other than the trivial NVB $\#adad\#$, and similar considerations apply to the primary stress and rightmost palatalization examples just discussed.

3.4 Input interactions

We can now understand feeding as an increase in the input-rank of a string, and bleeding as a decrease, defined as INPUT-PROVISION and INPUT-REMOVAL in (23). Recalling our chemical metaphor from §1, input-provision and input-removal are atoms that combine to form molecular interactions. These two atoms are contained in the feeding and bleeding molecules, respectively, but as will become clear, these molecules crucially also contain other atoms and these atoms are also shared by other molecular interactions.

(23) a. P input-provides Q ('$P{+}\mathrm{i}Q$') if there is a mapping $x \xrightarrow{P} y$ such that $\mathsf{IR}(Q, x) < \mathsf{IR}(Q, y)$.

b. P input-removes Q ('$P{-}\mathrm{i}Q$') if there is a mapping $x \xrightarrow{P} y$ such that $\mathsf{IR}(Q, x) > \mathsf{IR}(Q, y)$.

We will use arrow diagrams to visualize input-provision and input-removal, as follows. We use the horizontal direction for P-mappings, and the vertical direction for Q-mappings. Input-ranks of strings will be shown adjacent to those strings in the direction of the mapping: to the right of the string for P input-ranks, below the string for Q input-ranks. The following strings illustrate, with P being palatalization ($ki \longrightarrow \check{c}i$), and Q final devoicing ($d\# \longrightarrow t\#$).

(24) Examples of input-rank for different strings (P = palatalization, right; Q = final devoicing, below)

kit 1	kid 1	kikid 2	kat 0	kad 0
0	1	1	0	1

For palatalization and devoicing, there is no interaction – no feeding or bleeding, input-provision or input-removal. The presence or absence of an interaction can be read off the diagrams below as follows. A map P affects (input-provides, input-removes) another map Q if there are unequal Q-input-ranks at opposite ends of a P-mapping. The input-ranks being compared are framed in boxes. Since we are here considering the effects of P on Q, those boxes are below the input and output forms. Figure 1 shows a case with no interaction; input-provision and input-removal are illustrated in Figures 2 and 3, respectively. To highlight the relevant comparison, a thick dashed line is drawn between the boxes with unequal ranks.[12]

Figure 1: No interaction   Figure 2: Input-provision   Figure 3: Input-removal
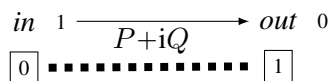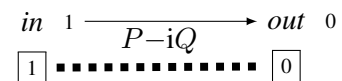
The diagrams in Figures 2 and 3 highlight a formal point: input-provision and input-removal are, strictly speaking, properties of particular input-output *mappings*, not of strings or of rules as a whole. Two rules P and Q, each defining an entire map and thus being a collection of mappings, may contain multiple instances of interaction atoms like $P{+}\mathrm{i}Q$ and $P{-}\mathrm{i}Q$, each originating with a particular mapping, or arrow in our diagrams (recall the example in fn. 9 of one rule feeding or bleeding another, depending on the input).

Let us now illustrate these interactions with feeding and bleeding. Once again, relevant input-ranks that determine the interactions are in boxes. The thick dashed lines connecting those boxed numbers highlight the relevant comparison: in the case of feeding in Figure 4, the input rank is higher in the output; in the case of bleeding in Figure 5, the input-rank is higher in the input. The downward Q-arrows are also shown, following the diagramatic convention described above. The role of these Q arrows will become clearer below.

---

[12] The numbers here are 0 and 1, but they could be e.g. 2 and 0, or 1 and 3 – as long as the Q-input-ranks at the opposite ends of the P-arrow are *unequal*, there is an interaction. (At the opposite ends of the P-arrow, P-input-ranks are also unequal, but what's relevant is how P affects the input-rank of Q.)
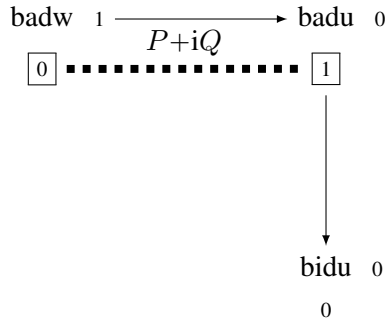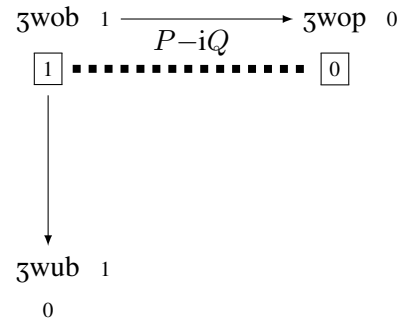
Figure 4: Feeding contains input-provision   Figure 5: Bleeding contains input-removal

The strategy deployed here will also work any time there is more than one locus of application in a string, e.g. in *kike* $\xrightarrow{P}$ *kiki* $\xrightarrow{Q}$ *čiči*. Here, vowel raising (P) increases the string's palatalization (Q) input-rank from 1 to 2, as shown in Figure 6.
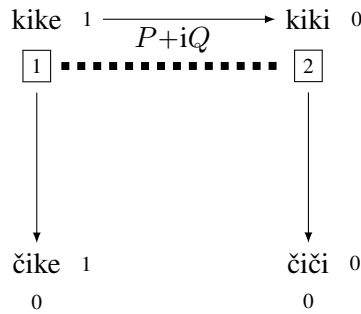
Figure 6: Input-provision with multiple loci

The definitions of input-provision and input-removal as proposed above fit closely with the informal notion of feeding and bleeding, respectively, and produce satisfactory results in almost all cases. Yet, because input-rank only counts applications of a map in a string without tracking specific loci, our definitions produce some awkward consequences under some very special circumstances, not threatening to the overall picture proposed here, as described in this footnote.[13]

The typology we have introduced so far provides a more solid formal foundation to the notions of feeding and bleeding, but it is still not able to distinguish classical feeding from seeding. Consider the Turkish example in (14) again. It appears to be indistinguishable from feeding, as the following figure shows.

---

[13] Consider again $P = \text{V} \longrightarrow \varnothing \: / \: \underline{} \text{V}$ and $Q = s \longrightarrow \int \: / \: \underline{} i$ from fn. 9. There are inputs where P input-provides Q, e.g. *sai*, and inputs where P input-removes Q, e.g. *sia*. So far so good: our definition detects the increase and decrease in input-ranks of these two strings. But what if the input contains both configurations, such as *saisia*? Then P does not change the input-rank of the string, because it increments it up by one and down by one in two distinct loci, resulting in a net change of zero. Yet, we should be able to say there is both feeding and bleeding in this string. To deal with these kinds of cases, we envision the following remedy, only sketched here. The composition of maps P and Q is also a map, call it R. As with any other map, R also imposes NVBs on any string. For input *saisia*, there is an NVB of rank 2, *sai-sia*, because each of these substrings undergoes R nonvacuously: $R(\textit{saisia}) = R(\textit{sai})R(\textit{sia}) = \textit{si}\int\!a$. We can augment the definitions of input-provision and input-removal such that instead of applying them to the whole string, they are applied separately to the NVB substrings for the composition of P and Q when there is an NVB of rank higher than 1. This only arises under very special circumstances: when a map is able to both feed and bleed another map, and when both of those configurations are present in the same string.

$$\text{bebekn} \ \ 1 \xrightarrow{\quad P+\mathrm{i}Q \quad} \text{bebekin} \ \ 0$$

(with boxed 0 connected by dotted line to boxed 1, arrow down to:)
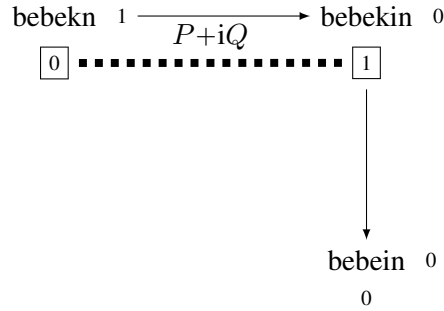
$$\text{bebein} \ \ 0$$
$$0$$

Figure 7: Seeding contains input-provision

The picture is incomplete because we have thus far only considered input interactions – whether strings produced by a rule are possible inputs to another rule. We should also consider and define *output interactions*: whether strings produced by a rule are possible outputs of another rule.

3.5 Output interactions: definitions

In this and following subsections we show that investigating the effect of one rule on the *output* of another brings linguistic insight on a par with investigating effects on inputs. In particular, output interactions unlock the full typology of process interactions, as we will see in §5, and they also play a key role in formally defining opacity, as discussed in §4. Just like input interactions can be thought of in terms of one process P adding members to or removing members from the input set $In(Q)$ of another process Q, output interactions can be conceived of as additions or removals of members of the output set $Out(Q)$. Limitations to the set-based approach for input-interactions were discussed above, and we have pursued a more technical method using the concept of input-rank. Likewise for output interactions: set membership serves as a reasonably good intuitive approximation, but a full formal account must rely on the parallel concept of OUTPUT-RANK.

Just like the input-rank of a string x measures how many times a map potentially applies to x, we can define output-rank to measure how many times *it has potentially applied* to x. Most basically, the output-rank of a string can be defined as the input-rank of its input. Yet there are some subtleties here. First, some strings have no possible inputs with respect to a given map. For example, there is no string that can be input to velar palatalization such that the output is *ki*. We will define the output-rank of such strings to be 0.

Second, strings typically have both vacuous and nonvacuous inputs for a given map. For velar palatalization, the string *či* has two possible inputs, *ki* and *či*, the latter with a vacuous mapping. This vacuous input *či* has an input-rank of 0, while the nonvacuous input *ki* has an input-rank of 1. In such cases our definition will ignore the vacuous mapping and its input-rank, ensuring that a string like *či* has an output-rank of 1 for the velar palatalization map. Sometimes a string may have more than one nonvacuous input with an input-rank of 1, e.g. the string *pat* for the word-final vowel deletion map has inputs *pata*, *pato*, *patu*, etc.

Finally, strings may have more than one nonvacuous input with various ranks. Such is the case for *čiči*, which can be nonvacuously derived from *kiki*, *kiči* or *čiki*, with respective input-ranks of 2, 1, and 1. The upshot is that the output-rank of a string can only be defined for a particular nonvacuous input of that string. This is captured in (25).

(25)   Let P be a map and y a string. Then:

    a.   for any nonvacuous mapping $x \xrightarrow{P} y$, the output-rank of y relative to P and input x, written $\mathsf{OR}(P, x, y)$, is equal to $\mathsf{IR}(P, x)$;

    b.   otherwise, if no nonvacuous mappings $x \xrightarrow{P} y$ exist, the output-rank of y relative to P, written $\mathsf{OR}(P, \epsilon, y)$, is 0.

Once again, parallel to input-rank, output-rank measures how many times a map has potentially applied to a string. We can then define output-provision and output-removal parallel to input-provision and input-removal: when P increases the output-rank of a string x for Q, P output-provides Q (with respect to x), and when P decreases the output-rank of x for Q, P output-removes Q (with respect to x). Parallel to the labels we use for the input interaction atoms, $P+\mathrm{i}Q$ for input-provision and $P-\mathrm{i}Q$ for input-removal, we use the labels $P+\mathrm{o}Q$ and $P-\mathrm{o}Q$ to refer to the output-provision and output-removal atoms, respectively. When the identity or ordering of the processes P and Q is not at issue – as it must be when comparing molecular interaction types with one another – we will often use the abbreviated labels $+\mathrm{i}$, $-\mathrm{i}$, $+\mathrm{o}$, and $-\mathrm{o}$.

In the simplest case, when one string has a Q-output-rank of 0 (and thus has no nonvacuous Q-inputs), while the other string has at least one Q-output-rank of 1 or greater, the increase or decrease of Q's output-rank is equivalent to the addition or removal of a member of the set $Out(Q)$. For example, a word-final vowel raising map applied to the string *če*, producing *či*, increases the output-rank for velar palatalization from 0 (because *če* lacks a nonvacuous palatalization input) to 1 (because *či* has a nonvacuous palatalization input, *ki*). Conversely, a word-final vowel *lowering* map applied to the string *či*, producing *če*, decreases the output-rank for velar palatalization from 1 (because *či* has a nonvacuous palatalization input, *ki*) to 0 (because *či* lacks a nonvacuous palatalization input). This idea is expressed more formally in (26).

(26)   Output interactions in case one string has a Q-output-rank of 0:
   a.   If there is a mapping $x \xrightarrow{P} y$ such that
       i. $\mathsf{OR}(Q, \epsilon, x) = 0$, and
       ii. $\mathsf{OR}(Q, y', y) \geqslant 1$ for some y',
       then P output-provides Q ('$P+\mathrm{o}Q$') at (x, y).
   b.   If there is a mapping $x \xrightarrow{P} y$ such that
       i. $\mathsf{OR}(Q, x', x) \geqslant 1$ for some x', and
       ii. $\mathsf{OR}(Q, \epsilon, y) = 0$,
       then P output-removes Q ('$P-\mathrm{o}Q$') at (x, y).

The example with the interaction of word-final vowel raising (as P) and velar palatalization (as Q) helps to unpack this definition. The situation where raising output-provides palatalization at (*če, či*) is diagrammed in Figure 8. Note that we expand the diagram conventions as follows: output-rank is shown to the left of a string for P-mappings, and above the string for Q-mappings, as illustrated in this footnote.[14]

In Figure 8, the output-rank of *če* for velar palatalization is 0 (the input *če* is vacuous and there is no other possible nonvacuous input). This satisfies clause (i) of the definition of output-provision in (26a). The result of applying raising to *če* is *či*, and this form has an output-rank of 1, because it has a nonvacuous input *ki* for velar palatalization, thus satisfying clause (ii) of the definition in (26a). The 0 and 1 output-ranks are indicated above these forms in the figure.

---

[14] The following schematic diagram illustrates the labeling conventions for strings in our diagrams, showing input-ranks to the right and below a string, and output-ranks above and to the left. To reduce visual clutter in our figures, we will omit irrelevant null input- and output-ranks – that is, any 0 rank for one form corresponding to a 0 rank for another.

$$\mathsf{OR}_Q$$
$$\mathsf{OR}_P \quad \textit{string} \quad \mathsf{IR}_P$$
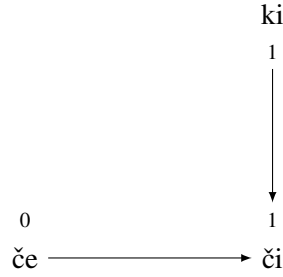$$\mathsf{IR}_Q$$

ki
1

0   1
če ⟶ či

Figure 8: Output-provision

The definition of output-removal (26b) can be illustrated in a similar way by replacing word-final raising with word-final lowering. We leave this as an exercise for the reader.

Turning now to the more complicated situations, consider the same maps applying to strings with possibly multiple loci of application for palatalization, as illustrated in Figure 9 with the same two maps, P = word-final vowel raising and Q = velar palatalization. Here the P-mapping *čiče* ⟶ *čiči* increases the output-rank of Q from 1 to 2. The definition of output-provision should be set up in a way such that this increase would fall under it. However, the same P-mapping also introduces additional Q output-ranks (i.e. additional potential Q-inputs) in a complex way, as illustrated in Figure 9. Setting up the definition of output-provision in these sorts of configurations is tricky; we thus sketch an approach rather than pursuing it with full rigor.

kiče   čiki   kiki   kiči

1   1   2   1
čiče ⟶ čiči
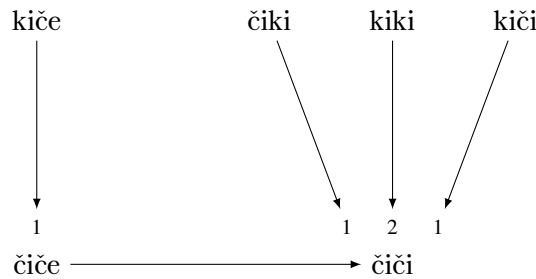
Figure 9: Output-rank increase

Suppose there are two strings x and y such that $P(x) = y$, and each of these strings has some set of output-ranks with respect to another map Q. For example, in Figure 9, string x = *čiče* has a Q-output-rank of 1, while string y = *čiči* has several Q-output-ranks of 1 and 2. There exist *pairings* of the Q-output-ranks of x and of y, or relations that match an output-rank of one string with *at most one* output-rank of another string. In the simpler case where each string only has one output-rank, such a relation is straightforward. In more complex cases such as the example in Figure 9, there are many possible pairings: the output-rank 1 of *čiče* can be paired with any of the three output-ranks of *čiči*.

We suggest that output-interactions hold when none of the possible pairings between the output-ranks of x and y are "well-behaved" in the sense of matching equal output-ranks with each other to the extent possible, and not leaving any output-ranks unmatched. In other words, output-provision (and, *mutatis mutandis*, output-removal) holds when in all pairings of output-ranks, string y contains an output-rank that either does not have a correspondent in string x, or that is higher than its correspondent. For example, for all of the possible output-rank pairings in Figure 9, either or both of these conditions holds: if 1 is paired with 2, there is an output-rank increase, while if 1 is paired with 1, then 2 remains unpaired. As far as we are aware, maps encountered in realistic phonologies of languages have output-rank configurations of the less complicated type that can be covered in (26), or are similar in configuration to Figure 9. In both of those situations, the notion of P's causation of Q-output-rank increase (or decrease) is intuitively sufficiently transparent.

3.6 Output-interactions: exemplification

The linguistic significance of output-provision and output-removal is harder to pin down than the input interactions, so let us turn to some examples.

It turns out that feeding and bleeding contain other atoms besides the input interactions investigated above: each of these molecular interactions also contains an output interaction atom. In the case of feeding, there is output-provision. With reference to the Bedouin Arabic example in (5), this can be seen in the mapping *bidw* $\xrightarrow{P}$ *bidu*. The input *bidw* is not a possible outcome of raising, so it has an output-rank of 0, and no other output rank. The output *bidu*, on the other hand, *is* a possible outcome of raising, with an output-rank of 1 (cf. input *badu*). The input-output pair (*bidw*, *bidu*) thus meets the definition of output-provision in (26a).

In the case of bleeding, there is output-removal. This can be seen in the mapping *ʒwub* $\xrightarrow{P}$ *ʒwup* of the Polish example in (6). The input *ʒwub* is a possible outcome of raising, with an output-rank of 1 (cf. input *ʒwob*), but the output *ʒwup* is *not* a possible outcome of raising, so it has an output-rank of 0 and no other output-ranks. The input-output pair (*ʒwub*, *ʒwup*) thus meets the definition of output-removal in (26b).

In general, in order to determine how P affects Q (that is, which of the four basic interactions $\{P+\mathrm{i}Q, P-\mathrm{i}Q, P+\mathrm{o}Q, P-\mathrm{o}Q\}$ is present), we must compare Q-ranks (input or output) horizontally across a P-arrow. This means taking two forms connected by a P-arrow and comparing (a) the two ranks indicated below those forms (those are the input-ranks) and the two ranks above those forms (those are the output-ranks). If either of those pairs is unequal, one of the four basic interactions is present. Conversely, just as P may affect Q in one of four ways, Q may affect P in the same kinds of ways: $\{Q+\mathrm{i}P, Q-\mathrm{i}P, Q+\mathrm{o}P, Q-\mathrm{o}P\}$. This is determined, *mutatis mutandis*, by comparing P-ranks vertically across a Q-arrow.

In the diagrams in Figures 10 and 11 below, molecular feeding and bleeding interactions are shown to contain pairs of the more basic atoms. In the case of feeding there are two provision atoms $\{P+\mathrm{i}Q, P+\mathrm{o}Q\}$, and in the case of bleeding there are two removal atoms $\{P-\mathrm{i}Q, P-\mathrm{o}Q\}$. There are no other P-on-Q atoms present here, and no Q-on-P atoms at all. Again, unequal ranks that determine the atoms are boxed and connected by thick dashed lines, and irrelevant null ranks are omitted to reduce visual clutter.[15]
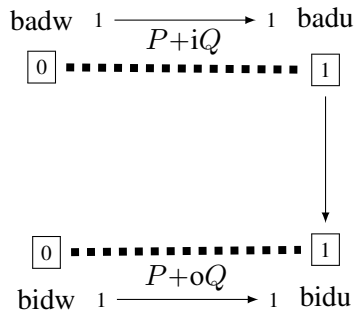

Figure 10: Feeding, $\{P+\mathrm{i}Q, P+\mathrm{o}Q\}$
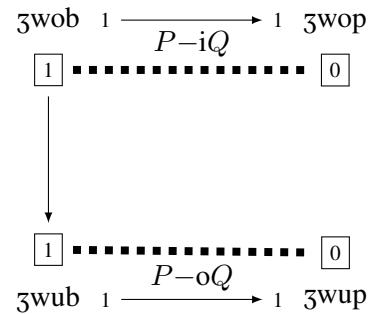
Figure 11: Bleeding, $\{P-\mathrm{i}Q, P-\mathrm{o}Q\}$

We take the configurations illustrated in Figures 10 and 11 to be definitional of feeding and bleeding, as follows. In these definitions and below, we flatten the two-dimensional diagrams in the figures into equivalent one-dimensional representations, such as $a \xrightarrow{P} b \xrightarrow{Q} c \xleftarrow{P} d$ in the case of feeding.

---

[15] The careful reader may have noted that the bleeding diagram in Figure 11 resembles our example of multiple loci of application in Figure 6, but this resemblance is superficial because the details of input-ranks and output-ranks are different in the two examples, and thus Figure 6 is not an example of bleeding.

(27)   a. P feeds Q if there exist strings $\{a, b, c, d\}$ such that $a \xrightarrow{P} b \xrightarrow{Q} c \xleftarrow{P} d$,
         P input-provides Q at (a, b), and P output-provides Q at (d, c).

      b. P bleeds Q if there exist strings $\{a, b, c, d\}$ such that $b \xleftarrow{P} a \xrightarrow{Q} c \xrightarrow{P} d$,
         P input-removes Q at (a, b), and P output-removes Q at (c, d).

      In such cases we will say that P feeds or bleeds Q at (a, b, c, d).

It is worth emphasizing that the discussion here is about the formal structure of interactions, and not about what actually happens in a particular language. Whether or not Bedouin Arabic *in actuality* has an underlying lexical representation like /bidw/ is irrelevant at the level of idealization assumed here. This holds of all subsequent examples where possible/potential as opposed to actual forms are placed in these diagrams in order to properly classify process interactions given a set of forms to which the processes apply.

Let's now consider the Turkish seeding example in (14) from the point of view of output interactions. Recall from Figure 7 that P (vowel epenthesis into a final cluster) input-provides Q (intervocalic velar deletion), i.e. P+iQ. However, seeding is different from feeding in that P does not also output-provide Q. Indeed, $Out(Q)$ includes only strings containing hiatus that potentially arose due to intervocalic velar deletion, and P does not itself create new instances of hiatus. Yet, P+iQ is not the only interaction here: Q also output-removes P. Indeed, Q takes a string containing VkiC, which is a possible output of P, and outputs the string ViC, which is not a possible output of P. The diagram in Figure 12 illustrates.
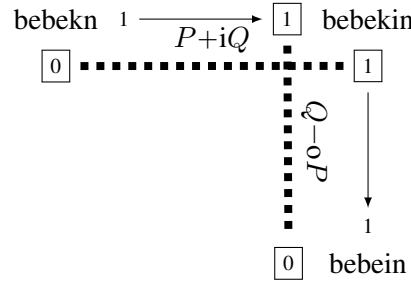
Figure 12: Seeding, $\{P\text{+i}Q, Q\text{−o}P\}$

Seeding thus shares formal characteristics (atoms) with both feeding and bleeding: like feeding, it has input-provision, and like bleeding, it has output-removal. As we will see, this formal atom-sharing with both feeding and bleeding is what gives seeding the mixed feeding-like behavior with counterbleeding-like opacity. More generally, this is the justification for admitting output interactions on a par with input interactions into our formal toolbox. Again, the configuration illustrated in Figure 12 is definitional of seeding, as follows.

(28)   a. P seeds Q if there exist strings $\{a, b, c\}$ such that $a \xrightarrow{P} b \xrightarrow{Q} c$,
         P input-provides Q at (a, b), and Q output-removes P at (b, c).

      In such cases we will say that P seeds Q at (a, b, c).

4   Opacity

4.1   Recap: atoms and molecules

We have "atomized" complex molecular interaction types into more basic elements like provision and removal of inputs and outputs. We claim that, more generally, the typology of interactions between two maps P and Q is best understood by examining these atoms.

Recall that, strictly speaking, atoms are properties of individual mappings. Thus, any pair of maps P, Q may contain many atoms. There are four basic atom types: $+i$ (input-provision), $-i$ (input-removal), $+o$ (output-provision), and $-o$ (output-removal). In the context of actual maps, any one atom may occur in either direction: P may input-provide Q, or Q may input-provide P. The molecular relations "P feeds Q", "P bleeds Q", and "P seeds Q" are reflections of particular atoms contained in some mappings of P and Q, as summarized in the following table. Not only the identity of the atoms ($+i$, $+o$, etc.), but also the directionality matters, since relations like "P feeds Q" are directional.

(29)

atoms / molecules	$X{+}iY$	$X{+}oY$	$X{-}iY$	$X{-}oY$
P feeds Q	$P{+}iQ$	$P{+}oQ$		
P bleeds Q			$P{-}iQ$	$P{-}oQ$
P seeds Q	$P{+}iQ$			$Q{-}oP$

The presence of these atoms can be observed at a level of abstraction where the actual order of application of the maps ($P > Q$ or $Q > P$) is irrelevant: the fact that $P{+}iQ$ is an atom is simply a matter of the structure of the maps, not how they apply in a specific language. However, at a more concrete level, in a given grammar of a language one of two orders must hold, $P > Q$ or $Q > P$. Some of these orders result in opacity: in particular, using the examples in (29), if P feeds Q, and Q applies first, there is underapplication opacity; if P bleeds Q and Q applies first, there is overapplication opacity, and if P seeds Q, both orders are opaque.

This allows us to arrive at a provisional characterization of these two types of opacity, as follows. The two atoms that give rise to opacity are input-provision ($+i$) and output-removal ($-o$); opacity holds if the map that is input-provided or output-removed applies first.

(30)   For two maps X and Y, when $X > Y$,

    a. X underapplies if $Y{+}iX$;

    b. X overapplies if $Y{-}oX$.

The decomposition of molecules into atoms gives a ready answer to why (counter)seeding resembles both (counter)feeding and (counter)bleeding: seeding shares one atom type ($+i$, input-provision) with feeding, and another ($-o$, output-removal) with bleeding. The upshot of these observations is that both forms of opacity result from particular interaction atoms. Nevertheless, opacity is not an entirely unified phenomenon because underapplication and overapplication result from the presence of distinct atoms. If P creates inputs (i.e. increases the input-rank of some form) for Q but Q fails to apply to that new input opportunity, then there is underapplication opacity ($P{+}iQ$, with Q applying first). This is what happens with counterfeeding and counterseeding: if Q applies first, then there is underapplication. Likewise with overapplication: if a form's Q-output-rank decreases as a result of the application of P, but Q has still applied despite the removal of the conditions for its application, then there is overapplication ($P{-}oQ$, with Q applying first). This is what happens with counterbleeding: if Q applies first, then there is overapplication. Likewise with seeding, but with P and Q reversed: since $Q{-}oP$, if P applies first, then there is overapplication.

4.2   Accommodating misapplication

Consider now the interaction type we called *transfusion* in §2.3, related to a third form of opacity we called *misapplication*. In those cases, application of the first of two processes has an unexpected surface result due to the application of the second process, such as stress ending up on the 'wrong' syllable (7) or the

'wrong' copy-vowel being epenthesized (8). Misapplication is neither underapplication nor overapplication, and transfusion is neither feeding nor bleeding. So where do these examples belong in our typology?

Informally, misapplication and overapplication share something in common: both result in a surface form to which a rule has applied but is inconsistent in some way with the usual or expected application of that rule. The difference between misapplication and overapplication is whether the alternative, transparent derivation is vacuous or not. It is vacuous in the case of overapplication ("the rule should not have applied but did"), but nonvacuous in the case of misapplication ("the rule should have applied in a different way than it did"). Formally, misapplication turns out to have the same signature as overapplication: the $-o$ atom. Let us return to the Palestinian Arabic example in (7) with the tools developed here. Recall that P is vowel epenthesis, and Q is the stress rule; the actual order in the language is $Q > P$.
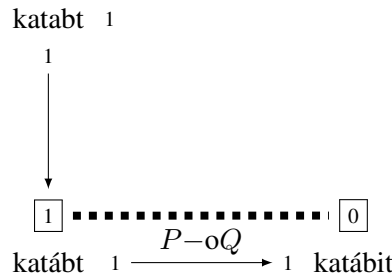
Figure 13: Transfusion, output-removal

The $-o$ atom here can now be seen as the signature of both misapplication and overapplication. Using the term *misapplication* to cover both types of opacity, with overapplication being a special case, we arrive at the following generalized characterization of opacity.

(31)   For two maps X and Y, when $X > Y$,

    a. X underapplies if $Y+iX$;

    b. X misapplies if $Y-oX$.

Thus, when $Q > P$ and $P-oQ$ in Palestinian Arabic, Q (the stress rule) is expected to misapply, which is exactly what is found in outputs like *katábit*. For completeness, and anticipating the discussion of the fuller typology in §5, note that $-o$ is not the only atom present in the Palestinian Arabic example because there are other connected forms not included in Figure 13. For example, *katabt* is an input to epenthesis (as indicated by its P-input-rank of 1), producing *katabit*, which in turn, like all unstressed forms, is an input to stress, producing *kátabit*, which in turn is a possible output of epenthesis, from input *kátabt*. The latter mapping, *kátabt* $\xrightarrow{P}$ *kátabit*, also shows that epenthesis output-provides stress: *kátabt* could not have been produced by the stress rule, but *kátabit* could.[16]

---

[16] A reviewer points out that in many analyses of such stress systems, already-stressed forms undergo the stress rule, e.g. input *kátabt* would map to *katábt*. The addition of these mappings would make the picture in Figure 14 more complex but does not change the inventory of atoms contained in it.
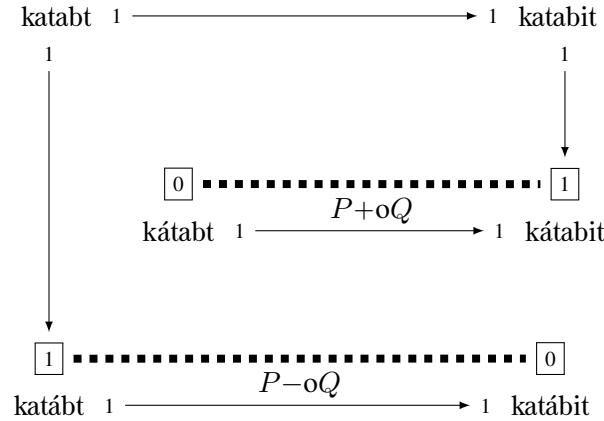
Figure 14: Transfusion, $\{P-\mathrm{o}Q, P+\mathrm{o}Q\}$

This example illustrates the utility of the rank-based approach. If we instead attempted to keep track of interaction by segmental changes, we would be forced to say that epenthesis in *katabt* feeds antepenultimate stress assignment and bleeds final stress assignment, but this statement is linguistically unsatisfactory and misses the point that epenthesis can reconfigure a form in a way that affects the location of stress. Stress still applies regardless of epenthesis, and this is captured by the fact that epenthesis is powerless to change the input-rank of stress, but the irregular placement of stress in forms like *katábit* is captured by the fact that epenthesis can change the output-rank of stress.

4.3   Misapplication vs. overapplication

Both misapplication and overapplication occur in situations where $Q > P$ and $P-\mathrm{o}Q$. Which of these two opaque patterns a given interaction displays does not depend on the structure of the interaction but on the nature of the processes involved. In our examples, typically bleeding and seeding display overapplication while transfusion displays the complementary form of misapplication, but this is not necessarily always the case. A hypothetical example is given in (32), slightly modified from Palestinian Arabic by restricting epenthesis to apply only after stressed vowels. This seeding interaction displays misapplication.

(32) P: Stress superheavy final, else heavy penult, else antepenult   (stress)

Q: $\varnothing \longrightarrow$ i / V́C ___ C#   (modified epenthesis)

$(P > Q)$: /katabt/ $\xrightarrow{P}$ katábt $\xrightarrow{Q}$ katábit   (misapplication of stress)

$(Q > P)$: /katabt/ $\xrightarrow{Q}$ katabt $\xrightarrow{P}$ katábt   (underapplication of epenthesis)

Stress seeds epenthesis in this example. The later application of epenthesis moves stress to an unexpected location, i.e. with misapplication because epenthesis output-removes stress, as shown in Figure 15.
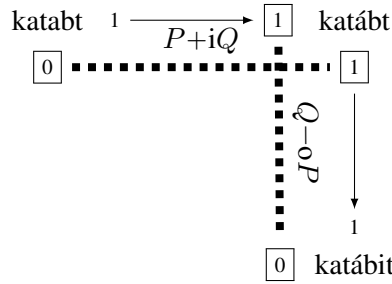
Figure 15: Seeding with misapplication

As we explained earlier, in the situation where $Q{-}\mathrm{o}P$ and $P > Q$, there is either misapplication or overapplication opacity, and the difference between misapplication and overapplication is about the alternative: what would P have done if Q had applied first? We have overapplication when P would not have applied if Q applied first, and we have misapplication when P would have still applied, but in a different way. Thus, whether a given interaction displays misapplication or overapplication depends on whether both the input and the output of Q are configurations to which P is differently applicable. The reason there is misapplication of stress in Palestinian Arabic (7) as well as in (32) is because stress is applicable regardless of whether epenthesis shifts the expected locus of stress. Similarly, vowel copy epenthesis in Yokuts (8) is applicable regardless of whether the vowel to be copied is lowered.

4.4 Misapplication on focus

Another, less intuitive variation on misapplication is found in some cases of interactions between rules affecting the same segment (so-called 'on-focus' interactions; McCarthy 1999). For example, consider the following Polish example (Rubach 1984; Łubowicz 2002). Velar stops palatalize (and affricate) before front vowels (Q), and voiced affricates spirantize (P).[17] The correct $Q > P$ derivation is an example of seeding. Q input-provides P by supplying new voiced affricates, while P in turn output-removes Q because $\widehat{d\textrm{ʒ}}$ but not $\textrm{ʒ}$ is a possible output of palatalization. As expected, the reverse order $P > Q$ exhibits underapplication of P, with possible outputs like $\widehat{d\textrm{ʒ}}$ that look like they should have undergone spirantization but haven't.

(33) P: $\begin{bmatrix} +\text{voice} \\ +\text{del. rel.} \end{bmatrix} \longrightarrow [+\text{cont}]$ (spirantization)

       Q: $[+\text{dorsal}] \longrightarrow \begin{bmatrix} -\text{back} \\ +\text{del. rel.} \end{bmatrix} / \underline{\hspace{1em}} \begin{bmatrix} -\text{cons} \\ -\text{back} \end{bmatrix}$ (palatalization)

         $(P > Q)$: /vagi$\widehat{\textrm{tɕ}}$/ $\xrightarrow{P}$ vagi$\widehat{\textrm{tɕ}}$ $\xrightarrow{Q}$ va$\widehat{\textrm{dʒ}}$i$\widehat{\textrm{tɕ}}$ (counterseeding, P underapplies)

    ✓$(Q > P)$: /vagi$\widehat{\textrm{tɕ}}$/ $\xrightarrow{Q}$ va$\widehat{\textrm{dʒ}}$i$\widehat{\textrm{tɕ}}$ $\xrightarrow{P}$ va$\textrm{ʒ}$i$\widehat{\textrm{tɕ}}$ 'weigh.INF'   (seeding, Q overapplies)

Misapplication of palatalization, resulting from the $Q{-}\mathrm{o}P$ atom, is expected here by (31). Compared to the Palestinian Arabic transfusion example in (7) and to the Turkish seeding example in (14), however, it is somewhat more difficult to grasp the sense in which P has misapplied here. Misapplication is 'wrong' application in the context of an expected alternative. With outputs like $\textrm{ʒ}i$ and the knowledge that palatalization has in fact applied, misapplication can be construed in terms of the expected alternative outcome of palatalization producing affricates from input stops. Instead, the output is a fricative, lacking the expected [–cont] value.

---

[17] In the actual Polish case there is also nonderived environment blocking of spirantization, but this is orthogonal to questions of ordering. Nonderived environment blocking is the suspension of otherwise expected ordering effects in a particular (well-defined) class of cases (Baković 2011:51), whether feeding or, in this case, seeding.

5 Typology

We have thus far demonstrated that four interaction types, *feeding*, *bleeding*, *seeding*, and *transfusion* are in fact composite molecules, consisting of different combinations of elementary atoms $\{+\mathrm{i}, -\mathrm{i}, +\mathrm{o}, -\mathrm{o}\}$. Given an order $X > Y$ between two rules X, Y, those molecules with the atom $Y+\mathrm{i}X$ exhibit underapplication of X, while those with $Y-\mathrm{o}X$ exhibit misapplication of X (or its more specific instantiation, overapplication). The next step is to flesh out the full typology of combinations of atoms defining possible molecular interactions. In this section we undertake a bottom-up investigation of this typology.[18]

5.1 Corners

Minimally, an interaction may arise when two mappings from different maps meet at a single string. Let us call such arrangements CORNERS – strings where two mapping arrows meet, one arrow from map P and the other from map Q. Let us consider the simplest case where all ranks are either 0 or 1, and there is at most one unmatched output-rank in every case (so there is only one instance of an output interaction). With two arrows from two maps, there are four possible types of corners, because each arrow can meet the corner at either one of its ends. The four corners are illustrated in Figures 16–19.
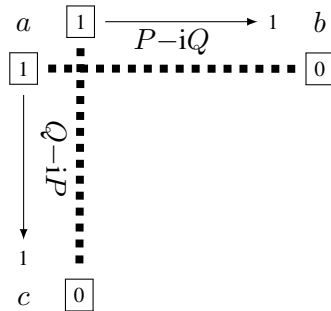
Figure 16: Upper left corner
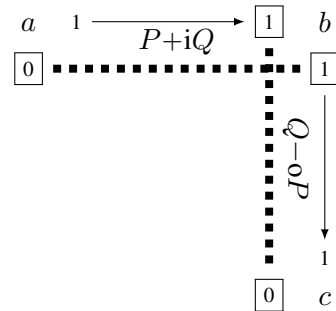
Figure 17: Upper right corner
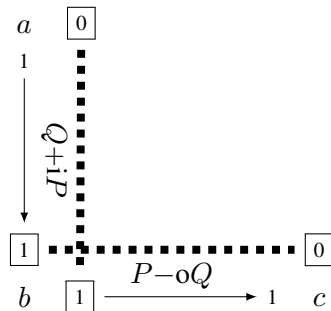
Figure 18: Lower left corner
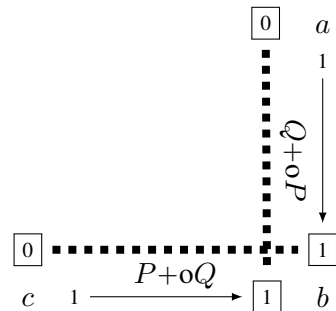
Figure 19: Lower right corner

We must make one point for presentational clarity, following a reviewer's suggestion. The reader may recall that Figures 4 and 5 in §3.4 also have the visual shapes of corners, but do not contain two atoms each. This appears to contradict our claim that the shapes in Figures 16–19 and atoms contained therein are the only possible corner-shaped interactions. The reason for this apparent contradiction is that, for expository purposes, in Figures 4 and 5 we omitted some forms and the arrows connecting them to other forms. We

---

[18] The top-down task of exhaustively characterizing the universe of possible molecules for all possible map pairs – a task of some mathematical complexity – is not attempted in this paper.

subsequently made those figures complete by introducing other arrows in Figures 10 and 11 in §3.6, and it turns out that the correct shape of those interactions is not a corner. More generally, in this section we examine interaction configurations by including all the arrows (i.e. mappings) that can possibly be included in the graph; so, e.g. in Figures 16–19 and in the examples that illustrate them below it is not possible to add any more arrows originating or ending in a, b, or c, and likewise for the more complex shapes below.

Returning to the typology, Figures 16–19 serve as visual aids to finding atoms in more complex configurations. Every atom results from a mismatch between a 1 and a 0 rank, and those mismatches are always located on one of the arrows (P or Q) at one of the four corners. An efficient way to identify the atom present in an arrow is to find it and the rank mismatches in one of these corner figures. Each of these corners contains two atoms. The upper right corner (Figure 17) and lower left corner (Figure 18), equivalent up to names of P and Q, instantiate seeding, with input-provision in one direction and output-removal in the other.

There are two other topologically distinct corners besides seeding. The first is the upper left corner (Figure 16), which instantiates MUTUAL BLEEDING and contains two atoms: $\{P{-}iQ,\ Q{-}iP\}$. Mutual bleeding is exemplified by a pattern in Russian (Kenstowicz & Kisseberth 1979), where dental stops delete before word-final consonants (P) and laterals delete word-finally after a consonant (Q). Input-removal is not diagnostic of opacity, consistent with the fact that both orders of P and Q here are transparent.

(34)　P: $\{\text{t, d}\} \longrightarrow \varnothing\ /\ \_\_\_ \text{C\#}$　　　　　　　　　　　　　　(dental stop deletion)

　　　Q: $\text{l} \longrightarrow \varnothing\ /\ \text{C} \_\_\_ \#$　　　　　　　　　　　　　　　　　(lateral deletion)

　　　$\checkmark (P > Q)$: /metl/ $\xrightarrow{P}$ mel $\xrightarrow{Q}$ mel 'sweep.NPFV.PST'　　　　　(bleeding, transparent)

　　　$(Q > P)$: /metl/ $\xrightarrow{Q}$ met $\xrightarrow{P}$ met　　　　　　　　　　　　(bleeding, transparent)
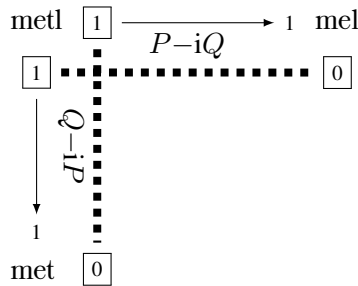

Figure 20: Mutual bleeding, $\{P{-}iQ,\ Q{-}iP\}$

The final corner is the lower right corner (Figure 19), containing the atoms $\{P{+}oQ,\ Q{+}oP\}$. We call this MERGER: two separate rules resulting in the same output in some cases. Consider first a case of merger based on examples from Maasai (Tucker & Mpaayei 1955; Archangeli & Pulleyblank 1994). Left-to-right ATR harmony (P) targets all vowels while right-to-left ATR harmony (Q) only targets non-low vowels.[19]

(35)　P: $\text{V} \longrightarrow [+\text{ATR}]\ /\ [+\text{ATR}]\ \text{C}_0 \_\_\_$　　　　　　　　　　　　　(L→R harmony)

　　　Q: $[-\text{low}] \longrightarrow [+\text{ATR}]\ /\ \_\_\_ \text{C}_0\ [+\text{ATR}]$　　　　　　　　　　(R→L harmony)

　　　$\checkmark (P > Q)$: /duŋ-ɪʃɔ-re/ $\xrightarrow{P}$ duŋ-iʃo-re $\xrightarrow{Q}$ duŋ-iʃo-re 'cut-INTR-APPL'　　　(merger)

　　　$\checkmark (Q > P)$: /duŋ-ɪʃɔ-re/ $\xrightarrow{Q}$ duŋ-iʃo-re $\xrightarrow{P}$ duŋ-iʃo-re　　　　　　　(merger)

---

[19]Low vowels targeted by left-to-right harmony are ultimately raised to [o], though this detail is irrelevant here.

Note that the two directional harmony rules P and Q in (35) result in the same output in the example given, regardless of their order. Whether P applies first, spreading [+ATR] rightward from the root vowel /duŋ/, or Q applies first, spreading [+ATR] leftward from the final suffix vowel /-re/, the vowels of the medial suffix /-ɪʃɔ/ surface correctly as [-iʃo]. This shows that the order of application of P and Q makes no difference, and so by our definition in §2.2 that an interaction holds iff $P(Q(x)) \neq Q(P(x))$, a merger is not strictly speaking an interaction.[20] Even though mergers don't involve interaction, they are within our formal typology and so we will proceed with some further discussion of them.

Note that the input sets of P and Q in (35) intersect, meaning that there exist inputs like /duŋ-ɪʃɔ-re/ to which both rules are nonvacuously applicable. But because the rules do not interact, both orders converge on the same output simply because the first rule to apply to a member of the intersection between their input sets leads directly to that output. Now consider the hypothetical example of merger in (36) involving two rules with *non*intersecting input sets, meaning that the rules are not applicable to any of the same inputs. High vowels are lowered to mid after uvulars (P) and low vowels are raised to mid when followed by high vowels (Q). Again, either order of P and Q yields the same output, but in this case because one rule or the other is not applicable at all in the derivation, depending on the input. This is indicated with a slash through the inapplicable rule in each of the hypothetical derivations (e.g. $\not{\!Q}$) shown in (36).

(36) P: $\begin{bmatrix} +\text{syll} \\ -\text{low} \end{bmatrix} \longrightarrow [-\text{high}] \;/\; \begin{bmatrix} +\text{cons} \\ -\text{high} \end{bmatrix} \text{———}$ (lowering)

       Q: $\begin{bmatrix} +\text{syll} \\ -\text{high} \end{bmatrix} \longrightarrow [-\text{low}] \;/\; \text{———} \, C_0 \begin{bmatrix} +\text{syll} \\ +\text{high} \end{bmatrix}$ (raising)

    $(P > Q)$: /qiti/ $\xrightarrow{\;P\;}$ qeti $\xrightarrow{\;\not{\!Q}\;}$ qeti; /qati/ $\xrightarrow{\;\not{\!P}\;}$ qati $\xrightarrow{\;Q\;}$ qeti   (merger, no interaction)

    $(Q > P)$: /qiti/ $\xrightarrow{\;\not{\!Q}\;}$ qiti $\xrightarrow{\;P\;}$ qeti; /qati/ $\xrightarrow{\;Q\;}$ qeti $\xrightarrow{\;\not{\!P}\;}$ qeti   (merger, no interaction)

The distinction between these two types of mergers turns out to be significant. In the hypothetical case of merger with nonintersecting input sets (36), there is only mutual output provision, P+oQ and Q+oP: each rule results in some outputs that the other rule would result in (albeit from a different input), and nothing more. This is shown on the right in Figure 21 below. In the Maasai-based case of merger with intersecting input sets (35), however, there is *both* mutual output provision *and also* mutual input removal, P−iQ and Q−iP. Both rules are applicable to the input /duŋ-ɪʃɔ-re/, and both rules generate from that same input the same output [duŋ-iʃo-re]. So when either rule accomplishes a mapping like this one, it accomplishes what any rule ordinarily does unto itself: remove an input and provide an output. This is shown on the left in Figure 21.
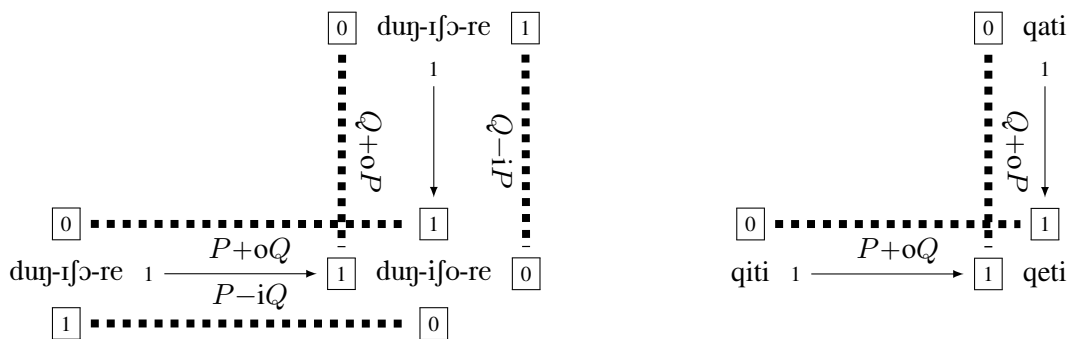

Figure 21: Intersecting input set merger (left), $\{P\text{+o}Q, Q\text{+o}P, P\text{−i}Q, Q\text{−i}P\}$;
Nonintersecting input set merger (right), $\{P\text{+o}Q, Q\text{+o}P\}$

[20]See Meinhardt, Mai, Baković & McCollum (2024) for specific discussion of the computational consequences of the lack of an interaction between (directional subsequential functions corresponding to) the Maasai rules in (35).

Both of these cases phenomenologically represent merger, but only the second, nonintersecting input set case is a corner. The intersecting input set case is more like what we call a *compound interaction*; see §5.6.

To represent the topology of the graphs, we can flatten the seeding graph to $\cdot\rightarrow\cdot\rightarrow\cdot$, where the dot '$\cdot$' stands for a string, and each arrow represents a nonvacuous mapping. Mutual bleeding thus takes the shape $\cdot\leftarrow\cdot\rightarrow\cdot$, and $\cdot\rightarrow\cdot\leftarrow\cdot$ represents merger. The table in (37) summarizes the discussion of corners thus far, including merger but ignoring the (non)intersecting input set distinction just discussed.

(37)

arrows	shape	interactions	Fig.	structure	atoms
2	corners	seeding	17	$\cdot\rightarrow\cdot\rightarrow\cdot$	$\{P{+}iQ, Q{-}oP\}$
		mutual bleeding	20	$\cdot\leftarrow\cdot\rightarrow\cdot$	$\{P{-}iQ, Q{-}iP\}$
		merger	21	$\cdot\rightarrow\cdot\leftarrow\cdot$	$\{P{+}oQ, Q{+}oP\}$

A final note on why merger is the only case that is not strictly speaking an interaction, in that the order of applying P and Q does not matter for any input. The three topologically distinct corner configurations are $a \xrightarrow{P} b \xrightarrow{Q} c$ (seeding), $a \xleftarrow{P} b \xrightarrow{Q} c$ (mutual bleeding), and $a \xrightarrow{P} b \xleftarrow{Q} c$ (merger). For seeding and mutual bleeding, order matters for at least one input. For seeding, $b = P(Q(a)) \neq Q(P(a)) = c$. For mutual bleeding, $c = P(Q(b)) \neq Q(P(b)) = a$. But for merger, applying P and Q to any of a, b, c in the configuration $a \xrightarrow{P} b \xleftarrow{Q} c$ results in the same output regardless of order (that output is always b). Thus, by inspection, merger is the only corner configuration that is not an "interaction" in the sense defined above.

5.2　Pies

Feeding and bleeding are structurally more complex than corners in that they minimally contain three, as opposed to two, nonvacuous mapping arrows. They have already been shown above in Figures 10 and 11, repeated here in Figures 22 and 23, respectively, with irrelevant null ranks omitted.
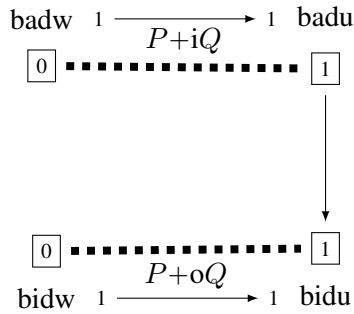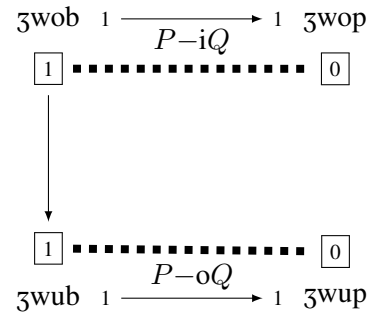

Figure 22: Feeding, $\{P{+}iQ, P{+}oQ\}$　　　　Figure 23: Bleeding, $\{P{-}iQ, P{-}oQ\}$

These configurations will be referred to as PIES, given their similarity to the shape of the Greek letter Π. They can be flattened to $\cdot\rightarrow\cdot\rightarrow\cdot\leftarrow\cdot$ and $\cdot\leftarrow\cdot\rightarrow\cdot\rightarrow\cdot$, respectively, and there are only two such pies – all other sequences of three arrows are topologically equivalent.

(38)

arrows	shape	interactions	Fig.	structure	atoms
2	corners	seeding	17	$\cdot\rightarrow\cdot\rightarrow\cdot$	$\{P{+}iQ, Q{-}oP\}$
		mutual bleeding	20	$\cdot\leftarrow\cdot\rightarrow\cdot$	$\{P{-}iQ, Q{-}iP\}$
		merger	21	$\cdot\rightarrow\cdot\leftarrow\cdot$	$\{P{+}oQ, Q{+}oP\}$
3	pies (Πs)	feeding	22	$\cdot\rightarrow\cdot\rightarrow\cdot\leftarrow\cdot$	$\{P{+}iQ, P{+}oQ\}$
		bleeding	23	$\cdot\leftarrow\cdot\rightarrow\cdot\rightarrow\cdot$	$\{P{-}iQ, P{-}oQ\}$

While it appears that $\cdot\rightarrow\cdot\rightarrow\cdot\rightarrow\cdot$ should be a distinct structure, it is not, as the graph in Figure 24 should make clear. This configuration simply consists of two corners, the upper right (cf. Figure 17; P seeds Q) and lower left (cf. Figure 18; Q seeds P). It contains two pairs of atoms, each pair contributed by a separate seeding interaction. This is the structure of both FED COUNTERFEEDING (Kavitskaya & Staroverov 2010) and DUKE-OF-YORK DERIVATIONS (Pullum 1976; McCarthy 2003; Norton 2003). More generally, the structures under consideration as basic molecular interaction types are substrings of $\cdot\rightarrow\cdot\rightarrow\cdot\leftarrow\cdot\leftarrow\cdot$ repeated periodically, i.e. paths along the square. The structure $\cdot\rightarrow\cdot\rightarrow\cdot\rightarrow\cdot$ in Figure 24 is not such a substring. Instead, it is an example of a COMPOUND interaction (see §5.6), which combine two or more basic molecules.
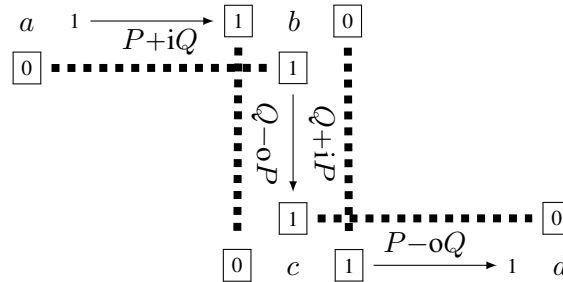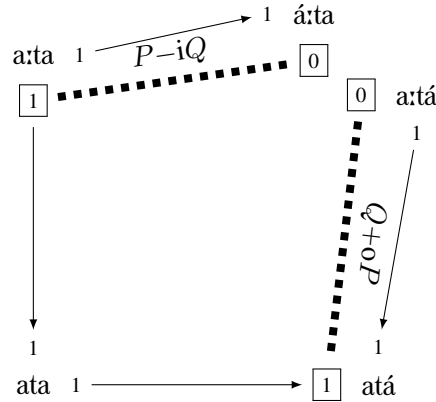

Figure 24: Two-corner compound interaction

The fact that the basic configurations we are considering here are substrings of $\cdot\rightarrow\cdot\rightarrow\cdot\leftarrow\cdot\leftarrow\cdot$ repeated periodically also makes clear that merger remains the only such structure where there is no "interaction" in the strict sense. Any path along the square larger than merger will include at least one corner other than the lower-right merger corner as a substructure, and thus the order of application of P and Q will matter for at least one input. The only configurations where order does not matter are basic or compound configurations whose only substructure is the lower right corner, such as those in Figure 21 instantiating merger.

5.3 Open jaws

The next example of a "basic" interaction is one step more complex than a pie. If an arrow is added at either end of a pie – or if arrows are added at both ends of a corner – but without closing the square, the structure becomes what we will refer to as an OPEN JAW. For a concrete illustration, Figure 25 shows an open-jaw interaction between the following two hypothetical processes.

(39) P: L-to-R quantity-sensitive iamb (initial if long, else peninitial)   (stress)

   Q: unstressed aː $\longrightarrow$ a / # ___   (shortening)

   $(P > Q)$: /aːta/ $\xrightarrow{P}$ áːta $\xrightarrow{Q}$ áːta   (transparent)

   $(Q > P)$: /aːta/ $\xrightarrow{Q}$ ata $\xrightarrow{P}$ atá   (transparent)

29

Figure 25: Hyper seeding, $\{P{-}\mathrm{i}Q,\ Q{+}\mathrm{o}P\}$

We refer to this open jaw interaction as HYPER SEEDING because it includes as a substructure the seeding structure $\cdot{\to}\cdot{\to}\cdot$ in the lower left corner. Hyper seeding has the structure of a square with the upper right corner not closed. Topologically equivalent up to the names P and Q is a square with the bottom left corner not closed, including the seeding structure in the upper right corner as a substructure. Unlike seeding, hyper seeding is transparent no matter the order of the maps. If stress applies first, it prevents shortening from applying; if shortening applies first, stress is assigned to the peninitial vowel instead of to the initial vowel. This mutual transparency is reflected by the absence of the opacity atoms $+\mathrm{i}$ and $-\mathrm{o}$ in Figure 25.

Additionally, there are two other topologically distinct open jaws: one where the lower right corner is open, and another where the upper left corner is open. These are illustrated in Figures 26 and 27. The atoms of these open jaws are symmetrical with respect to P and Q. Figure 26 represents mutual output removal, and includes as a substructure the mutual bleeding structure $\cdot{\leftarrow}\cdot{\to}\cdot$ in the upper left corner; we will thus refer to it as HYPER MUTUAL BLEEDING. Figure 27 represents mutual input provision, and includes as a substructure the merger structure $\cdot{\to}\cdot{\leftarrow}\cdot$ in the lower right corner; we will thus refer to it as HYPER MERGER.
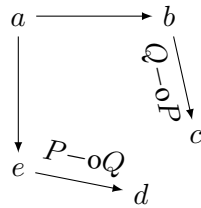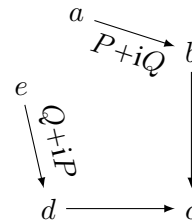
Figure 26: Hyper mutual bleeding　　　　　　　　　　　Figure 27: Hyper merger

A case of hyper mutual bleeding is evident based on examples from Dakota (Shaw 1980, 1985; Kennedy 1994; Alderete 1999; Pruitt 2023) with the rules in (40).[21] Stress falls on the second syllable of polysyllables, else initially (P), and a low vowel is epenthesized after a word-final consonant (Q). (There are no coda consonants in Dakota so epenthesis is probably syllabically conditioned; this detail is irrelevant here.)

---

[21] Adopting terminology from Rasin (2022), Pruitt (2023:503) classifies the interaction in (40) as a case of *counter-shifting*, a term that also covers the type of stress-epenthesis interaction seen in Palestinian Arabic (7) that we have chosen here to call *transfusion*. Collapsing the distinction between these cases is appropriate for the specific purposes of Rasin (2022) and Pruitt (2023), but not for ours: both orders of the rules in a hyper mutual bleeding interaction involve misapplication opacity, whereas only one of the orders of the rules in a transfusion interaction does.

(40) P: L-to-R iamb (peninitial if polysyllabic, else initial)   (stress)

   Q: $\varnothing \longrightarrow$ a / C ___ #   (epenthesis)

   $\checkmark (P > Q)$: /ček/ $\xrightarrow{P}$ ček $\xrightarrow{Q}$ čéka 'stagger'   (P misapplies)

   $(Q > P)$: /ček/ $\xrightarrow{Q}$ čeka $\xrightarrow{P}$ čeká   (Q misapplies)

Unlike mutual bleeding, this interaction is opaque no matter the order of the maps. The correct $P > Q$ order renders stress opaque, because stress ends up on the initial syllable despite there being an eligible peninitial syllable. With the opposite (incorrect) $Q > P$ order, epenthesis is opaque in the technical sense explained in §4.4: the epenthetic vowel is ultimately stressed despite the fact that it was inserted as a stressless vowel. The misapplication opacity of both orders is reflected by the two −o atoms in Figure 28.
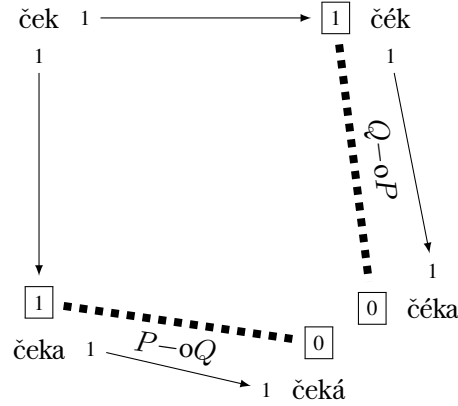
Figure 28: Hyper mutual bleeding, $\{P{-}\text{o}Q, Q{-}\text{o}P\}$

In (41) we have constructed a hypothetical example of hyper merger, the diagram in Figure 27 with mutual input provision atoms $\{P{+}\text{i}Q, Q{+}\text{i}P\}$. (For simplicity, we focus only on the featural changes of these two rules, ignoring their possible contexts of application.) There are two crucial aspects of the two rules in (41) that must be pointed out. The first is that the voicing rule (P) applies only to [+del. rel.] obstruents, targeting fricatives $\{s, z\}$ and affricates $\{\widehat{t\!f}, \widehat{d\!\!\!\!3}\}$ to the exclusion of stops $\{t, d\}$. The second is that the affrication rule (Q) applies only to obstruents that share [±voi] and [±cont] values, targeting $\{s, t\}$ to the exclusion of $\{z, d\}$. Given an input /s/, then, the $P > Q$ order produces a transparent, feeding-like derivation while the $Q > P$ order produces a counterfeeding-like derivation with underapplication opacity. Given an input /t/, we have exactly the opposite: $P > Q$ is opaque, and $Q > P$ is transparent.

(41) P: $\begin{bmatrix} -\text{son} \\ +\text{del. rel.} \end{bmatrix} \longrightarrow [+\text{voi}]$ (voicing)

   Q: $\begin{bmatrix} -\text{son} \\ \alpha\text{voi} \\ \alpha\text{cont} \end{bmatrix} \longrightarrow \begin{bmatrix} +\text{del. rel.} \\ -\text{cont} \end{bmatrix}$ (affrication)

   $(P > Q)$: /s/ $\xrightarrow{P}$ z $\xrightarrow{Q}$ $\widehat{\text{dz}}$ (transparent)   /t/ $\xrightarrow{P}$ t $\xrightarrow{Q}$ $\widehat{\text{ts}}$ (underapplication of P)

   $(Q > P)$: /s/ $\xrightarrow{Q}$ s $\xrightarrow{P}$ z (underapplication of Q)   /t/ $\xrightarrow{Q}$ $\widehat{\text{ts}}$ $\xrightarrow{P}$ $\widehat{\text{dz}}$ (transparent)

The diagram in Figure 29 summarizes this hypothetical case of hyper merger in our formal terms. Unlike merger, hyper merger is an interaction. However, unlike other interactions, hyper merger is either transparent or opaque depending both on the order of application of the two rules and on the input.
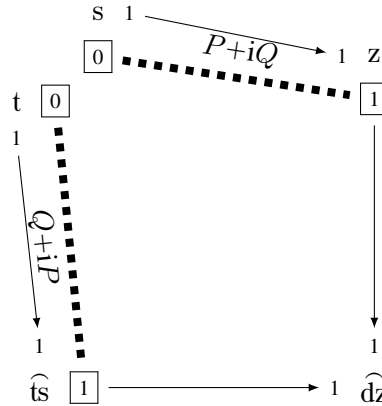
Figure 29: Hyper merger, $\{P{+}iQ, Q{+}iP\}$

Open jaws are added to corners and pies in the following updated table.

(42)

arrows	shape	interactions	Fig.	structure	atoms
2	corners	seeding	17	•→•→•	$\{P{+}iQ, Q{-}oP\}$
		mutual bleeding	20	•←•→•	$\{P{-}iQ, Q{-}iP\}$
		merger	21	•→•←•	$\{P{+}oQ, Q{+}oP\}$
3	pies (Πs)	feeding	22	→•→•←	$\{P{+}iQ, P{+}oQ\}$
		bleeding	23	←•→•→	$\{P{-}iQ, P{-}oQ\}$
4	open jaws	hyper seeding	25	←•←•→•→	$\{P{-}iQ, Q{+}oP\}$
		hyper m-bleeding	26	→•←•←•→	$\{P{-}oQ, Q{-}oP\}$
		hyper merger	27	→•→•←•←	$\{P{+}iQ, Q{+}iP\}$

5.4 Snails

The final distinct set of atom combinations results from making open jaws one step more complex, by adding one more arrow at either end. We will call these convoluted structures SNAILS. One such structure has already been witnessed, with the example of the transfusion interaction between stress and epenthesis in Palestinian Arabic in Figure 14. The other transfusion interaction discussed in §2.3, between long vowel lowering (P) and vowel copy epenthesis (Q) in Yokuts (8), is unsurprisingly also a snail. Note that the atom set in Figure 30, $\{P{+}oQ, P{-}oQ\}$, is identical to the set found for Palestinian Arabic in Figure 14 – and as in that case, the atom $P{-}oQ$ is diagnostic of misapplication in case Q applies first. This is what happens in the mapping *sduːk-* $\xrightarrow{Q}$ *suduːk-* $\xrightarrow{P}$ *sudoːk-* observed in Yokuts: the vowel copy epenthesis rule misapplies by apparently inserting the 'wrong' vowel *u* instead of *o*.
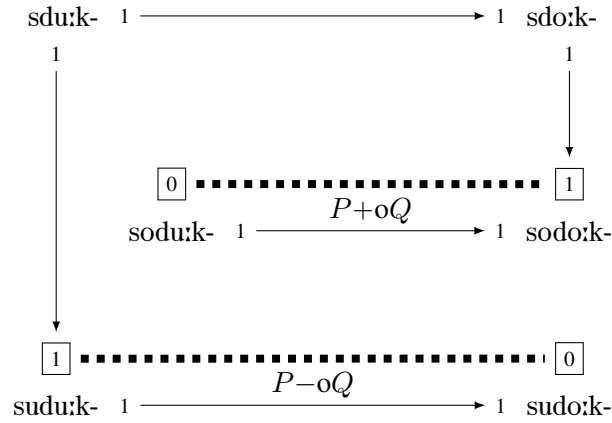
Figure 30: Transfusion, $\{P+\mathsf{o}Q,\ P-\mathsf{o}Q\}$

Another interaction type resulting in a distinct type of snail can be obtained by reversing the direction of vowel copy epenthesis (rule Q) in Yokuts: instead of inserting a vowel copy, this anti-Yokuts rule deletes the first of two copies of a vowel. The resulting rules and their application to a critical form are shown in (43).

(43) P: [+long] $\longrightarrow$ [−high]   (lowering)

       Q: αV $\longrightarrow$ $\varnothing$ / #C —— C αV   (deletion)

         $(P > Q)$: /soduːk-/ $\xrightarrow{P}$ sodoːk- $\xrightarrow{Q}$ sdoːk   (transparent)

         $(Q > P)$: /soduːk-/ $\xrightarrow{Q}$ soduːk- $\xrightarrow{P}$ sodoːk-   (Q underapplies)

We will refer to this distinct interaction type as INFUSION. The reversal of Q from the transfusion case results in a corresponding reversal in the orientation of the atoms, from the output orientation of the transfusion atoms $\{P+\mathsf{o}Q,\ P-\mathsf{o}Q\}$ to an input orientation of infusion atoms $\{P+\mathsf{i}Q,\ P-\mathsf{i}Q\}$. And because infusion contains the atom $P+\mathsf{i}Q$, it is expected to show underapplication when $Q > P$. Indeed, when the input is *soduːk-*, Q fails to apply, because Q only deletes a vowel if the following one is identical. Then P applies, creating *sodoːk-*, to which Q could apply but doesn't – a straightforward case of underapplication.

The structure for the anti-Yokuts infusion case in (43) is given in Figure 31.
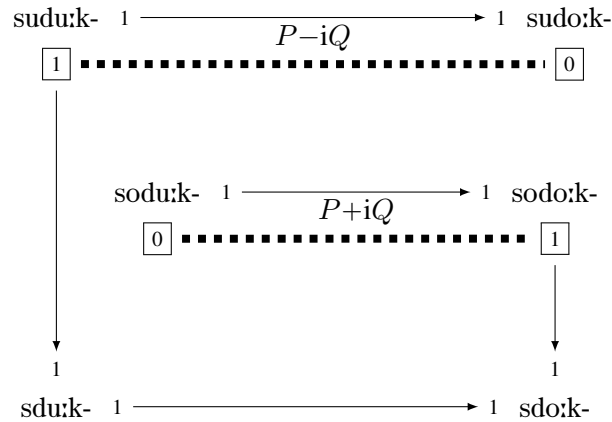
Figure 31: Infusion, $\{P-\mathsf{i}Q,\ P+\mathsf{i}Q\}$

The full table of possible two-atom interactions is in (44), including the expected opacity type depending on rule order: U for underapplication, M for misapplication (with overapplication as a special case).

(44)

arrows	shape	interactions	Fig.	structure	atoms	U/M
2	corners	seeding	17	•→•→•	$\{P{+}iQ, Q{-}oP\}$	U, M
		mutual bleeding	20	•←•→•	$\{P{-}iQ, Q{-}iP\}$	
		merger	21	•→•←•	$\{P{+}oQ, Q{+}oP\}$	
3	pies (Πs)	feeding	22	•→•→•←•	$\{P{+}iQ, P{+}oQ\}$	U
		bleeding	23	•←•→•→•	$\{P{-}iQ, P{-}oQ\}$	M
4	open jaws	hyper seeding	25	•→•←•←•→•	$\{P{-}iQ, Q{+}oP\}$	
		hyper m-bleeding	26	•←•←•→•→•	$\{P{-}oQ, Q{-}oP\}$	M
		hyper merger	27	•→•→•←•←•	$\{P{+}iQ, Q{+}iP\}$	U
5	snails	transfusion	30	•→•←•←•→•→•	$\{P{+}oQ, P{-}oQ\}$	M
		infusion	31	•→•→•←•←•→•	$\{P{+}iQ, P{-}iQ\}$	U

5.5 Beyond snails

The spiral can be extended even further, but with no new basic sets of atomic interactions. For example, adding another arrow to transfusion results in a more complex diagram that has the atoms $\{Q{+}iP, P{-}oQ\}$, the same combination as in seeding, as in Figure 32.
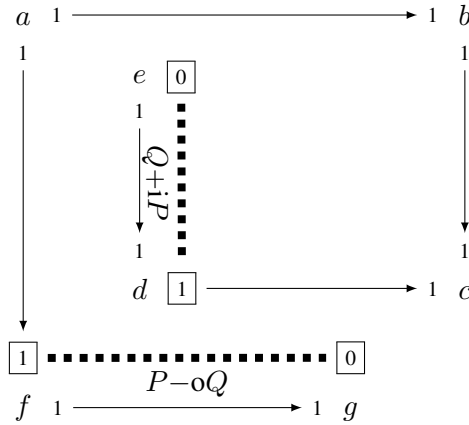

Figure 32: Hyper snail, $\{Q{+}iP, P{-}oQ\}$

We are not aware of real or even reasonable hypothetical examples that would instantiate such an interaction. However, the fact that the loop has closed, arriving back at the atom combinations contained by one of our two-arrow molecules (seeding), suggests that we have reached the end of the line as far as the typology of basic molecular interaction types is concerned.

5.6 Compound interactions

As demonstrated by the discussion in the preceding subsections, any pair of interacting maps minimally contains two atoms.[22] Of course, any given pair of maps can contain more than this minimum. For example, recall from fns. 9 and 13 the example of vowel deletion (P) and palatalization (Q) resulting in feeding or bleeding, depending on the input: *sai* $\xrightarrow{P}$ *si* $\xrightarrow{Q}$ *ʃi* vs. *sia* $\xrightarrow{P}$ *sa* $\xrightarrow{Q}$ *sa*. Such situations form two disjoint

---

[22] The general question of what atom combinations are possible in an interaction, and whether it is possible for an interaction to e.g. contain only one atom or more than two atoms, requires a mathematical investigation which is not undertaken in the present paper.

graphs, each with the appropriate topology ($\rightarrow\cdot\rightarrow\cdot\leftarrow$ for feeding, $\leftarrow\cdot\rightarrow\cdot\rightarrow$ for bleeding), and have been called COMPOUND interactions in previous work (Baković, Blumenfeld, Heinz & Riggle 2016; Baković & Blumenfeld 2016a,b); the specific case of concurrent feeding and bleeding is there called AMBIVALENCE.

In other situations, compound interactions can arise not in disjoint graphs, but in a single graph. For example, consider the compound interaction between apocope (P) and word-final nonapical deletion (Q) in (45), based on examples from Lardil (Hale 1973; Kavitskaya & Staroverov 2010).

(45) P: V $\longrightarrow \varnothing$ / $\sigma\sigma$ ___ #   (apocope)

       Q: [–apical] $\longrightarrow \varnothing$ / ___ #   (nonapical deletion)

       ✓$(P > Q)$: /dibirdibi/ $\xrightarrow{P}$ dibirdib $\xrightarrow{Q}$ dibirdi 'rock cod'   (P underapplies)

         $(Q > P)$: /dibirdibi/ $\xrightarrow{Q}$ dibirdibi $\xrightarrow{P}$ dibirdib   (Q underapplies)

Each of these rules underapplies under one of the orders because they are in what is called a FED COUNTERFEEDING interaction Kavitskaya & Staroverov (2010). Fed counterfeeding is a compound interaction, not a basic one, as shown in Figure 33 (recall Figure 24).
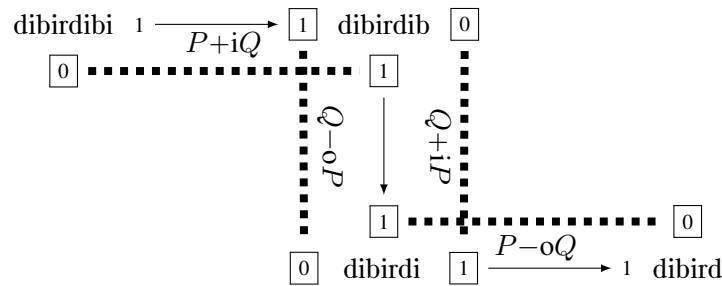

Figure 33: Fed counterfeeding, $\{P{+}\text{i}Q,\ Q{-}\text{o}P\} + \{Q{+}\text{i}P,\ P{-}\text{o}Q\}$

This graph illustrates two seeding interactions arising from mutual input-provision and mutual output-removal of these two rules. Because apocope applies only if its result is at least bisyllabic, and nonapical deletion only applies to nonapical consonants, there are some natural limits on the 'depth' of this fed counterfeeding interaction with most imaginable inputs – but, in principle, the mutual seeding can continue in stair-step fashion so long as consonants are nonapical and vowels are preceded by at least two syllables.

Viewed even more abstractly, the potentially unbounded stair-step graph in Figure 33 is something of an illusion. The set of strings represented by the 'original' input *dibirdibi* in the upper left includes the set represented by *dibirdi* in the lower middle, which means that the downward Q-arrow is effectively a loop back to the graph's origin. The loop-like quality of this type of interaction is more transparently obvious in the case of DUKE-OF-YORK DERIVATIONS (Pullum 1976; McCarthy 1999, 2003), not coincidentally called FED COUNTERFEEDING ON FOCUS interactions by Baković (2011:46). A case of this type is illustrated in (46), based on examples from Nuu-chah-nulth (formerly known as Nootka; Sapir & Swadesh 1978). This type of case can be represented either as an unbounded stair-step graph like Figure 33, or more simply and directly as an infinite loop (or with a bidirectional arrow: $m'oq \underset{Q}{\overset{P}{\longleftrightarrow}} m'oq^w$).

(46) P: [+dors] $\longrightarrow$ [+round] / [+round] ___   (labialization)

       Q: [+dors] $\longrightarrow$ [–round] / ___ $]_\sigma$ (delabialization)

       ✓$(P > Q)$: /m'oːq$^{(w)}$/ $\xrightarrow{P}$ m'oːq^w $\xrightarrow{Q}$ m'oːq 'throwing off sparks'   (P underapplies)

         $(Q > P)$: /m'oːq$^{(w)}$/ $\xrightarrow{Q}$ m'oːq $\xrightarrow{P}$ m'oːq^w (Q underapplies)

The same considerations apply to the 'interaction' between individual applications of e.g. an iterative feature spreading rule: each application of the rule creates a new instance of the same rule's structural description. The mutual seeding stair-step graph in this case would be bounded only by the length of a given string; again, viewed more abstractly, it would be a loop on a single node.

Beyond these types of examples, the compound interaction landscape is likely vast and remains to be fully explored and properly charted.

6   Relations between interactions

Another view of the typological space can be gained by relating the various molecular interaction types in the typology to each other via two formal 'conversion' operations (Baković & Blumenfeld 2019).

To start, feeding and bleeding can be related by an operation called FLIPPING (Hein, Murphy & Zaleska 2014). Suppose rule P feeds rule Q, and P is of the form $a \longrightarrow b \;/\; c \underline{\quad} d$. The flipping operation converts P to rule $\mathbf{f}(P)$, $b \longrightarrow a \;/\; c \underline{\quad} d$. Then, $\mathbf{f}(P)$ bleeds Q. For example, suppose P is word-final raising, which feeds assibilation (Q), based on examples from Finnish (Kiparsky 1973a, 1993).[23] The flipped version of P, $\mathbf{f}(P)$, is word-final lowering, which instead bleeds assibilation.

(47) P: e $\longrightarrow$ i $/$ ___ #   (raising)

       $\mathbf{f}(P)$: i $\longrightarrow$ e $/$ ___ #   (lowering)

       Q: t $\longrightarrow$ s $/$ ___ i   (assibilation)

       ✓ $(P > Q)$: /vete/ $\xrightarrow{P}$ veti $\xrightarrow{Q}$ vesi 'water'   (P feeds Q)

       $(\mathbf{f}(P) > Q)$: /veti/ $\xrightarrow{\mathbf{f}(P)}$ vete $\xrightarrow{Q}$ vete   ($\mathbf{f}(P)$ bleeds Q)

The flipping operation on rules, as loosely defined above in terms of rule notation, is only coherent under certain circumstances. For example, the debuccalization rule C $\longrightarrow$ ʔ $/$ ___ # does not have a flipped counterpart, because the notation ʔ $\longrightarrow$ C is incoherent. More generally, such a flipping operation can only be defined for maps that do not contain nonvacuous neutralizing mappings. A subtler point is that flipping a rule as defined in terms of notation is not the same thing as reversing a map. The Finnish raising map, for example, cannot be reversed, because it is not an injective one: a string like *veti* is the image of both *vete* and *veti*. The flipped Finnish rule *i* $\longrightarrow$ *e* $/$ ___ # simply makes a choice and maps *veti* to *vete*.

Flipping is thus defined in (48). If the original map contains any nonvacuous neutralizations, flipping is undefined. Otherwise, flipping reverses all nonvacuous arrows.

(48)   Let P be a map. The FLIPPED map, $\mathbf{f}(P)$:

     a.   is undefined if there exist pairwise distinct a, b, c such that $P(a) = c$ and $P(b) = c$;

     b.   otherwise, for every nonvacuous mapping $P(a) = b$, $\mathbf{f}(P)(b) = a$.

With this definition in hand, it is clear how feeding and bleeding are related via flipping. The schematic minimal configuration for feeding is $\cdot \xrightarrow{P} \cdot \xrightarrow{Q} \cdot \xleftarrow{P} \cdot$. Flipping the P arrows results in $\cdot \xleftarrow{P} \cdot \xrightarrow{Q} \cdot \xrightarrow{P} \cdot$, which is precisely the minimal configuration for bleeding. Likewise for the two-arrow corners: starting with merger, $\cdot \xrightarrow{P} \cdot \xleftarrow{Q} \cdot$, flipping Q would give us seeding, $\cdot \xrightarrow{P} \cdot \xrightarrow{Q} \cdot$, and then flipping P would give us mutual bleeding, $\cdot \xleftarrow{P} \cdot \xrightarrow{Q} \cdot$. Similarly, transfusion ($\cdot \xrightarrow{P} \cdot \xleftarrow{Q} \cdot \xleftarrow{P} \cdot \xrightarrow{Q} \cdot \xrightarrow{P} \cdot$) and infusion ($\cdot \xrightarrow{P} \cdot \xrightarrow{Q} \cdot \xleftarrow{P} \cdot \xleftarrow{Q} \cdot \xrightarrow{P} \cdot$) are related by flipping Q. In fact, the anti-Yokuts infusion case in (43) and Figure 31 was constructed precisely in this way, by flipping one of the rules of the Yokuts transfusion case in (8) and Figure 30.

---

[23] In the actual Finnish example there is also nonderived environment blocking of assibilation; recall fn. 17.

Another operation relating interaction types is CROPPING (Baković & Blumenfeld 2019). For example, suppose the Finnish raising rule is narrowed in scope to apply only after nonsibilants: $e \longrightarrow i$ / [–strid] —— #. This cropped raising rule, or $\mathbf{c}$(raising), now seeds rather than feeds assibilation, as instantiated by the sequence $te \longrightarrow ti \longrightarrow si$. The first step, $\mathbf{c}$(raising), creates the conditions for assibilation, but the application of assibilation then destroys the condition for $\mathbf{c}$(raising), which does not apply after sibilants. The effect of cropping is to narrow the extension of the raising rule by excising one of the mappings, namely $se \longrightarrow si$, thus converting the feeding topology $\cdot \xrightarrow{P} \cdot \xrightarrow{Q} \cdot \xleftarrow{P} \cdot$ to the seeding topology $\cdot \xrightarrow{P} \cdot \xrightarrow{Q} \cdot$, as illustrated more specifically using the Finnish-like example below.
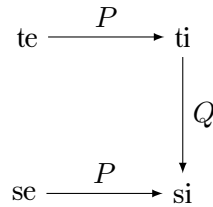
Figure 34: P (raising) feeds Q (assibilation)
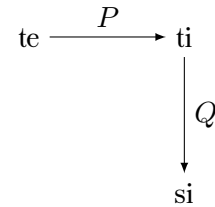
Figure 35: P ($\mathbf{c}$(raising)) seeds Q (assibilation)

Cropping, understood symmetrically to include both removal and addition of arrows to structures, relates n-arrow interactions (e.g. the 2-arrow corners, seeding and mutual bleeding) with $(n{+}1)$-arrow interactions (e.g. the 3-arrow pies, feeding and bleeding). In fact, our typological survey in §5 began with the minimal 2-arrow corners and proceeded by progressively adding arrows to the structures, making it clear how in the sequence *corner—pie—open jaw—snail*, successive steps are related by cropping.

But cropping is a relatively loose operation: in any given case, there are many possible crops that will adequately relate one interaction with another. For example, in the Finnish case, we could have cropped raising such that it applies only word-finally after stops, or only word-finally after noncontinuants, or in any word-final environment except after fricatives, etc., and each of these distinct croppings would have converted feeding to seeding in some mappings. Thus, the meaning of the claim "feeding can be converted to seeding by cropping" is something like "if P feeds Q, then there exists a way to remove some mappings from P to produce a new cropped map, $\mathbf{c}(P)$, such that $\mathbf{c}(P)$ seeds Q."

Making seemingly minimal changes to a given rule can also result in what appears to be an extended form of cropping, whereby interactions differing by more than one arrow can be directly related to each other. For example, consider the Palestinian Arabic 5-arrow transfusion case in (7) and Figure 14 and its conversion into the hypothetical 2-arrow seeding case in (32) and Figure 15. This conversion was accomplished simply by limiting epenthesis to apply after stressed vowels, eliminating three arrows in one fell swoop. Clearly, more work needs to be done to properly define cropping such that it only relates arrow-wise 'adjacent' interactions – or at least, to define the conditions under which adjacent vs. nonadjacent interactions can be related.

The relations observed in previous work now simply follow from our formal machinery: flipping and cropping have a clear effect on the arrow topology for all of the interactions in the typology, and they produce exactly the expected effects. Figure 36 summarizes the operations that relate various interaction types.
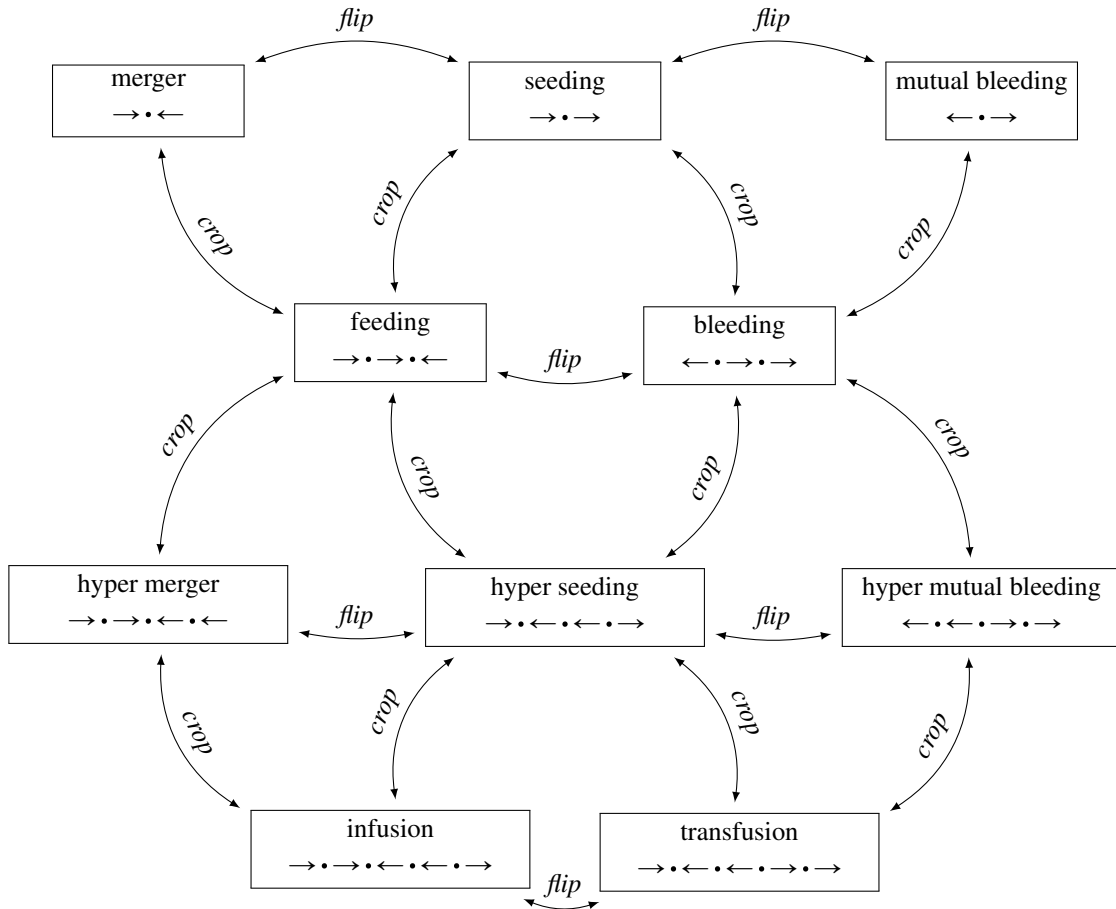
Figure 36: Conversion operations and interaction types

Recall from §5.1 that merger is not strictly speaking an interaction, given that $Q(P(x)) = P(Q(x))$. Nevertheless, merger and its relations to seeding (via flipping) and feeding (via cropping) are included in Figure 36 for completeness. Interactions can thus be related to noninteracting structures. Another such relation can be established by adding an arrow to a pie structure (via cropping) to close the square. Consider the bleeding pie structure in the Polish example, repeated below in Figure 37. Here, word-final devoicing (P) interacts with mid vowel raising before word-final voiced nonnasals (Q). To close the square, raising can be extended to apply before all word-final nonnasals regardless of voicing. There is no interaction between this extended raising and word-final devoicing, and there are also no provision or removal atoms in this case.
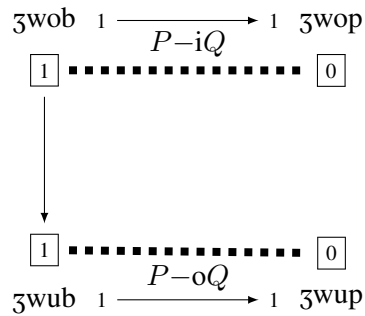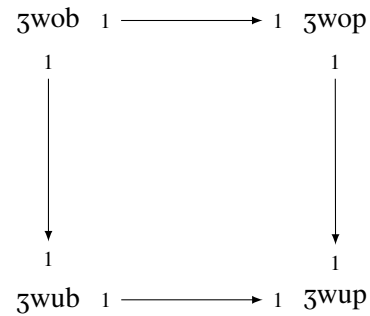

Figure 37: Bleeding, $\{P+\mathrm{i}Q,\ P+\mathrm{o}Q\}$

Figure 38: No interaction

7 Conclusion

One significant result of our work on the algebra and resulting typology of process interactions is that the opacity question (2) posed in §1 can now be answered with some precision. Two factors make a derivation involving two processes P and Q opaque: their order of application, and the presence of one or both of the input-provision (+i) and output-removal (−o) atoms. Furthermore, these two atoms produce distinct signatures in the opaque interactions. Input-provision is responsible for derivations with the character of underapplication, where it looks like a rule should have applied, but it did not. Output-removal is responsible for derivations with the character of misapplication, where a process applies in some unexpected way relative to an otherwise expected alternative. This alternative can be either nonapplication (in the case of overapplication) or application in a different locus or fashion (in the complementary case of misapplication).[24] These properties are evident in all of the examples examined in this paper, or are predicted for those interaction types that we have thus far been unable to identify in attested language patterns.

The more general interaction question (1) also receives a precise answer. An interaction arises whenever a process P is able to change the properties of a string with respect to another process Q. The extensive prior literature on this topic has focused on input properties with respect to Q – whether the input or output of P is also a possible input of Q. We have demonstrated that to understand the space of possible interactions it is also necessary to consider output properties with respect to Q. This move permits us to map out a typology of nine basic interactions (not including merger), of which only five (feeding, bleeding, seeding, transfusion, and mutual bleeding) have previously been described as distinct phenomena, albeit without the full understanding of what makes them distinct and what generates the opaque derivations observed in some of them.

The formal focus of this paper is what has made these results possible. Stepping back from particular phenomena in particular languages and treating maps as abstract algebraic operations allows us to uncover the formal properties of their interactions and leads us back to the concrete typology and its empirical predictions.

References

Akinlabi, Akinbiyi. 1996. Featural affixation. *Journal of Linguistics* 32. 239–389.

Al-Mozainy, H. Q. 1981. *Vowel alternations in a Bedouin Hijazi Arabic dialect: Abstractness and stress*. Austin, TX: University of Texas dissertation.

Alderete, John. 1999. Head dependence in stress-epenthesis interactions. In Ben Hermans & Marc van Oostendorp (eds.), *The derivational residue in Optimality Theory*, 29–50. Amsterdam: Benjamins.

Andersson, Samuel. 2020. I'm sorry but time travel isn't real: Against counterfeeding from the past. *University of Pennsylvania Working Papers in Linguistics* 26(1). 2. https://repository.upenn.edu/pwpl/vol26/iss1/2.

Archangeli, Diana & Douglas Pulleyblank. 1994. *Grounded phonology*. Cambridge, MA: MIT Press.

Baković, Eric. 2005. Antigemination, assimilation and the determination of identity. *Phonology* 22. 279–315. https://doi.org/10.1017/S0952675705000631.

Baković, Eric. 2007. A revised typology of opaque generalisations. *Phonology* 24. 217–259. https://doi.org/10.1017/S0952675707001194.

Baković, Eric. 2011. Opacity and ordering. In John A. Goldsmith, Jason Riggle & Alan C. L. Yu (eds.), *The handbook of phonological theory*, 40–67. Malden, MA: Wiley Blackwell. https://doi.org/doi:10.1002/9781444343069.ch2.

Baković, Eric. 2013. *Blocking and complementarity in phonological theory*. London: Equinox.

Baković, Eric & Lev Blumenfeld. 2016a. Compound relations between phonological maps. Invited talk, 9th North American Phonology Conference, Concordia University.

---

[24]With the caveat that −o atoms in on-focus interactions may produce unintuitive results; recall §4.4.

Baković, Eric & Lev Blumenfeld. 2016b. Compound relations between phonological maps: Ambivalence and reciprocity. Unpublished manuscript, UC San Diego and Carleton University.

Baković, Eric & Lev Blumenfeld. 2018. Overapplication conversion. In Ryan Bennett, Andrew Angeles, Adrian Brasoveanu, Dhyana Buckley, Nick Kalivoda, Shigeto Kawahara, Grant McGuire & Jaye Padgett (eds.), *Hana-bana: A Festschrift for Junko Itô and Armin Mester*, Santa Cruz, CA: Department of Linguistics, University of California, Santa Cruz. https://itomestercelebration.sites.ucsc.edu/.

Baković, Eric & Lev Blumenfeld. 2019. Rule interaction conversion operations. *Loquens* 6(2). e062. https://doi.org/10.3989/loquens.2019.062.

Baković, Eric, Lev Blumenfeld, Jeff Heinz & Jason Riggle. 2016. Decomposing complex relations between phonological maps. Talk presented at the 90th Annual Meeting of the LSA, Washington, DC.

Bethin, Christina. 1978. *Phonological rules in the nominative singular and genitive plural of the Slavic substantive declension*. Champaign-Urbana, IL: University of Illinois dissertation.

Bright, William. 1957. *The Karok Language* (University of California Publications in Linguistics). Berkeley: University of California Press.

Burzio, Luigi. 2000. Cycles, non-derived-environment blocking and correspondence. In Joost Dekkers, Frank van der Leeuw & Jeroen van de Weijer (eds.), *Optimality Theory: Phonology, syntax and acquisition*, 47–87. Oxford: Oxford University Press.

Burzio, Luigi. 2011. Derived environment effects. In Marc van Oostendorp, Colin Ewen, Beth Hume & Keren Rice (eds.), *The Blackwell companion to phonology*, Oxford: Wiley-Blackwell. https://doi.org/10.1002/9781444335262.wbctp0088.

Bye, Patrik. 2002. *Virtual phonology: Rule sandwiching and multiple opacity in North Saami*. Tromsø: University of Tromsø dissertation.

Bye, Patrik. 2011. Derivations. In Nancy C. Kula, Bert Botma & Kuniya Nasukawa (eds.), *Continuum companion to phonology*, 135–173. London: Continuum.

Chandlee, Jane. 2021. Nonderived environment blocking and input-oriented computation. *Evolutionary Linguistic Theory* 3. 129–153. https://doi.org/10.1075/elt.00031.cha.

Chomsky, Noam. 1967. Some general properties of phonological rules. *Language* 43. 102–128. https://doi.org/10.2307/411387.

Chomsky, Noam & Morris Halle. 1968. *The sound pattern of English*. New York: Harper & Row.

Hale, Kenneth. 1973. Deep-surface canonical disparities in relation to analysis and change: An Australian example. In Thomas Sebeok (ed.), *Current trends in linguistics*, 401–458. The Hague: Mouton. https://doi.org/10.1515/9783111418797-018.

Halle, Morris. 1962. Phonology in generative grammar. *Word* 18(1–3). 54–72. https://doi.org/10.1080/00437956.1962.11659765.

Hayes, Bruce. 1995. *Metrical stress theory: Principles and case studies*. Chicago: University of Chicago Press.

Hayes, Bruce & James White. 2015. Saltation and the P-map. *Phonology* 32(2). 267–302. https://doi.org/10.1017/S0952675715000159.

Hein, Johannes, Andrew Murphy & Joanna Zaleska. 2014. Rule flipping and the feeding–bleeding relationship. In Anke Assmann, Sebastian Bank, Doreen Georgi, Timo Klein, Philipp Weisser & Eva Zimmermann (eds.), *Topics at InfL*, vol. 92 Linguistische Arbeitsberichte, 1–32. Leipzig: Universität Leipzig.

Iverson, Gregory K. 1995. Rule ordering. In John A. Goldsmith (ed.), *The handbook of phonological theory*, 609–614. Oxford: Blackwell.

Joshi, S. D. & Paul Kiparsky. 1979. Siddha and asiddha in Paninian phonology. In Daniel A. Dinnsen (ed.), *Current approaches to phonological theory*, 223–250. Bloomington, IN: Indiana University Press.

Joshi, S. D. & Paul Kiparsky. 2005. The extended *siddha*-principle. *Annals of the Bhandarkar Oriental Research Institute* 86. 1–26. https://www.jstor.org/stable/41692378.

Kavitskaya, Darya & Peter Staroverov. 2010. When an interaction is both opaque and transparent: the paradox of fed counterfeeding. *Phonology* 27. 255–288. https://doi.org/10.1017/S0952675710000126.

Kennedy, Christopher. 1994. Morphological alignment and head projection: Toward a nonderivational account of epenthesis and exceptional stress in Dakota. *Phonology at Santa Cruz* 3. 47–64.

Kenstowicz, Michael J. & Charles W. Kisseberth. 1979. *Generative phonology: Description and theory*. San Diego, CA: Academic Press.

Kiparsky, Paul. 1968. Linguistic universals and linguistic change. In Emmon Bach & Robert T. Harms (eds.), *Universals in linguistic theory*, 170–202. New York: Holt, Reinhart and Winston. https://doi.org/10.1515/9783111666242.13.

Kiparsky, Paul. 1971. Historical linguistics. In William O. Dingwall (ed.), *A survey of linguistic science*, 576–642. College Park: University of Maryland Linguistics Program.

Kiparsky, Paul. 1973a. How abstract is phonology? In Osamu Fujimura (ed.), *Three dimensions of linguistic theory*, 5–56. Tokyo: TEC.

Kiparsky, Paul. 1973b. Abstractness, opacity and global rules. In Osamu Fujimura (ed.), *Three dimensions of linguistic theory*, 57–86. Tokyo: TEC.

Kiparsky, Paul. 1973c. "Elsewhere" in phonology. In Stephen R. Anderson & Paul Kiparsky (eds.), *A Festschrift for Morris Halle*, 93–106. New York: Holt, Reinhart and Winston.

Kiparsky, Paul. 1993. Blocking in nonderived environments. In Sharon Hargus & Ellen M. Kaisse (eds.), *Studies in Lexical Phonology*, vol. 4 Phonetics and Phonology, 277–313. San Diego, CA: Academic Press. https://doi.org/10.1016/B978-0-12-325071-1.50016-9.

Kiparsky, Paul. 2000. Opacity and cyclicity. *The Linguistic Review* 17. 351–365. https://doi.org/10.1515/tlir.2000.17.2-4.351.

Kiparsky, Paul. 2015. Stratal OT: A synopsis and FAQs. In Yuchau E. Hsiao & Lian-Hee Wee (eds.), *Capturing phonological shades within and across languages*, 2–44. Newcastle upon Tyne: Cambridge Scholars Library.

Kula, Nancy C. 2008. Derived environment effects: A representational approach. *Lingua* 118. 1328–1343. https://doi.org/10.1016/j.lingua.2007.09.011.

Kuroda, S.-Y. 1967. *Yawelmani phonology*. Cambridge, MA: MIT Press.

Lee, Minkyung. 1999. A case of sympathy in Javanese affixation. In Karen Baertsch & Daniel A. Dinnsen (eds.), *Optimal green ideas in phonology*, 31–36. Bloomington, IN: IULC Publications.

Lee, Minkyung. 2007. OT-CC and feeding opacity in Javanese. *Studies in Phonetics, Phonology and Morphology* 13(2). 333–350.

Łubowicz, Anna. 2002. Derived environment effects in Optimality Theory. *Lingua* 112. 243–280. https://doi.org/10.1016/S0024-3841(01)00043-2.

Mascaró, Joan. 2011. Rule ordering. In Marc van Oostendorp, Colin Ewen, Beth Hume & Keren Rice (eds.), *The Blackwell companion to phonology*, 1736–1760. Oxford: Wiley-Blackwell. https://doi.org/10.1002/9781444335262.wbctp0074.

McCarthy, John J. 1999. Sympathy and phonological opacity. *Phonology* 16(3). 331–399. https://doi.org/10.1017/S0952675799003784.

McCarthy, John J. 2003. Sympathy, cumulativity and the Duke-of-York gambit. In Caroline Féry & Ruben van de Vijver (eds.), *The syllable in Optimality Theory*, 23–76. Cambridge: Cambridge University Press.

McCarthy, John J. 2007a. Derivations and levels of representation. In Paul de Lacy (ed.), *The Cambridge handbook of phonology*, 99–117. Cambridge: Cambridge University Press.

McCarthy, John J. 2007b. *Hidden generalizations: Phonological opacity in Optimality Theory*. London: Equinox.

Meinhardt, Eric M., Anna Mai, Eric Baković & Adam G McCollum. 2024. Weak determinism and the computational consequences of interaction. *Natural Language & Linguistic Theory*. https://doi.org/10.1007/s11049-023-09578-1.

Moreton, Elliott & Paul Smolensky. 2002. Typological consequences of local constraint conjunction. *Proceedings of the West Coast Conference on Formal Linguistics* 21. 306–319.

Norton, Russell James. 2003. *Derivational phonology and optimality phonology: Formal comparison and synthesis*. Colchester, UK: University of Essex dissertation. ROA-613, Rutgers Optimality Archive, http://roa.rutgers.edu/.

Potts, Christopher & Geoffrey K. Pullum. 2002. Model theory and the content of OT constraints. *Phonology* 19. 361–393. https://doi.org/10.1017/S0952675703004408.

Prillop, Külli. 2018. Interacting processes in phonological theory. *Phonology* 35(3). 441–479. https://doi.org/10.1017/S095267571800012X.

Pruitt, Kathryn. 2023. Serialism and opacity in phonological theory. *Annual Review of Linguistics* 9. 497–517.

Pullum, Geoffrey K. 1976. The Duke of York gambit. *Journal of Linguistics* 12(1). 83–102. https://www.jstor.org/stable/4175335.

Rasin, Ezer. 2016. A rule-ordering theory of blocking in nonderived environments. In Gunnar Ólafur Hansson, Ashley Farris-Trimble, Kevin McMullin & Douglas Pulleybank (eds.), *Supplemental proceedings of the Annual Meeting on Phonology 2015*, Washington, DC: Linguistic Society of America. https://doi.org/10.3765/amp.v3i0.3689.

Rasin, Ezer. 2022. Shifting interactions and countershifting opacity: A note on opacity in Harmonic Serialism. *Linguistic Inquiry*. 53(4). 836–851. https://doi.org/10.1162/ling_a_00430.

Rubach, Jerzy. 1984. *Cyclic and lexical phonology: The structure of Polish*. Dordrecht: Foris.

Sapir, Edward & Morris Swadesh. 1978. *Nootka texts: Tales and ethnological narratives, with grammatical notes and lexical material*. New York: AMS Press.

Sayeed, Ollie. 2016. Phonological time travel: 'Counterfeeding from the future'. Talk presented at the Annual Meeting of the Linguistics Association of Great Britain, University of York.

Shaw, Patricia. 1980. *Theoretical issues in Dakota phonology and morphology*. London: Routledge.

Shaw, Patricia. 1985. Modularisation and substantive constraints in Dakota lexical phonology. *Phonology Yearbook* 2. 173–202. https://www.jstor.org/stable/4419956.

Smith, Jennifer L. 2022. Some formal implication of deletion saltation. *Linguistic Inquiry*. 53(4). 852–864. https://doi.org/10.1162/ling_a_00426.

Sprouse, Ronald. 1997. A case for enriched inputs. ROA-193, Rutgers Optimality Archive, http://roa.rutgers.edu.

Steriade, Donca. 1986. Yokuts and the vowel plane. *Linguistic Inquiry* 17(1). 129–146. http://www.jstor.org/stable/4178478.

Tesar, Bruce. 2014. *Output-driven phonology: Theory and learning*. Cambridge: Cambridge University Press.

Tucker, Archibald Norman & John Tompo Ole Mpaayei. 1955. *A Maasai grammar, with vocabulary*. London: Longmans, Green & Co.

Watson, Janet C. E. 2011. Word stress in Arabic. In Marc van Oostendorp, Colin J. Ewen, Elizabeth Hume & Keren Rice (eds.), *The Blackwell companion to phonology*, chap. 135, 2990–3018. London: Wiley-Blackwell. https://doi.org/10.1002/9781444335262.wbctp0124.

White, James. 2013. *Bias in phonological learning: Evidence from saltation*. Los Angeles: UCLA dissertation.

White, James. 2014. Evidence for a learning bias against saltatory phonological alternations. *Cognition* 130(1). 96–115. https://doi.org/10.1016/j.cognition.2013.09.008.

White, James. 2017. Accounting for the learnability of saltation in phonological theory: A maximum entropy model with a P-map bias. *Language* 93. 1–36. https://doi.org/10.1353/lan.2017.0001.

Wilson, Colin. 2006. Counterfeeding from the past. Unpublished manuscript, UCLA.

Wolf, Matthew. 2011. Limits on global rules in Optimality Theory with candidate chains. *Phonology* 28. 87–128. https://www.jstor.org/stable/41303155.

Zwicky, Arnold. 1987. Rule interactions: Another gloss on K&K. In *Innovations in Linguistics Education*, vol. 5, 91–111.

Eric Baković

Linguistics Department

University of California, San Diego

9500 Gilman Dr., #0108

La Jolla, CA 92093-0108 USA

ebakovic@ucsd.edu

Lev Blumenfeld

School of Linguistics and Language Studies

Carleton University

1125 Colonel By Dr.

Ottawa ON K1S 5B6 Canada

lev.blumenfeld@carleton.ca