A formal typology of process interactions

Eric Baković, UC San Diego
Lev Blumenfeld, Carleton University

October 6, 2022

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. We adopt this general formalization of processes and interactions in this article for the purposes of answering 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 of interaction behind the terms FEEDING, BLEEDING, COUNTERFEEDING, and COUNTERBLEEDING, and relations among them, are familiar to phonologists from a large body of work.[1]

Some of this work, however — at different times and in different ways — has exposed gaps in the classical typology of these pairwise rule orders. Most significantly, additional pairwise rule interaction types have been identified — MUTUAL BLEEDING (Kiparsky 1971), DUKE-OF-YORK DERIVATIONS (Pullum 1976, McCarthy 2003, Norton 2003), TRANSFUSION or SHIFTING (Zwicky 1987, Kiparsky 2015, Rasin to appear), SELF-DESTRUCTIVE FEEDING and NON-GRATUITOUS FEEDING (Baković 2007), 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).

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. This question concerns the notion of OPACITY, which has received many treatments and definitions in previous work (see the handbook chapters cited in fn. 1). Opaque orders result in outputs where some generalization stated by one of the rules is in some way obscure. While intuitive, this notion has proven difficult to understand formally, and this is where our algebraic approach most obviously proves its usefulness.

(2) **The opacity question**
What makes an interaction between two processes opaque (conversely, transparent)?

Again, 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

[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.

(2015), Prillop (2018), and Rasin (to appear). We show here that the same algebraic formalization of process interactions that answers the interaction question in (1) leads to a satisfying answer to the opacity question in (2).

Nothing in our formal machinery relies on a process 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 $a \longrightarrow b \ / \ c \_\_ d$, 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. Although most of our examples are stated in this formalism, we are not beholden to it, such as in the statement of the (disjunctive) stress rule in Palestinian Arabic discussed in §2.2.

Phonological forms are assumed to be strings of segments defined in terms of binary features, with appropriately-placed boundaries. The rule $a \longrightarrow b \ / \ c \_\_ d$ APPLIES to all strings containing *cad* as a substring. Note that we use the term APPLIES strictly, to mean 'applies *non-vacuously*' — that is, if a rule "applies" to an input, it produces an output that is distinct from that input.

The scope of our approach has some limitations. First, our focus on *pairwise* interactions is not intended to ignore interactions among three or more processes (e.g. 'rule sandwiching'; Bye 2002). Given serial ordering as our model, any such case can be recursively understood as 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 will often affect strings in complex and/or distributed ways that isolated processes as we've circumscribed them above typically cannot, but we assert that the formal landscape of interaction is the same. Some interactions not statable with conjunctively ordered rules *tout court* are outside of our scope: disjunctive rule application (Chomsky & Halle 1968, Kiparsky 1973c, Baković 2013), non-derived environment blocking (Kiparsky 1968, 1993, Łubowicz 2002, Burzio 2000, 2011, Kula 2008, Rasin 2016, Chandlee 2021), cross-derivational feeding (Baković 2005, 2007), saltation (White 2013, 2014, 2017, Hayes & White 2015, Smith to appear), and so on.

With these preliminaries as background, the article unfolds as follows. In §2, we elaborate on the classical view of process 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 formal typology of process interactions in §5, answering the interaction question. Along the way, in §4, we'll see that this approach provides the key to answering the opacity question. In §6 we discuss two ways to characterize the typology, one in terms of a general principle that governs all pairwise interactions and another in terms of a set of formal relationships among the various interaction types in the typology. We conclude the article in §7.

2 Interactions

2.1 The classical view

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

Suppose there are two rules P, Q. Minimally, an INTERACTION between two rules for some input x takes place if the order of application of the two rules matters; 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 $Q(P(x)) = x \neq Q(x)$. Such a situation is what is traditionally called BLEEDING.

Feeding and bleeding as defined above 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, self-citation omitted for review).

Additionally, while feeding and bleeding can be understood as abstract properties that do not depend on the actual ordering of the two rules in a given phonological grammar two orders are available in any given grammar, $P \succ Q$ and $Q \succ P$, and this results in another 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 is ordered before Q. The reverse order $Q \succ P$ results in derivations (or relations) called COUNTERFEEDING and COUNTERBLEEDING.

This typology of interaction is often discussed in the context of OPACITY and its complement, TRANSPARENCY. Opacity is a property of the output of a serial derivation relative to the rules that have applied to it. Kiparsky (1973b: 79) offers the following characterization of two types of opacity: opaque derivations are ones ending in outputs to which some rule that was expected to have applied has not in fact applied (3a), or ones ending in outputs to which some rule has applied unexpectedly (3b).

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

a. instances of a in the environment c __ d, or

b. instances of b derived by R that occur in environments other than c __ d.

The first part in (3a) refers to surface inputs to rule R, and is now commonly referred to as UNDERAPPLICATION (McCarthy 1999). To establish underapplication, all that is needed is an inventory of rules and surface structures. The second part in (3b) is more complex: OVERAPPLICATION opacity occurs 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.

The classical view attributes transparency to feeding and bleeding and opacity to their counter-orders, underapplication being a symptom of counterfeeding and overapplication a symptom of counterbleeding. The full classical typology of specific situations is given in (4).

(4)

	$P \succ Q$	$Q \succ P$
P causes application of Q	feeding	counterfeeding
P prevents application of Q	bleeding	counterbleeding
	transparent	opaque

Our aim in this article 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 non-final open syllables (Q), shown in (5). In this case, vocalization potentially feeds raising. If vocalization applies before raising, then feeding takes place, otherwise counterfeeding, as in actual Bedouin Arabic. The form [badu] 'Bedouin' shows underapplication (3a): raising should have applied to it 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 is indicated by a checkmark (✓) next to the rule order. The choice of labels P and Q is arbitrary; the language's actual rule order may be $P \succ Q$ in some examples and $Q \succ P$ in others.

(5) P: [−cons] $\longrightarrow$ [+syll] / C __ # (vocalization)
Q: a $\longrightarrow$ i / __ CV (raising)

($P \succ Q$): /badw/ $\xrightarrow{P}$ badu $\xrightarrow{Q}$ bidu (feeding, transparent)
✓($Q \succ P$): /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 voiced non-nasals (Q), shown in (6). In this case, devoicing potentially bleeds raising. If devoicing applies before raising, then bleeding takes place, otherwise counterbleeding, as in Polish. The form [ʒwup] 'crib' shows overapplication: raising should not have applied to it but did.

(6) P: [−son] $\longrightarrow$ [−voi] / __ # (devoicing)
Q: o $\longrightarrow$ u / __ $\begin{bmatrix}\text{+voi}\\ \text{−nas}\end{bmatrix}$ (raising)

($P \succ Q$): /ʒwob/ $\xrightarrow{P}$ ʒwop $\xrightarrow{Q}$ ʒwop (bleeding, transparent)
✓($Q \succ P$): /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 a disjunction of underapplication and overapplication. 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.

2.2 Misapplication and the Simultaneous Application Condition

The first step beyond the classical feeding/bleeding dyad 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 often happens with stress rules or, more generally, in cases where a rule's locus of application can be changed by the application of another. In that case, opacity is neither underapplication nor overapplication, but might instead be called MISAPPLICATION, or application in an otherwise unmotivated locus.

Consider the case in (7), based on examples from Palestinian Arabic (Hayes 1995: 125, Kiparsky 2000, Watson 2011). An epenthesis rule breaks up final CC clusters (P), and the relevant part of the stress rule places stress on final syllables when superheavy, else on heavy penults, else on antepenults (Q). So, for input /katabt/ 'I wrote', both rules are applicable. The actual output is [katábit], with penultimate stress, because the stress rule applies prior to epenthesis. The reverse order would have given *[kátabit].

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

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

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

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

Intuitively, the correct outcome in (7) seems opaque: the location of stress in [katábit] is inconsistent with the stress rule. But, at the same time, it is difficult to interpret these derivations in terms of underapplication or overapplication, or even in terms of feeding and bleeding. The apparent opacity is in the unexpected penultimate location of stress in the output [katábit], but can we say that antepenultimate stress has underapplied despite its conditions being met on the surface? or that penultimate stress has overapplied despite its conditions *not* being met on the surface? The answer is neither, showing that (3) is incomplete as a characterization of opacity because it does not cover misapplication.

A more general characterization of 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, 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.

(8) **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 definition, a rule (= input-output map) Q is opaque with respect to a rule P if (a) for at least one input x, the output of x depends on the order of application of 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 — $P(Q(x)) = P, Q(x)$.

What unifies various forms of opacity is that the conditions of an earlier rule Q are obscured by the effects of a later rule P either because P newly provides Q's conditions (= underapplication of Q), because P removes them (= overapplication of Q), or because P reconfigures the form in a way that changes the locus of application of Q (the stress-epenthesis case). The Simultaneous Application Condition works for these cases because $P, Q(x))$ ignores these effects of P on Q, and thus $P, Q(x))$ has the effect of applying Q first but not P first.

This is illustrated below using the Bedouin Arabic and Polish examples discussed previously, for which the Simultaneous Application Condition correctly identifies Q as opaque when applied before P. For the Bedouin Arabic input /badw/ (9), applying raising before vocalization yields the same result as applying both simultaneously, and a different result than applying vocalization first. This is because raising is not applicable until after vocalization.

(9) P: vocalization; Q: raising

$$\begin{array}{ccccc} P(Q(\text{badw})) & = & P, Q(\text{badw}) & \neq & Q(P(\text{badw})) \\ \text{badu} & = & \text{badu} & \neq & \text{bidu} \end{array}$$

For the Polish input /ʒwob/ (10), applying raising before devoicing yields the same result as applying both simultaneously, and a different result than applying devoicing first. In this case, this is because raising is no longer applicable after devoicing.

(10) P: devoicing; Q: raising

$$\begin{array}{ccccc} P(Q(\text{ʒwob})) & = & P, Q(\text{ʒwob}) & \neq & Q(P(\text{ʒwob})) \\ \text{ʒwup} & = & \text{ʒwup} & \neq & \text{ʒwop} \end{array}$$

The Simultaneous Application Condition also correctly diagnoses the misapplication of stress in Palestinian Arabic as opaque. For the input /katabt/ (11), applying stress before epenthesis yields the same result as applying both simultaneously, and a different result than applying epenthesis first. Stress is applicable before and after epenthesis, but because epenthesis disrupts both the composition of syllables and their position relative to each other, stress falls on a different syllable if applied before epenthesis than if applied afterwards.

(11) P: epenthesis; Q: stress

$$\begin{array}{ccccc} P(Q(\text{katabt})) & = & P, Q(\text{katabt}) & \neq & Q(P(\text{katabt})) \\ \text{katábit} & = & \text{katábit} & \neq & \text{kátabit} \end{array}$$

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.

2.3 Further beyond the classical typology

The classical view summarized in Table (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. An interaction outside of its boundaries is identified by Baković (2007, 2011), who calls it SELF-DESTRUCTIVE FEEDING. Here and below, we will refer to such interactions more succinctly as SEEDING. A case based on examples from Turkish illustrates (Sprouse 1997). Here, epenthesis into final clusters (P) potentially interacts with intervocalic velar deletion (Q). Let us start with the second order listed in (12), 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 $Q \succ P$ order has the signature of counterfeeding with underapplication; it looks like [bebekin] should have undergone deletion, but it did not.

(12) P: $\varnothing \longrightarrow$ i / C ___ C# (epenthesis)
Q: k $\longrightarrow \varnothing$ / V ___ V (deletion)
✓($P \succ Q$): /bebekn/ $\xrightarrow{P}$ bebekin $\xrightarrow{Q}$ bebein 'baby.ACC' (seeding, P overapplies)
($Q \succ P$): /bebekn/ $\xrightarrow{Q}$ bebekn $\xrightarrow{P}$ bebekin (counterseeding, Q underapplies)

The reverse order $P \succ Q$ — the order actually found in Turkish — presents a problem for the Simultaneous Application Condition. On the one hand, P feeds 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 feeding interaction is expected given the apparent counterfeeding nature of the $Q \succ P$ order. However, the output [bebein] has the character of *overapplication* opacity: it looks like epenthesis should not have applied, but it did. 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 destruction by causing another rule to apply that destroys the environment of epenthesis (Baković 2007: 227). This type of interaction is an unexpected case of opaque feeding. It shares some properties of feeding, in that one rule creates inputs to another rule, and some properties of counterbleeding, in that the outcome is opaque with overapplication.[2]

Another interaction outside the bounds of the classical typology is what Baković (2007) refers to as NON-GRATUITOUS FEEDING, so named because an OT analysis of this type of interaction does not involve a 'gratuitous' violation of faithfulness (McCarthy 1999) and so it is a type of overapplication that can be analyzed in classic OT. In other respects, however, it is just like seeding. An example from Classical Arabic illustrates (McCarthy 2007b).

(13) P: $\varnothing \longrightarrow \alpha$V / # ___ CC αV (vowel copy epenthesis)
Q: $\varnothing \longrightarrow$ ʔ / # ___ V (ʔ-insertion)
✓($P \succ Q$): /ktub/ $\xrightarrow{P}$ uktub $\xrightarrow{Q}$ ʔuktub 'write.IMP.MSG' (seeding, P overapplies)
($Q \succ P$): /ktub/ $\xrightarrow{Q}$ ktub $\xrightarrow{P}$ uktub (counterseeding, Q underapplies)

[2]Indeed, both Moreton & Smolensky (2002: 315) and Potts & Pullum (2002: 384–385) misclassify the correct Turkish derivation in (12) as an instance counterbleeding, as noted by Baković (2007: 228).

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 $Q \succ P$ 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 $P \succ Q$ — the order actually found in Classical Arabic — again presents a problem for the Simultaneous Application Condition. On the one hand, P classically feeds 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 classical feeding interaction is again expected given the classically counterfeeding nature of the $Q \succ P$ order. However, the output [ʔuktub] again has the character of 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, was 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 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, the opaque derivations are the ones with classical counterfeeding, referred to above as counterseeding: epenthesis applying before deletion in Turkish, ʔ-insertion applying before vowel copy epenthesis in Classical Arabic; see (12) and (13).

(14) P: epenthesis; Q: deletion

$$\begin{array}{ccccc} Q(P(\text{bebekn})) & \neq & P,Q(\text{bebekn}) & = & P(Q(\text{bebekn})) \\ \text{bebein} & \neq & \text{bebekin} & = & \text{bebekin} \end{array}$$

(15) P: vowel copy epenthesis; Q: ʔ-insertion

$$\begin{array}{ccccc} Q(P(\text{ktub})) & \neq & P,Q(\text{ktub}) & = & P(Q(\text{ktub})) \\ \text{ʔuktub} & \neq & \text{uktub} & = & \text{uktub} \end{array}$$

There is no mystery as to why the seeding interaction with overapplication is not detected by the Simultaneous Application Condition. Generally, simultaneous application ignores the 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.[3]

[3]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 Q as opaque we need to show that no form x exists such that $Q(P(x)) \neq P,Q(x) = P(Q(x))$. For Turkish, there must be no form where the outcome of simultaneous application differs from the outcome of applying epenthesis (P) first. Indeed, in such a form, if it could exist, the application of epenthesis would depend on whether or not deletion has applied. But epenthesis requires a word-final CC# cluster, and intervocalic deletion is incapable of either destroying such a cluster (because neither of the Cs is intervocalic), or creating one (because no Cs outside of the VkV sequence are affected).

In sum, we have demonstrated above that the classical typology of feeding and bleeding and opacity fails to characterize seeding, which shares properties with feeding (creation of inputs) and counterbleeding (overapplication opacity). In the remainder of the article we will flesh out a formal typology that solves this problem, demonstrating precisely what seeding shares with feeding and what it shares with bleeding. In doing so, we will go far beyond the feeding/bleeding/seeding triad, and characterize the formal space of all possible pairwise process interactions. As a preliminary step to developing the typology, however, we need a formal and precise understanding of the notions of causing and preventing of application that underlie these concepts, and that is where we turn next.

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.[4] We will use the term MAPPING and the notation $a \xrightarrow{P} b$, $P(a) = b$, and (a, b) for individual input-output pairs. A VACUOUS mapping is a mapping where the input and output are segmentally identical, and the correspondence relation between the input and output is one-to-one and preserves precedence.[5]

In this paper, the notion of mapping is only formally defined for linear representations, i.e. strings of segments. While a full formalization of the present discussion 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 non-vacuously. Likewise, the OUTPUT SET of map P, written $Out(P)$, is the subset of Σ^* which have non-vacuous 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 a 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 (16).

[4] 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 $S \times T$.

[5] Formally: let x be a string of length n, consisting of pairs (i, s_i) for $1 \leqslant i \leqslant n$ (see fn. 4). The mapping $P(x)$ is vacuous if $P(x) = x$, and the correspondence relation for this mapping consists of $((i, s_i), (i, s_i))$ for all i between 1 and n.

(16) 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 (16) refer to relations between entire maps, those relations are evidenced by specific strings that fit the requirements of (16). In common practice, the terms *feeding* and *bleeding* are ambiguous: they can either be used globally to refer to map relations, or locally and specifically to refer to individual mappings, and we will continue this usage. In particular, there is an expectation that the definitions in (16) work in reverse, as it were: it should be true that any sequence $x \xrightarrow{P} y \xrightarrow{Q} z$ that we understand pretheoretically to involve feeding or bleeding should fit the set membership definition in (16). As we will see, this is where the naive set-theoretic idea in (16) 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 from the previous section. 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. (Of course, in both of these examples Q is extrinsically ordered before P, leading to the observed counter-orders).

There is also promise in the observation that these definitions do not apply to stress-epenthesis interactions like the Palestinian Arabic example discussed above. 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. This reinforces our contention in §2.2 that feeding and bleeding are not appropriate to characterize those cases.

However, even outside of stress rules, there are situations where the definitions in (16) 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 processes.

(17) 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, the palatalization of the second [k] would not have applied. But this example doesn't work in the set-theoretic terms of (16): 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 (16). 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.[6]

[6]Another strategy might be to try to keep track of the loci of application of rules, to be able to compare

3.3 String input-rank

ounting 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 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 non-vacuous break (NVB).

(18) Let P be a map, and x a string in $In(P)$. Then strings $a_1, a_2, \ldots, a_n$ are called a NON-VACUOUS 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 non-vacuous applications of P that are independent of each other. For velar palatalization, the string /taki/ has only a trivial NVB of rank 1, /taki/. The string /kiki/ has a trivial NVB of rank 1, /kiki/, and an 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.

(19) Let P be a map and x a string. The input-rank of x, written $\mathsf{IR}(P, x)$ is:

a. 0 if $x \notin In(P)$;

b. otherwise, the highest possible NVB rank of x for map P.

Input-rank thus partitions the set Σ^* into non-overlapping subsets: those strings to which the rule doesn't apply at all, those strings to which the rule applies once, to which it applies twice, etc., for example, as shown in (20) for velar palatalization (k $\longrightarrow$ č / ___ i). For many maps, there is in principle no upper limit for input-rank. This is true for velar palatalization, and in fact for most ordinary segmental maps. Indeed, for a map specified by the rule $a \longrightarrow b$ / c ___ d, strings with arbitrarily large input-ranks can be constructed by concatenating *cad* arbitrarily many times.

applications of one map across strings and thus make it possible to determine whether rule P causes a new instance of application of Q. This approach was pursued in earlier versions of this work ([self-citation omitted for review]). However, it has proven difficult to make coherent for maps involving deletion and insertion of segments.

(20)

string examples	input-rank
pa ku banana	0
ki taki kibulipami	1
kiki kipaki takiki	2
...	
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, the condition (18b) of being an NVB isn't satisfied: P(ad)P(ad) = atat ≠ 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 selects a single, privileged member of a string. In practice, such rules use words like "leftmost" or "rightmost" in their description. This is the case for rules like final devoicing and for stress rules. Further afield, consider a version of a segmental rule like palatalization that only applies to the *rightmost* eligible string, so /kipa/ → [čipa], but /kiki/ → [kiči]. This rule has no strings with input-rank higher than 1.

3.4 Input-provision, input-removal

We can now understand feeding as an increase in the input-rank of a string, and bleeding as a decrease. The following definitions state this idea. Instead of calling these interactions 'feeding' and 'bleeding', we use the terms INPUT-PROVISION and INPUT-REMOVAL for reasons that will become clear in §3.5.

(21) a. P input-provides 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 if there is a mapping $x \xrightarrow{P} y$ such that $\mathsf{IR}(Q, x) > \mathsf{IR}(Q, y)$.

In such cases, we will say that the pair (x, y) or the mapping $x \longmapsto y$ contains input-provision or input-removal, or that P input-provides/removes Q at (x, 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 $\longrightarrow$ či), and Q final devoicing (d# $\longrightarrow$ t#).

(22) Examples of input-rank (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. Map P affects (input-provides, input-removes) another map Q if there are unequal Q-input-ranks at the opposite ends of a P-mapping.

In the following diagrams, 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. Figure 1 shows a case with no interaction; input-provision and input-removal are illustrated in figures 2 and 3, respectively.

In these examples, the numbers are 0 and 1, but they could be something else, 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 a non-trivial interaction. (At the opposite ends of the P-arrow, the numbers of the P-input-ranks are also unequal, but this is irrelevant to how P affects Q. What's relevant is how one map affects the input-ranks of another map). To highlight the relevant comparison, a thick dashed line is drawn between the boxes with unequal ranks. The labels 'P+iQ' and 'P–iQ' indicate input-provision and input-removal, respectively.

$$\begin{array}{ccc} \text{input}\ \ 1 & \longrightarrow & \text{output}\ \ 0 \\ \boxed{1} & & \boxed{1} \end{array}$$

Fig. 1: No interaction

$$\begin{array}{ccc} \text{input}\ \ 1 & \xrightarrow[P+\mathrm{i}Q]{} & \text{output}\ \ 0 \\ \boxed{0} & & \boxed{1} \end{array}$$

Fig. 2: Input-provision

$$\begin{array}{ccc} \text{input}\ \ 1 & \xrightarrow[P-\mathrm{i}Q]{} & \text{output}\ \ 0 \\ \boxed{1} & & \boxed{0} \end{array}$$

Fig. 3: Input-removal

These diagrams highlights a formal point: input-provision and input-removal are, strictly speaking, properties of mappings, and not properties of strings or of rules as a whole. Two entire maps P and Q, each being a collection of mappings, may contain multiple instances of interactions like P+iQ and P–iQ, each originating with a particular mapping, or arrow in our diagrams.

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

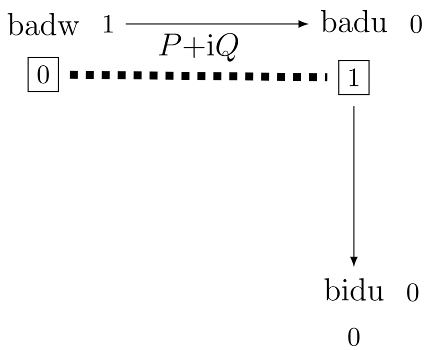


Fig. 4: Feeding: input-provision

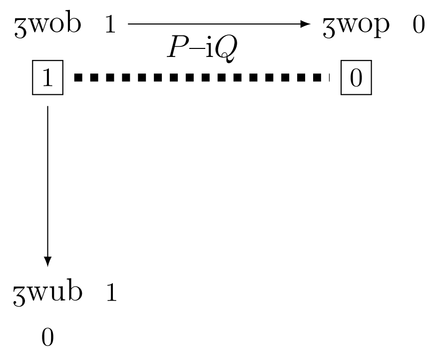


Fig. 5: Bleeding: input-removal

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 increases the string's palatalization input-rank from 1 to 2, as shown below.

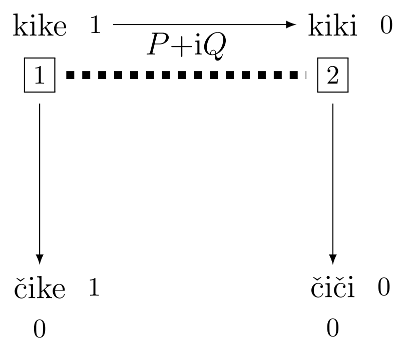


Fig. 6: Multiple-loci: input-provision

The definition as proposed above fits closely with the pretheoretical notion of feeding and bleeding, and produces satisfactory results in almost all cases. Yet, because it only counts applications of a map in a string without tracking specific loci, it produces some awkward consequences under some very special circumstances, not threatening to the overall picture proposed here, as described in this footnote.[7]

[7]Consider P: V $\longrightarrow \varnothing$ /— V (vowel deletion), and Q: s $\longrightarrow$ ʃ /— i (palatalization). 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, sketched here but not developed rigorously. The composition of maps P and Q is also a map, call it R. As any other map, it 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 non-vacuously, resulting in [si, ʃa],

The typology we have introduced so far, while giving a more solid formal foundation to the notions of feeding and bleeding, is still not able to distinguish classical feeding from seeding. Consider the Turkish example again. In the framework described so far, it is indistinguishable from feeding, as the following figure shows.

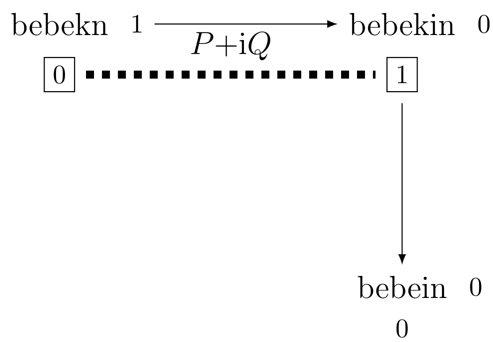


Fig. 7: Seeding: input-provision

We claim the following: the picture is incomplete because we are only considering the input properties, whether strings are possible inputs to a map. We should also consider output properties: whether strings are possible outputs of a map. To this end we define output interactions parallel with input interactions.

3.5 Output interactions

In this subsection we show that investigating the effect of one process on the *output* properties of another process brings linguistic insight on a par with investigating effects on inputs. In particular, output interactions unlock the full typology of map interactions, as we will see in §5, and they also play a key role in formally defining opacity, as discussed in §4.

Pretheoretically, just like input interactions can be thought of in terms of one process P adding or removing members of 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 $Out(Q)$. However, this set-based approach has some limitations that were discussed above, and we have pursued a more technical method using the concept of input-rank. Likewise for output interactions, set membership is a good intuitive approximation of the effects, but a full formal account must rely on the parallel concept of OUTPUT-RANK, which is defined and discussed in the remainder of this section.

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

respectively, and $R(\text{saisia}) = R(\text{sai})R(\text{sia}) = \text{siʃa}$. So, we would augment the definitions of input-provision and input-removal as follows: instead of applying them to the whole string, they are applied separately to the NVB substrings for the composition of P and Q, in case there is an NVB of rank higher than 1. This kind of situation 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.

a given map. For example, the string [ki] has no possible input for the velar palatalization map (k $\longrightarrow$ č / ___ i). We will define the output-rank of such strings to be 0.

Second, strings typically have both vacuous and non-vacuous inputs. For velar palatalization, [či] has two possible inputs, /ki/ and /či/, the latter with a vacuous mapping. The vacuous input /či/ has input-rank 0, while the non-vacuous input /ki/ has input-rank 1. In such cases our definition will ignore the vacuous mapping and its input-rank, ensuring that a string like [či] has output-rank of 1 for the velar palatalization map. Sometimes a string may have more than one non-vacuous input with input-rank 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 non-vacuous input with various ranks. Such is the case for [čiči], which can be non-vacuously derived from /kiki/, /kiči/ or /čiki/, with input-ranks of 2, 1, and 1, respectively.

The upshot is that the output-rank of a string can only be defined for a particular non-vacuous input of that string. This is captured in the following definition.

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

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

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

Once again, parallel to input-rank, output-rank will measure how many times a map has potentially applied to a string. Thus, we can define output-provision and output-removal parallel to input-provision and input-removal: when one map P increases the output-rank of a string x for another map Q, P output-provides Q (with respect to x), and when P decreases the output-rank of a string for Q, P output-removes Q (with respect to x).

In the simplest case, when the strings have output-ranks of either 0 or 1, the definitions are straightforward. For example, a final-vowel raising map applying to /če/ and producing [či] would increase the output-rank of that string for the velar palatalization map from 0 (because [če] lacks a non-vacuous input for velar palatalization) to 1 (because [či] has a non-vacuous velar palatalization input /ki/). In such cases the increase or decrease of Q's output-rank is equivalent to the addition or removal of a member of the set $Out(Q)$. In informal discussion of examples we may refer to the effects in these terms. However, here we offer a more rigorous definition of output-provision and output-removal in terms of output-rank, starting with the simpler case where each string may only have an output-rank of 1 or 0, and expanding it to more complex situations where set membership is insufficient.

(24) Output interactions for strings with output-ranks of 0 or 1.

a. P output-provides Q if there is a mapping $x \xrightarrow{P} y$ such that $\mathsf{OR}(Q, \epsilon, x) = 0$ and $\mathsf{OR}(Q, y', y) = 1$ for some y'.

b. P output-removes Q if there is a mapping $x \xrightarrow{P} y$ such that $\mathsf{OR}(Q, x', x) = 0$ and $\mathsf{OR}(Q, \epsilon, y)$ for some x'.

In such cases we say that the pair (x, y) or the mapping $x \longmapsto y$ contains output-provision or output-removal, or that P output-provides/removes Q at (x, y).

Parallel to the labels for input interactions P+iQ and P–iQ, we will use P+oQ and P–oQ to refer to output-provision and output-removal.

In the more complex case with multiple possible output-ranks, we need to define output-provision and output-removal in such a way that it captures the *causation* of the increase or decrease of the output-rank of one map by another. Comparing two strings, x and y, an output-rank of one string is UNMATCHED if there is no equal output-rank of the other string. Output-provision and output-removal are then defined as appearance or disappearance of unmatched mappings.

(25) **Unmatched mapping.** Let $x \xrightarrow{P} y$ be a mapping, and let $x' \xrightarrow{Q} x$ be a mapping such that $\mathsf{OR}(Q, x', x) \geqslant 1$. The mapping $x' \xrightarrow{Q} x$ is UNMATCHED if there is no mapping $y' \xrightarrow{Q} y$ such that $\mathsf{OR}(Q, x', x) = \mathsf{OR}(Q, y', y)$.

(26) If for some x', y' (possibly ϵ) such that at least one of $x' \xrightarrow{Q} x$ or $y' \xrightarrow{Q} y$ is unmatched,

a. $\mathsf{OR}(Q, x', x) < \mathsf{OR}(Q, y', y)$, then P output-provides Q;

b. $\mathsf{OR}(Q, x', x) > \mathsf{OR}(Q, y', y)$, then P output-removes Q.

In such cases we say that the pair (x, y) or the mapping $x \longmapsto y$ contains output-provision or output-removal, or that P output-provides/removes Q at (x, y).[8]

The output-interactions P+oQ and P–oQ, like corresponding input-interactions, are strictly speaking properties of individual mappings. Each mapping may or may not contain such an interaction depending on the comparison of output-ranks as laid out in (25) and (26). An entire pair of maps P and Q may contain multiple instances of these interactions, each originating from a particular mapping.

The intuitive meaning of output-provision and output-removal is harder to pin down than the input interactions, so let's turn to some examples.

Although not usually recognized as such, classical feeding and bleeding involve output interactions. In the case of feeding, there is output-provision. With reference to the Bedouin Arabic example in (5) again, 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, but the output [bidu] *is* a possible outcome of raising, with an output-rank of 1 (cf. input /badu/).

In the case of bleeding, there is output-removal. With reference to the Polish example in (6) again, this can be seen in the mapping /ʒwub/ $\xrightarrow{P}$ [ʒwup]. 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.

These interactions are illustrated in the diagrams below. 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. So, considering two maps, P and Q, every string in

[8] It is possible that there are multiple unmatched ranks that satisfy both the (26a) and (26b) parts of the definition, e.g. if the output-rank of x is 2, while y has two output-ranks, 1 and 3. In such a case the mapping (x, y) would contain both provision and removal. We are not aware of realistic configurations with this structure.

our diagrams will be surrounded by four numbers showing input-ranks and output-ranks for these two maps.

In general, in order to determine how P affects Q (that is, which of the four basic interactions $\{P$+iQ, P–iQ, P+oQ, P–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: $\{Q$+iP, Q–iP, Q+oP, Q–o$P\}$. This is determined, *mutatis mutandis*, by comparing P-ranks vertically across a Q arrow.

In the diagrams below, feeding and bleeding are shown to contain pairs of basic interactions. In the case of feeding there are two provision interactions $\{P$+iQ, P+o$Q\}$, and in the case of bleeding there are two removal interactions $\{P$–iQ, P–o$Q\}$. There are no other P-on-Q interactions present here, and no Q-on-P interactions at all. Again, unequal ranks that determine interactions are boxed and connected by dashed lines.[9]

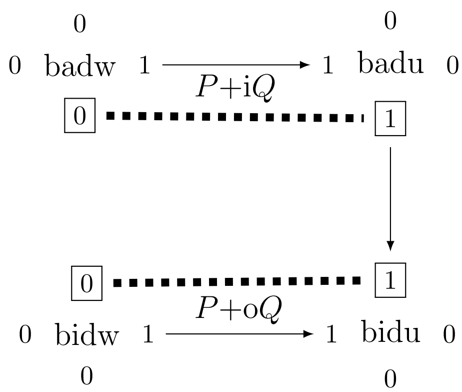


Fig. 8: Feeding

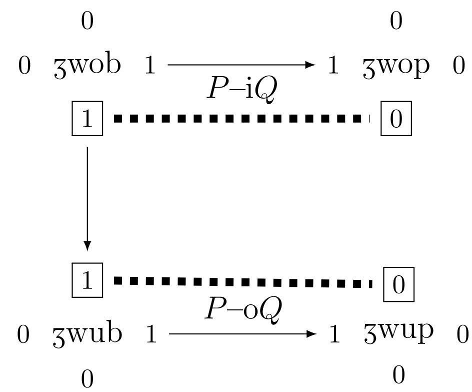


Fig. 9: Bleeding

We take the configurations illustrated in Figs. 8 and 9 to be definitional of feeding and bleeding, as follows.

(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).

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 input like /bidw/ is irrelevant at the level of idealization

[9]The careful reader may have noted that the shape of the bleeding graph in Fig. 9 resembles our example of multiple loci of application (6), but this resemblance is superficial because the details of input-ranks and output-ranks are different in the two examples, and thus (6) is not an example of bleeding.

assumed here. The same applies to all subsequent examples where possible/potential as opposed to actual forms are introduced into these diagrams in order to properly classify the rules and their interactions.

Let's now consider the Turkish seeding example in (12) from the point of view of output interactions. Recall from Fig. 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 (VV) that potentially arose due to deletion, and P cannot create strings that contain hiatus. Yet, P+iQ is not the only interaction here: Q also output-removes P. Indeed, Q takes a string containing V*kin*, which is a possible output of P, and outputs the string V*in*, which is not. The following diagram illustrates.

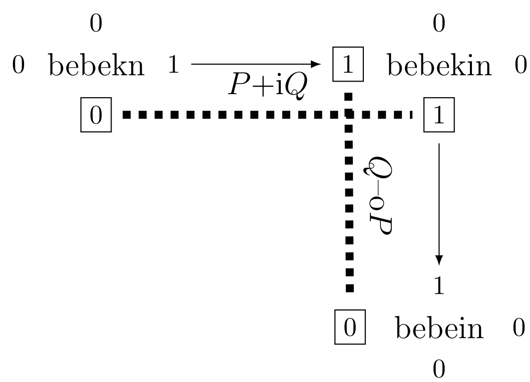


Fig. 10: Seeding

Seeding thus shares formal characteristics with both feeding and bleeding: like feeding, it has input-provision, and like bleeding, it has output-removal. As we will see, this formal property-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 Fig. 10 is definitional of seeding, as follows.

(28) P seeds Q at 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 Atoms and molecules

It is useful at this point to think of the interactions we have discussed thus far (feeding, bleeding, and seeding) as "molecules", each of which contains two "atoms" — more basic interactions of provision and removal. This is summarized in the following table.

(29)

atoms / molecules	+i	+o	–i	–o
P feeds Q	P+iQ	P+oQ		
P bleeds Q			P–iQ	P–oQ
P seeds Q	P+iQ			Q–oP

Viewed this way, seeding shares one atom (+i) with feeding, and one (–o) with bleeding. This is the key to making sense of what seeding intuitively shares in common with feeding and with the overapplication aspect of counterbleeding.

Recall that the seeding interaction is opaque no matter what the order: if $P \succ Q$, then there is overapplication, while if $Q \succ P$, there is underapplication (see (12)). This allows us to arrive at a provisional characterization of these two types of opacity, as follows.

(30) For two maps X and Y, when $X \succ Y$,
  a. X underapplies if Y+iX;
  b. X overapplies if Y–oX.

The two atoms that give rise to opacity are +i and –o. If map P creates inputs (i.e. increases the input-rank of some form) for map 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 feeding and seeding: 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 still has applied, there is overapplication. This is what happens with bleeding: if Q applies first, there is overapplication. Likewise with seeding, but with P and Q reversed: since Q–oP, if P applies first, there is overapplication.

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.

4.2 Accomodating misapplication

Consider now the third form of opacity discussed in §2.2, in the context of stress-epenthesis interactions, which we called *misapplication*. In those cases a process applies in an unmotivated locus, such as stress ending up on the 'wrong' syllable in the surface form. This type of opacity is neither classical underapplication nor classical overapplication, and neither feeding nor bleeding. Where does it 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 non-vacuous in the case of misapplication ("the rule should have applied in a different way than it did"). Formally, as it turns out, misapplication has the same signature as overapplication: the –o atom. Let us return to the Palestinian Arabic stress-epenthesis 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 $Q \succ P$.

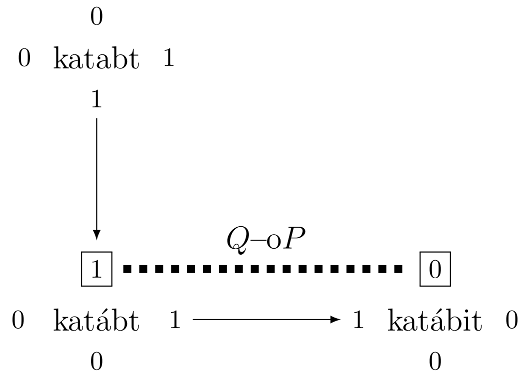


Fig. 11: Palestinian Arabic stress and epenthesis: output-removal

Note the interaction P–oQ in this diagram, now understood as the signature of both misapplication and overapplication. Using the term *misapplication* to cover both types of phenomena, with overapplication being a special case, we arrive at the following generalized characterization of opacity.

(31) For two maps X and Y, when $X \succ Y$,
  a. X underapplies if Y+iX;
  b. X misapplies if Y–oX.

Thus, when $Q \succ P$ and P–oQ in Palestinian, 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 Fig. 11. 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. Thus, the stress-epenthesis interaction presents a new, unique combination of atoms, unlike those of feeding, bleeding, or seeding: P+oQ, P–oQ. The convoluted, snail-like arrangement of inputs and outputs is shown below. In the fuller typology in §5, it will be called TRANSFUSION (a term due to Zwicky 1987, citing unpublished work by Don Churma).[10]

[10] Rasin (to appear) refers to such interactions as SHIFTING; see also Pruitt (to appear).

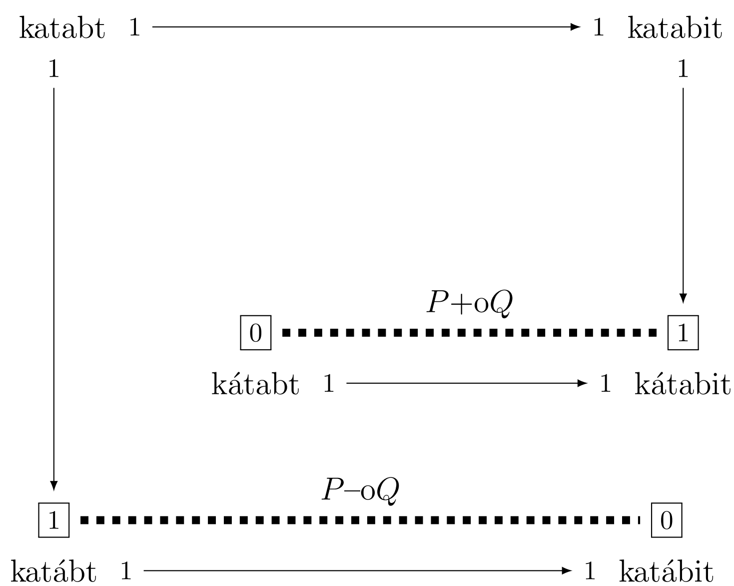


Fig. 12: Transfusion in Palestinian Arabic: the full picture

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 is ordered first and P–oQ. 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. Here is a hypothetical example, 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)
($P \succ Q$): /katb/ $\xrightarrow{P}$ kátb $\xrightarrow{Q}$ kátib (misapplication of stress)
($Q \succ P$): /katb/ $\xrightarrow{Q}$ katb $\xrightarrow{P}$ kátb (underapplication of epenthesis)

Stress seeds epenthesis in this example, and the subsequent application of epenthesis then moves stress to an unexpected location, i.e. with misapplication. This happens because epenthesis output-removes stress, as shown in the following figure.

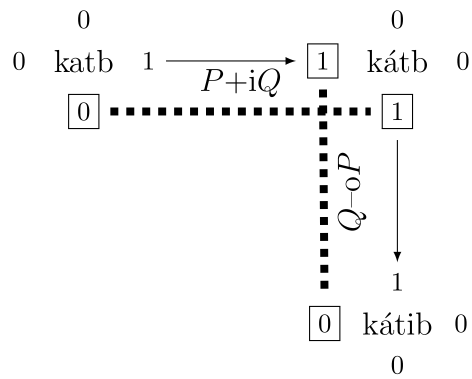


Fig. 13: Seeding with misapplication

Intuitively, it appears that stress rules typically display misapplication while segmental rules display overapplication. No doubt this difference has some formal underpinnings, but we leave their investigation for another day.

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).[11]

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

Q: $[\text{+dorsal}] \longrightarrow \begin{bmatrix}\text{–back}\\ \text{+del. rel.}\end{bmatrix} \; / \; \_\_ \begin{bmatrix}\text{–cons}\\ \text{–back}\end{bmatrix}$ (palatalization)

$(P \succ Q)$: /vagit͡ɕ $\xrightarrow{P}$ vagit͡ɕ $\xrightarrow{Q}$ [vad͡ʒit͡ɕ] (counterseeding, P underapplies)

✓$(Q \succ P)$: /vagit͡ɕ/ $\xrightarrow{Q}$ vad͡ʒit͡ɕ $\xrightarrow{P}$ [vaʒit͡ɕ] 'weigh.INF' (seeding, Q overapplies)

The correct Polish derivation in (33), with $Q \succ P$, is an example of seeding. Q input-provides P by supplying new voiced velar affricates, while P in turn output-removes Q because [d͡ʒ] but not [ʒ] is a possible output of velar palatalization. (As expected for seeding, the reverse order $P \succ Q$ exhibits underapplication of P, with possible outputs like [d͡ʒ] that look like they should have undergone spirantization but haven't.) Misapplication of palatalization, resulting from the Q–oP atom, is expected here by (31). Compared to the Palestinian Arabic transfusion-and-misapplication example in (7) and to the Turkish seeding-and-overapplication example in (12), however, it is intuitively 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 [ʒi] 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 segment is a fricative, lacking the expected [–cont] value.

[11] In the actual Polish example there is also non-derived environment blocking of spirantization, but this is orthogonal to questions of ordering.

4.5 Summary

This preliminary discussion of feeding, bleeding, seeding, and transfusion interactions illustrates that the accepted classification of map interactions is the result of different combinations of more basic, atomic elements — the four basic atoms $\{$+i, –i, +o, –o$\}$.

This observation paves the way to our exhaustive typology of interactions. To achieve that goal, we need to investigate what kinds of atom combinations are possible and how they are instantiated in interactions. This is the task we take up in the following sections, starting bottom-up with a kind of enumeration in §5, and then proceeding top-down with a theorem about possible atom combinations in §6.

5 Typology

We have thus far demonstrated that the types of interactions traditionally known as *feeding* and *bleeding* are in fact composite — they are molecules, not atoms, consisting of various combinations of more elementary, or atomic, components. There are four such 'atoms' we have defined: input provision (+i), input removal (–i), output provision (+o), and output removal (–o). The next natural step is to flesh out the full typology of the possible combinations of such atoms. Are feeding, bleeding, seeding, and transfusion the limit? If not, what else is possible? Can atoms occur on their own? What other molecules are out there? We begin here with our bottom-up investigation of the typology.

5.1 Corners

Minimally, a non-trivial 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 Fig. 14–17. Irrelevant null input-ranks and output-ranks are not shown here nor in other figures in the remainder of this section.

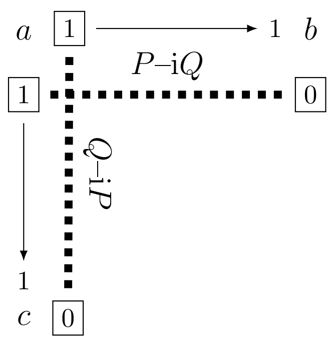


Fig. 14: Upper left corner

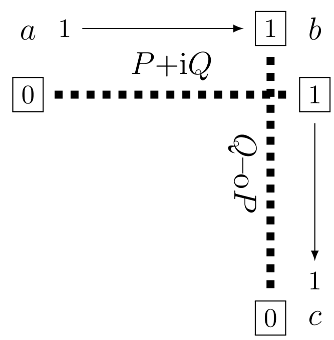


Fig. 15: Upper right corner

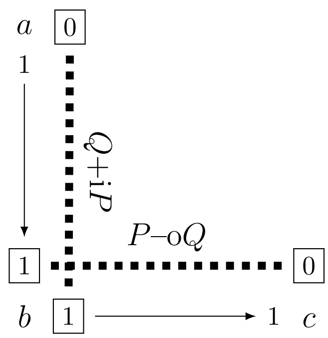


Fig. 16: Lower left corner

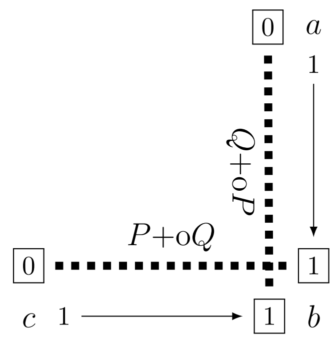


Fig. 17: Lower right corner

Figs. 14–17 serve as visual aids to finding atoms in more complex configurations. We will refer back to them frequently. 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 (Fig. 15) and lower left corner (Fig. 16), equivalent up to names of P and Q, instantiate seeding, as discussed in the previous section: these corners consist of input-provision in one direction and output-removal in the other.

There are two other topologically distinct corners besides seeding: the upper left corner (Fig. 14) and the lower right corner (Fig. 17). These new corners reflect natural categories of interactions: MUTUAL BLEEDING and MERGER. Each contains two atoms: $\{P{+}\mathrm{o}Q,\ Q{+}\mathrm{o}P\}$ for merger, and $\{P{–}\mathrm{i}Q,\ Q{–}\mathrm{i}P\}$ for mutual bleeding.

The following patterns can serve as examples illustrating these two types of interactions. 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).

(34) P: $\begin{bmatrix}-\text{son}\\-\text{cont}\\+\text{cor}\end{bmatrix} \longrightarrow \varnothing$ / ___ C# (dental stop deletion)

Q: [+lat] $\longrightarrow \varnothing$ / C ___ # (lateral deletion)

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

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

Merger is exemplified by a pattern in Spanish (Harris 1969), where nasals become coronal word-finally (P) and nasals assimilate in place to following consonants, including across a word boundary (Q).

(35) P: [+nas] $\longrightarrow \begin{bmatrix}+\text{cor}\\+\text{ant}\end{bmatrix}$ / ___ # (nasal coronalization)

Q: [+nas] $\longrightarrow$ [αplace] / ___ (#) $\begin{bmatrix}+\text{cons}\\\alpha\text{place}\end{bmatrix}$ (nasal place assimilation)

These examples are illustrated in the comparison graphs below. Evidence for Spanish words ending underlyingly in non-coronal nasals is slim at best (Harris 1999), and minimal pairs on this basis are all but non-existent, so the 'examples' in Fig. 19 are schematic.

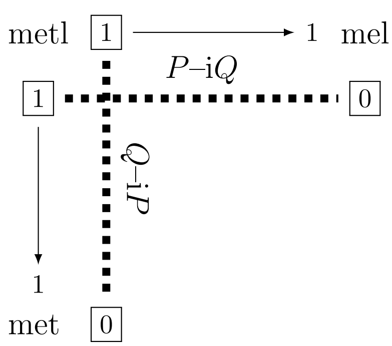


Fig. 18: Mutual bleeding

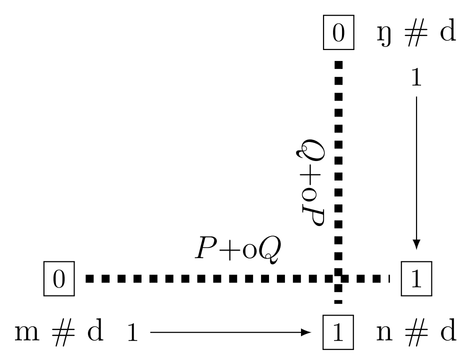


Fig. 19: Merger

To represent the topology of the graphs, we can flatten the seeding graph to ·→·→·, where dot · stands for a string, and each arrow represents a non-vacuous mapping. The other two corners take the shapes ·←·→· (mutual bleeding) and ·→·←· (merger).

(36)

arrows	shape	interactions	Fig.	structure	atoms
2	corners	seeding	15	·→·→·	P+iQ, Q–oP
		mutual bleeding	18	·←·→·	P–iQ, Q–iP
		merger	19	·→·←·	P+oQ, Q+oP

As expected, since neither mutual bleeding nor merger contain the crucial opacity-generating atoms +i and –o, there are no opaque interactions no matter what the order of P and Q for these two types.

5.2 Pies

Feeding and bleeding are structurally more complex than corners in that they minimally contain three, as opposed to two, non-vacuous mapping arrows. They have already been shown above in Figs. 8 and 9, repeated here in Figs. 20 and 21, respectively.

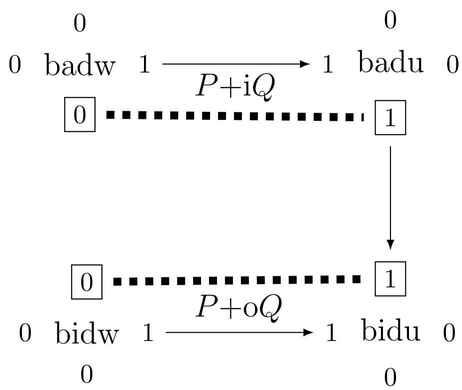


Fig. 20: Feeding

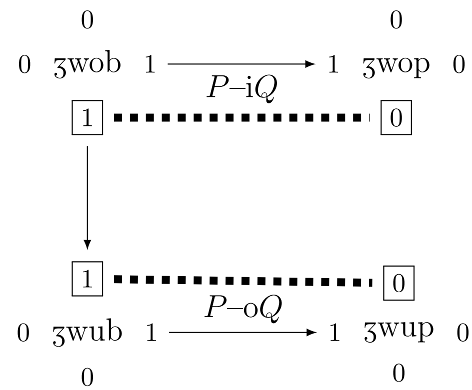


Fig. 21: Bleeding

These configurations will be referred to as PIES (after the shape of the Greek letter Π). They can be flattened to ·→·→·←· and ·←·→·→·, respectively, and there are only two such pies — all other sequences of three arrows are topologically equivalent.

(37)

arrows	shape	interactions	Fig.	structure	atoms
2	corners	seeding	15	·→·→·	P+iQ, Q–oP
		mutual bleeding	18	·←·→·	P–iQ, Q–iP
		merger	19	·→·←·	P+oQ, Q+oP
3	pies (Πs)	feeding	20	·→·→·←·	P+iQ, P+oQ
		bleeding	21	·←·→·→·	P–iQ, P–oQ

It appears at first blush that ·→·→·→· should be a distinct structure, but it is not, as the graph in Fig. 22 should make clear. This configuration simply consists of two corners, the upper right (cf. Fig. 15; P seeds Q) and lower left (cf. Fig. 16; Q seeds P). It contains two pairs of atoms, each pair contributed by the separate seeding interaction. This is the structure of *fed counterfeeding* interactions (Kavitskaya & Staroverov 2010); see §5.6.

More generally, structures under consideration as basic interactions are substrings of ·→·→·←·←· repeated periodically, i.e. paths along the square. The structure ·→·→·→· from Fig. 22 is not such a substring.

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 without closing the square — or if arrows are added at both ends of a corner — the structure becomes an OPEN JAW. For a concrete illustration, Fig. 23 shows an open-jaw interaction between the following two hypothetical processes.

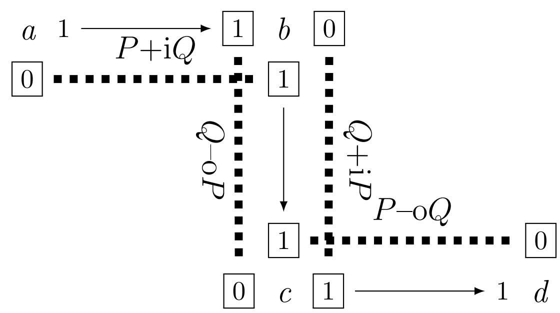


Fig. 22: Two-corner composite interaction

(38) P: L-to-R quantity-sensitive iamb (initial if long, else peninitial) (stress)
Q: unstressed aː → a / # __ (shortening)
($P \succ Q$): /aːta/ $\to_P$ áːta $\to_Q$ áːta
($Q \succ P$): /aːta/ $\to_Q$ ata $\to_P$ atá

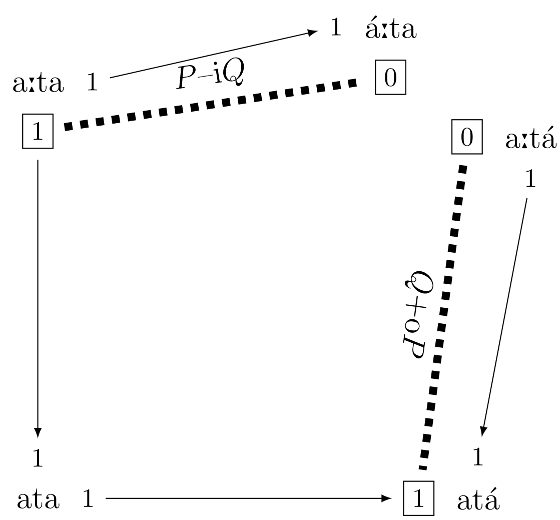


Fig. 23: Hyper seeding

We refer to this open jaw interaction as HYPER SEEDING, because it includes as a substructure the seeding structure ·→·→· in the lower left corner. But unlike seeding, the interaction is transparent no matter the order of the maps. If initial stress applies first, it prevents shortening from applying; if shortening applies first, it shifts stress assignment from the initial vowel to the peninitial. This mutual transparency is consistent with the absence of opacity-generating atoms, +i and –o, in Fig. 23.

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. 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. They are illustrated in schematic form in Figs. 24–25 below. The atoms of these open jaws are symmetrical with respect to P and Q. Fig. 24 represents mutual output removal, and includes as a substructure the mutual bleeding structure $\cdot\!\leftarrow\!\cdot\!\rightarrow\!\cdot$ in the upper right corner. We will thus refer to this open jaw interaction as HYPER MUTUAL BLEEDING. Fig. 25 represents mutual input provision, and includes as a substructure the merger structure $\cdot\!\rightarrow\!\cdot\!\leftarrow\!\cdot$ in the lower right corner. We will thus refer to this open jaw interaction as HYPER MERGER.

$$\begin{tikzcd} a \arrow[r] \arrow[dd] & b \arrow[d, "Q\text{–o}P"] \\ & c \\ e \arrow[r, "P\text{–o}Q"] & d \end{tikzcd}$$

Fig. 24: Hyper mutual bleeding

$$\begin{tikzcd} a \arrow[r, "P\text{+i}Q"] & b \arrow[dd] \\ e \arrow[d, "Q\text{+i}P"] & \\ d \arrow[r] & c \end{tikzcd}$$

Fig. 25: Hyper merger

A hyper mutual bleeding interaction is evident in a case based on examples from Dakota (Shaw 1980, 1985, Kennedy 1994, Alderete 1999) with the rules given in (39). 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 otherwise no coda consonants in Dakota so epenthesis is probably syllabically conditioned; this detail is irrelevant here.)

(39) P: L-to-R iamb (peninitial if polysyllabic, else initial) (stress)
Q: $\varnothing \longrightarrow$ a / C __ # (epenthesis)
✓($P \succ Q$): /ček/ $\xrightarrow{P}$ čék $\xrightarrow{Q}$ čéka 'stagger' (P misapplies)
($Q \succ P$): /ček/ $\xrightarrow{Q}$ čeka $\xrightarrow{P}$ čeká (Q misapplies)

Unlike mutual bleeding, the interaction is opaque no matter the order of the maps. The correct $P \succ Q$ order renders stress opaque, because stress ends up on the initial syllable despite the fact that there is a second syllable for it to fall on. With the opposite (incorrect) $Q \succ P$ 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 Fig. 26.

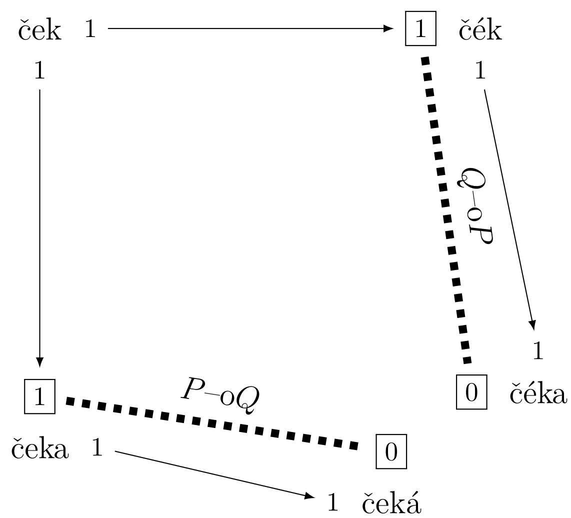


Fig. 26: Hyper mutual bleeding in Dakota

Hyper merger, unlike merger, is also expected to be doubly opaque, though this time with underapplication (two +i atoms). We are presently unaware of any examples, nor can we imagine any, that might instantiate hyper merger.

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

(40)

arrows	shape	interactions	Fig.	structure	atoms
2	corners	seeding	15	·→·→·	P+iQ, Q–oP
		mutual bleeding	18	·←·→·	P–iQ, Q–iP
		merger	19	·→·←·	P+oQ, Q+oP
3	pies (Πs)	feeding	20	→·→·←	P+iQ, P+oQ
		bleeding	21	←·→·→	P–iQ, P–oQ
4	open jaws	hyper seeding	23	←·←·→·→	P–iQ, Q+oP
		hyper m-bleeding	24	→·←·←·→	P–oQ, Q–oP
		hyper merger	25	→·→·←·←	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 example has already been witnessed with the transfusion interaction between stress and epenthesis in Palestinian Arabic, in Fig. 12. A structually identical snail is attested as an interaction between long vowel lowering (P) and vowel copy epenthesis (Q) in Yokuts (Kuroda 1967, Steriade 1986), as shown in (41). Note that the atom set in Fig. 27, P+oQ and P–oQ, is identical to the set found for Palestinian Arabic stress and epenthesis.

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

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

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

✓($Q \succ P$): /sduːk-/ $\xrightarrow{Q}$ suduːk- $\xrightarrow{P}$ sudoːk- ‘remove’ (Q misapplies)

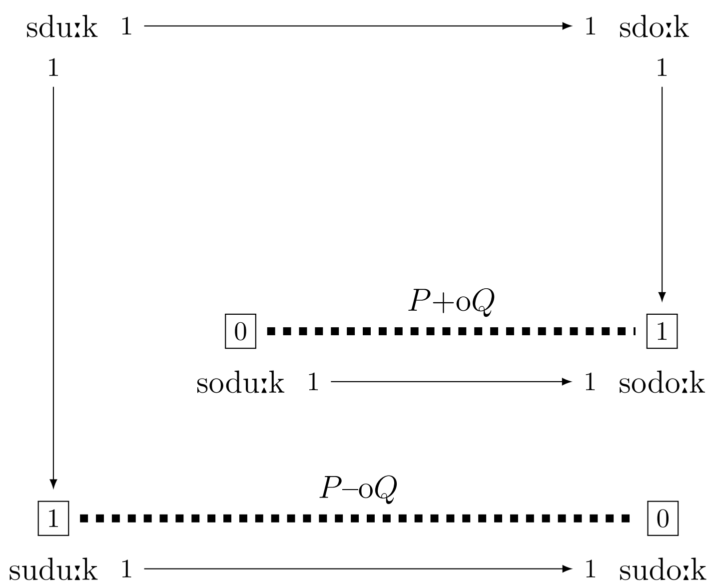


Fig. 27: Transfusion in Yokuts

As with the Palestinian Arabic example, Yokuts transfusion involves the atom P–oQ, and thus shows misapplication in case Q applies first. This is what happens in the mapping /sduːk-/ $\xrightarrow{Q}$ suduːk- $\xrightarrow{P}$ sudoːk-: the copy epenthesis rule misapplies by inserting the apparently wrong vowel, [u] instead of [o].

Besides transfusion, there is one other distinct type of snail. It can be obtained by reversing the direction of Yokuts epenthesis (41): instead of inserting a vowel copy, the anti-Yokuts rule would delete the first of two copies of a vowel. The resulting interaction has the atoms P+iQ, P–iQ, and the structure shown in Fig. 28. We will refer to it as INFUSION. Because the example is hypothetical, only the vowels are shown in Fig. 28.

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

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

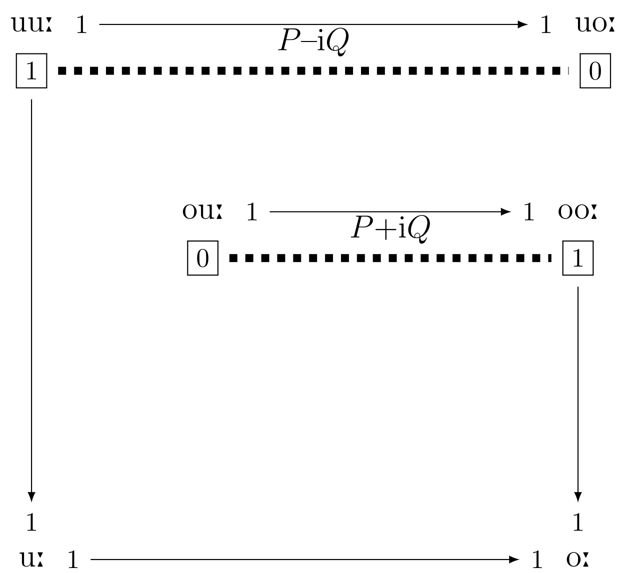


Fig. 28: Infusion in Anti-Yokuts

Infusion contains the atom P+iQ, and thus is expected to show underapplication with the order Q before P. Indeed, in case the input is /ouː/, Q fails to apply, because Q only deletes a vowel if the following vowel is identical. Then P applies, creating *oo*:, to which Q could apply but doesn't — classic underapplication.

The full table of possible two-atom interactions is in (43), including the expected opacity type (depending on the order of P and Q): U for underapplication, M for misapplication.

(43)

arrows	shape	interactions	Fig.	structure	atoms	opacity
2	corners	seeding	15	·→·→·	P+iQ, Q–oP	UM
		mutual bleeding	18	·←·→·	P–iQ, Q–iP	
		merger	19	·→·←·	P+oQ, Q+oP	
3	pies (Πs)	feeding	20	·→·→·←·	P+iQ, P+oQ	U
		bleeding	21	·←·→·→·	P–iQ, P–oQ	M
4	open jaws	hyper seeding	23	·→·←·←·→·	P–iQ, Q+oP	
		hyper m-bleeding	24	·←·←·→·→·	P–oQ, Q–oP	M
		hyper merger	25	·→·→·←·←·	P+iQ, Q+iP	U
5	snails	transfusion	27	·→·←·←·→·→·	P+oQ, P–oQ	M
		infusion	28	·→·→·←·←·→·	P+iQ, P–iQ	U

5.5 Beyond snails

The spiral can be further extended, but with no new basic sets of atomic interactions. For example, adding another arrow to our transfusion (see Fig. 27) results in a more complicated picture that has the atoms $\{Q$+iP, P–o$Q\}$, i.e. Q seeds P, as in Fig. 29 below.

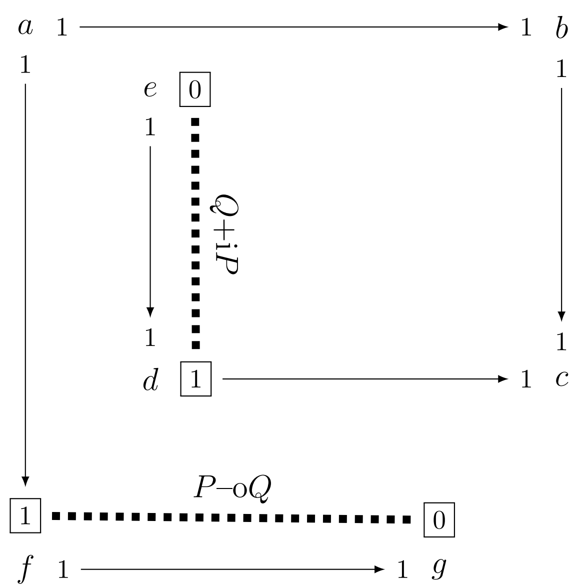


Fig. 29: Hyper snail

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 suggests that we have reached the end of the line as far as typology is concerned. Indeed, we will demonstrate rigorously below that the inventory in (43) exhausts the possibilities of two-atom molecules.

5.6 Basic and compound interactions

Any pair of maps minimally contains a two-atom interaction, as demonstrated by the discussion in the preceding subsections, and more formally below. Of course, any given pair of maps can contain more than this minimum. For example, there can be disjoint graphs instantiating different basic interactions. Suppose P is vowel deletion in hiatus (V $\longrightarrow \varnothing$ / __ V), and Q is palatalization (s $\longrightarrow$ ʃ / __ i). Then there are interactions where P feeds Q (/sai/ $\longrightarrow$ si $\rightarrow$ [ʃi]), and others where P bleeds Q (/sia/ $\longrightarrow$ [sa]). These situations form two disjoint graphs, each with the appropriate topology ($\rightarrow\!\cdot\!\rightarrow\!\cdot\!\leftarrow$ for feeding, $\leftarrow\!\cdot\!\rightarrow\!\cdot\!\rightarrow$ for bleeding). Such situations, where the same map pair P, Q instantiates two different basic interactions, have been called COMPOUND interactions in previous work ([self-citation omitted for review]), and 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 word-final apocope (P) and word-final non-apical deletion (Q), based on examples from Lardil (Hale 1973, Kavitskaya & Staroverov 2010).

(44) P: V $\longrightarrow \varnothing$ / $\sigma\sigma$ __ # (apocope)

Q: [–apical] $\longrightarrow \varnothing$ / __ # (C-deletion)

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

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

Both rules underapply because they are in what Kavitskaya & Staroverov (2010) call a *fed counterfeeding* relation. Fed counterfeeding is a compound interaction, not a basic one, as shown in Fig. 30 (recall Fig. 22). This graph illustrates two simultaneous seeding interactions arising from mutual input-provision and mutual output-removal of these two rules.

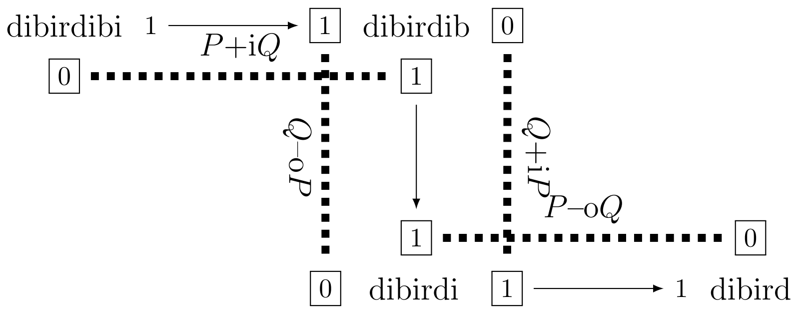


Fig. 30: Fed counterfeeding in Lardil

Because apocope applies only if its result is at least bisyllabic, and C-deletion only applies to non-apical consonants, there are some natural limits on the 'depth' of this fed counterfeeding interaction with most imagineable inputs — but, in principle, the mutual seeding can continue in stair-step fashion so long as the consonants are non-apical and the vowels are preceded by at least two syllables.

Beyond these examples, the compound interaction landscape is vast and remains to be explored.

6 Characterizing the typology

6.1 The Composite Interaction Theorem

In the preceding section we explored the typology of interaction by enumeration: we looked at various configurations of arrows (i.e. mappings), from simpler to more complex, and reached an inventory of interactions in table (43). More complex interactions, such as the two-corner structure in Figs. 22 or 30, seem to be combinations of simpler ones. In this section we attack the typology from the opposite direction: rather than building it up from examples, we derive it from a general principle that we call the COMPOSITE INTERACTION THEOREM.

Given two maps, P and Q, there are four interaction atoms in each direction, or eight atoms total. Table (43) above lists some pairs of interactions that are possible and attested, e.g. $\{P\text{+i}Q,\ P\text{+o}Q\}$ represents feeding. All interactions involve more than one atom, and some pairs don't appear on the list: for example, there is no interaction specified by $\{P\text{+i}Q,\ Q\text{–i}P\}$.

Here we demonstrate that these and other facts are a consequence of a general principle that governs the structure of interactions. First, we assign the eight atoms to two CHIRALITY CLASSES, which we call CLOCKWISE and COUNTERCLOCKWISE. The assignment is shown in (45) using bra-ket notation;[12] clockwise atoms are enclosed in $|\quad\rangle$, e.g. $|P\text{–i}Q\rangle$, $|P\text{+o}Q\rangle$;

[12] See for example `https://en.wikipedia.org/wiki/Bra-ket_notation`.

counterclockwise atoms in $\langle \quad |$, e.g. $\langle Q\text{–i}P|$, $\langle Q\text{+o}P|$. We will subsequently show that a basic interaction must contain both a clockwise and a counterclockwise element.

(45) Atom chirality assignment
  a. Clockwise: $|Q\text{–o}P\rangle$, $|P\text{+o}Q\rangle$, $|Q\text{+i}P\rangle$, $|P\text{–i}Q\rangle$
  b. Counterlockwise: $\langle P\text{+i}Q|$, $\langle Q\text{+o}P|$, $\langle P\text{–o}Q|$, $\langle Q\text{–i}P|$

The assignment to chirality classes is not based on any intrinsic property of the atoms, but on symmetries. Any atom will change to the opposite chirality class if one of the following operations is applied to it: switching 'P' and 'Q', switching '+' and '–', or switching 'i' and 'o'. In other words, if two atoms differ by only one characteristic (P and Q labels, provision vs. removal, input vs. output), they belong to opposite chirality classes. If they differ in two characteristics, they belong to the same class. If they differ in all three characteristics, they again belong to opposite classes. In fact, there are only two assignments with all three of these symmetries: the one shown in (45), and the mirror-image one with the 'clockwise' and 'counterclockwise' labels reversed.

We are now in a position to state the Composite Interaction Theorem (CIT), the general principle that governs all finite two-map graphs. The CIT is stated in two forms, a strong form and a weak form. The strong form holds of a certain well-behaved subset of all map pairs, those defined in (46). The weak form holds of all map pairs but is currently stated as a conjecture (see details in the Appendix).

(46) A map pair P, Q is LIMITED if no input-rank or output-rank in their interaction is greater than 1.

(47) **The Strong Composite Interaction Theorem (Strong CIT)**
For any limited map pair P, Q and some finite subset of Σ^*, the number of distinct clockwise and counterclockwise atoms contained in the set of mappings of P and Q is equal.

(48) **The Weak Composite Interaction Conjecture (Weak CIC)**
For any map pair and some finite subset of Σ^*, there is at least one clockwise and one counterclockwise atom contained in the set of mappings of P and Q, or else there are no atoms.

Because 'atoms' of opposite chirality are typically present in 'molecules', interactions can be symbolized by stringing together the counterclockwise and clockwise atoms and omitting the extra $|$, e.g. $\langle P\text{+i}Q|P\text{+o}Q\rangle$ for feeding.

The CIT applies only to finite subsets of Σ^*. An infinite graph that violates the CIT can be constructed by repeating the spiral in Fig. 29 to infinity; the resulting infinite spiral will have only one atomic interaction. Alternatively, a graph can consist of an infinte number of corners or other shapes, where the notion of "equal number" may be incoherent. Needless to say, this requirement of finiteness is not a substantive limitation when dealing with phonologies of natural languages.

A fuller proof sketch of the Strong CIT, along with a discussion of the conjecture, is given in the Appendix starting on p. 45. The strategy is by backwards induction. First we show that graphs containing a single corner obey the CIT. (In fact, this much is already obvious from Figs. 14–17, by inspection of the four corners shown there.) Second, we show that every finite graph can be reduced to a collection of such corner graphs by an operation that does not affect the relative number of clockwise and counterclockwise atoms.

With these remarks, we can use the (Strong) CIT to derive the observed typology. A minimal configuration that conforms to the CIT must contain one clockwise and one counterclockwise atom. Since there are four atoms of each class, there are 16 total possible opposite-chirality pairs shown in (49), some of which are the same up to names of P and Q, leaving 10 distinct molecules, all of which have already been seen in the preceding section.

(49)

cntr-wise \ c-wise	$\vert P\text{–i}Q\rangle$	$\vert Q\text{–o}P\rangle$	$\vert Q\text{+i}P\rangle$	$\vert P\text{+o}Q\rangle$
$\langle Q\text{–i}P\vert$	m-bleed	bleed	infuse	h-seed
$\langle P\text{+i}Q\vert$	infuse	seed	h-merge	feed
$\langle Q\text{+o}P\vert$	h-seed	transfuse	feed	merge
$\langle P\text{–o}Q\vert$	bleed	h-m-bleed	seed	transfuse

These molecules can also be classified by symmetry. There are four interactions in the same direction (e.g. from P to Q), four symmetical mutual interactions (one for each of the four basic types), and two asymmetrical mutual interactions.

(50)

interaction	atoms	category
feeding	$\langle P\text{+i}Q\vert P\text{+o}Q\rangle$	unidirectional
bleeding	$\langle P\text{–o}Q\vert P\text{–i}Q\rangle$	
transfusion	$\langle P\text{–o}Q\vert P\text{+o}Q\rangle$	
infusion	$\langle P\text{+i}Q\vert P\text{–i}Q\rangle$	
merger	$\langle Q\text{+o}P\vert P\text{+o}Q\rangle$	symmetrical mutual
hyper-merger	$\langle P\text{+i}Q\vert Q\text{+i}P\rangle$	
mutual bleeding	$\langle Q\text{–i}P\vert P\text{–i}Q\rangle$	
hyper-mutual-bleeding	$\langle P\text{–o}Q\vert Q\text{–o}P\rangle$	
seeding	$\langle P\text{+i}Q\vert Q\text{–o}P\rangle$	asymmetrical mutual
hyper-seeding	$\langle Q\text{–i}P\vert P\text{+o}Q\rangle$	

Additionally, the space of possible basic interactions can be understood using basic operations that can convert one into another, to which we turn next.

6.2 Relations between interactions

Interaction types are known to be related to each other through various operations. Specifically, the feeding and bleeding relations can be related by an operation called FLIPPING (Hein et al. 2014; [self-citation omitted for review]). Suppose rule P feeds rule Q, and P is of the form $a \longrightarrow b \ / \ c \_\_ d$. The flipping operation converts P to rule $\mathbf{f}(P)$, $b \longrightarrow a \ / \ c \_\_ d$. Then, $\mathbf{f}(P)$ bleeds Q.

For example, suppose rule P is word-final raising, which feeds assibilation (Q), based on examples from Finnish (Kiparsky 1973a, 1993).[13] The flipped version of P, $\mathbf{f}(P)$, is word-final lowering, which instead bleeds assibilation.

(51) P: e $\longrightarrow$ i / ___ # (raising)
$\mathbf{f}(P)$: i $\longrightarrow$ e / ___ # (lowering)
Q: t $\longrightarrow$ s / ___ i (assibilation)
✓($P \succ Q$): /vete/ $\xrightarrow{P}$ veti $\xrightarrow{Q}$ vesi 'water' (feeding)
($\mathbf{f}(P) \succ Q$): /veti/ $\xrightarrow{\mathbf{f}(P)}$ vete $\xrightarrow{Q}$ vete (bleeding)

The flipping operation on rules, as loosely defined above in terms of rule notation, is only coherent under special circumstances. For example, the debuccalization rule C $\longrightarrow$ ʔ / ___ # does not have a flipped counterpart: the notation ʔ $\longrightarrow$ C is incoherent. More generally, such a flipping operation can be defined for maps that do not contain non-vacuous neutralizing mappings.

We define flipping more carefully below. If the original map contains any non-vacuous neutralizations, flipping is undefined. Otherwise, flipping reverses all non-vacuous arrows.

(52) 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 non-vacuous mapping $P(a) = b$, $\mathbf{f}(P)(b) = a$.

With this definition in hand, it is clear why 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, $\cdot\xleftarrow{P}\cdot\xrightarrow{Q}\cdot$.

Likewise, the two-arrow and three-arrow interactions are related by another operation, called CROPPING in previous work ([self-citation omitted for review]). For example, considering the Finnish raising/assibilation case, suppose the raising rule is narrowed in scope to apply after non-sibilants: e $\longrightarrow$ i / [–strid] ___ #. The interaction of this new cropped raising, or $\mathbf{c}$(raising), with assibilation would be one of seeding, instantiated by the sequence /te/ $\longrightarrow$ ti $\longrightarrow$ [si]. The first step, raising, creates the conditions for assibilation, but the application of assibilation then destroys the conditions for raising which, in its present cropped version, 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.

[13] In the actual Finnish example there is also non-derived environment blocking of assibilation, but this is orthogonal to questions of ordering.

$$\begin{tikzcd} \text{te} \arrow[r, "P"] & \text{ti} \arrow[d, "Q"] \\ \text{se} \arrow[r, "P"] & \text{si} \end{tikzcd}$$

Fig. 31: Feeding with raising

$$\begin{tikzcd} \text{te} \arrow[r, "P"] & \text{ti} \arrow[d, "Q"] \\ & \text{si} \end{tikzcd}$$

Fig. 32: Seeding with cropped raising

Cropping relates the two-arrow operations (seeding, mutual bleeding, merger) to the three-arrow operations (feeding and bleeding). Cropping is a more loose notion than flipping. In any given case there are many cropping operations that would do the job of converting one interaction to another. For example, in the Finnish case, we could have cropped to apply raising only word-finally after stops, or only word-finally after non-continuants, 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 that "feeding can be converted to seeding by cropping" can be interpreted as 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."

With these caveats in mind, 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 the basic interactions, and they produce exactly the expected effects. This is true not only of seeding and feeding, but more generally of all basic interactions identified above, including the open jaws, infusion, and transfusion, are also related by these operations of cropping and flipping. (In fact, our example of Anti-Yokuts infusion in (42) and Fig. 28 was constructed by flipping one of the rules in Yokuts transfusion, (41) and Fig. 27.) The following picture summarizes the operations that convert various types into each other.

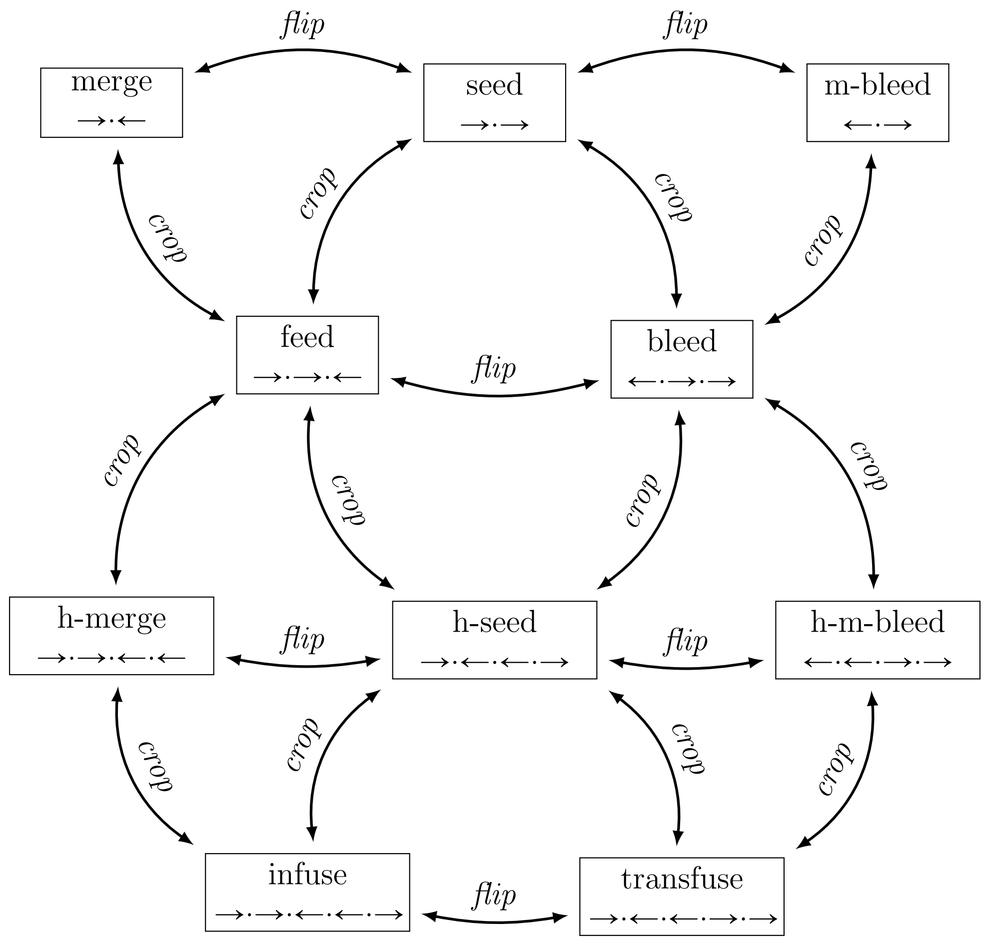


Fig. 33: Conversion operations and interaction types

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 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 +i and –o atoms.

Furthermore, the atoms +i and –o produce distinct signatures in the opaque interactions. The atom +i is responsible for derivations with the character of underapplication, where "it looks like a rule should have applied, but it did not." The atom –o 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 non-application (in the case of overapplication) or application in a different locus (in the complementary case of misapplication), with the caveat that –o atoms in on-focus interactions may produce unintuitive results (recall §4.4). These opacity properties were observed in all of the examples examined in this article, and are predicted for some interaction types where we were unable to find or construct plausible examples. One such case is hyper merger (see Fig. 25), containing the interactions P+iQ and Q+iP. If such an interaction is ever found, we expect it to display underapplication opacity in both orders.

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. While the previous literature has focused on input properties with respect to Q — how Q treats strings as inputs — we have demonstrated that to understand the space of possible interactions it is also necessary to consider output properties with respect to Q. It is this move that permits us to map out an exhaustive typology of ten basic interactions, of which only five (feeding, bleeding, seeding, transfusion, and mutual bleeding) have 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 these interactions.

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

References

Al-Mozainy, H. Q. 1981. *Vowel Alternations in a Bedouin Hijazi Arabic Dialect: Abstractness and Stress*. Austin: University of Texas PhD 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.

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

Baković, Eric. 2007. A revised typology of opaque generalisations. *Phonology* 24. 217–259. doi: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, Mass.: Wiley Blackwell 2nd edn. doi:doi:10.1002/9781444343069.ch2.

Baković, Eric. 2013. *Blocking and Complementarity in Phonological Theory*. London: Equinox.

Bethin, Christina. 1978. *Phonological Rules in the Nominative Singular and Genitive Plural of the Slavic Substantive Declension*. Champaign-Urbana, IL: PhD dissertation.

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. doi:10.1002/9781444335262.wbctp0088.

Bye, Patrik. 2002. *Virtual Phonology: Rule Sandwiching and Multiple Opacity in North Saami*: University of Tromsø PhD 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. doi:10.1075/elt.00031.cha.

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

Chomsky, Noam & Morris Halle. 1968. *The Sound Pattern of English*. 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. doi:10.1515/9783111418797-018.

Halle, Morris. 1962. Phonology in Generative Grammar. *Word* 18(1–3). 54–72. doi:10.1080/00437956.1962.11659765.

Harris, James W. 1969. *Spanish Phonology*. Cambridge, MA: MIT Press.

Harris, James W. 1999. Nasal depalatalization *no*, morphological wellformedness *sí*; the structure of Spanish word classes. In Karlos Arregi, Benjamin Bruening, Corneila Krause & Vivian Lin (eds.), *Papers on Morphology and Syntax, Cycle One*, vol. 33, 47–82. MIT Working Papers in Linguistics.

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

Hayes, Bruce & James White. 2015. Saltation and the P-map. *Phonology* 32(2). 267–302. doi: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. 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. doi: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. doi:10.1515/9783111666242.13. [Reprinted in *Explanation in Phonology*, 13–55. Dordrecht: Foris, 1982.].

Kiparsky, Paul. 1971. Historical linguistics. In William O. Dingwall (ed.), *A Survey of Linguistic Science*, 576–642. College Park: University of Maryland Linguistics Program. [Reprinted in *Explanation in Phonology*, 57–80. Dordrecht: Foris, 1982.].

Kiparsky, Paul. 1973a. How abstract is phonology? In Osamu Fujimura (ed.), *Three Dimensions of Linguistic Theory*, 5–56. Tokyo: TEC. [Reprinted in *Explanation in Phonology*, pp. 119-163, 1982.].

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. doi:10.1016/B978-0-12-325071-1.50016-9.

Kiparsky, Paul. 2000. Opacity and cyclicity. *The Linguistic Review* 17. 351–365. doi: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. doi:10.1016/j.lingua.2007.09.011.

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

Łubowicz, Anna. 2002. Derived environment effects in Optimality Theory. *Lingua* 112. 243–280. doi: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. doi:10.1002/9781444335262.wbctp0074.

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

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 PhD 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. doi:10.1017/S0952675703004408.

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

Pruitt, Kathryn. to appear. Serialism and opacity in phonological theory. *Annual Review of Linguistics*.

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 *Supplemental Proceedings of the Annual Meeting on Phonology 2015*, Washington, DC: Linguistic Society of America. doi:10.3765/amp.v3i0.3689.

Rasin, Ezer. to appear. Shifting interactions and countershifting opacity: A note on opacity in Harmonic Serialism. *Linguistic Inquiry*. doi:10.1162/ling_a_00430.

Rubach, Jerzy. 1984. *Cyclic and Lexical Phonology: The Structure of Polish*. Dordrecht: Foris.

Shaw, Patricia. 1980. *Theoretical issues in Dakota phonology and morphology*. 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. to appear. Some formal implication of deletion saltation. *Linguistic Inquiry.* doi: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`.

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. doi:10.1002/9781444335262.wbctp0124.

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

White, James. 2014. Evidence for a learning bias against saltatory phonological alternations. *Cognition* 130(1). 96–115. doi: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. doi:10.1353/lan.2017.0001.

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

Appendix: proof sketch of the CIT

We will refer to interaction structures between pairs of maps as GRAPHS below, and strings as NODES of the graph. Mappings are EDGES or ARROWS, which can begin or end at a node. Graphs considered here consist of arrows (mappings) of two distinct maps, P and Q; thus we will refer to P-arrows and Q-arrows.

(53) A graph P, Q is LIMITED if no input-rank or output-rank in any mapping is greater than 1.

Atoms are assigned to two chirality classes as follows:

(54) Atom chirality assignment
  a. Clockwise: $|Q\text{–o}P\rangle$, $|P\text{+o}Q\rangle$, $|Q\text{+i}P\rangle$, $|P\text{–i}Q\rangle$
  b. Counterclockwise: $\langle P\text{+i}Q|$, $\langle Q\text{+o}P|$, $\langle P\text{–o}Q|$, $\langle Q\text{–i}P|$

(55) **The Strong Composite Interaction Theorem (Strong CIT)**
For a limited graph P, Q and some finite subset of Σ^*, the number of distinct clockwise and counterclockwise atoms contained in the set of mappings of P and Q is equal.

(56) **The Weak Composite Interaction Conjecture (Weak CIC)**
For any limited graph P, Q and some finite subset of Σ^*, there is at least one clockwise and one counterclockwise atom contained in the set of mappings of P and Q, or else there are no atoms.

Proof sketch of the Strong CIT

The proof sketch of the CIT (55) will proceed as follows. First we will show that certain simple graphs (graphs containing only one "active" node) satisfy the CIT. We will then show that any finite graph can be reduced to one of those simple graphs without affecting the chiral parity of the atomic interactions.

First, some definitions.

(57) a. A node a is an ACTIVE NODE if there are both P and Q arrows that begin or end at a.
  b. A graph is a SINGLE-NODE GRAPH if it contains one and only one active node.
  c. A graph is a CORNER if it is a single-node graph such that if one arrow begins at its active node and another ends at its active node, these two arrows belong to different maps.

Note that all corners are single-node graphs, but not vice versa. For example, the following graph is a single-node graph that is not a corner: $P = \{a \mapsto b\}, Q = \{c \mapsto b, b \mapsto d\}$. Note also that every single-node graph is a union of distinctly shaped corners.

Lemma 1: Every corner satisfies the CIT.

Proof sketch. In each corner, the (one and only) active node is adjacent to either the input or the output of the P-arrow, and to either the input or the output of the Q-arrow.

There are thus four types of corners. The CIT for each of these four types can be verified by inspecting Figs. 14–17 on p. 25. □

Lemma 2: Let K and L be two distinctly shaped corners with the same active node such that their union is a map. Then the union of the mappings of K and L contains the union of atomic interactions of K and L.

Proof sketch. Each atom present in a corner is due to an input-rank or output-rank of the corner's active node being equal to 1 (see Figs. 14–17). Each of these ranks of 1 results from a presence of an arrow originating or ending at the active node, and does not depend on the presence of any other arrows at the corner. A P-input-rank of 1 results from a P-arrow originating at that node, a P-output-rank of 1 results from a P-arrow ending at that node, and *mutatis mutandis* for Q-arrows. The union operation can only add but not remove such arrows, and thus is unable to reduce an input- or output-rank from 1 to 0. Thus, all the same atomic interactions are present in the union as in the individual corners.

Next, we show that CIT holds for single-node graphs.

Lemma 3: Every single-node graph satisfies the CIT.

Proof sketch. Every single-node graph is a union of corners. By Lemma 1, each such corner contains a pair of atomic interactions of opposite chirality. Inspection of Figs. 14–17 shows that the four corners' sets of interactions are non-overlapping. By Lemma 2, the union of such corners contains the union of their interactions, and therefore satisfies the CIT. □

Now we turn to the main line of the proof of the Strong CIT: the demonstration that any graph can be reduced to a single-node graph without affecting what might be called the CHIRAL PARITY, or the relative number of clockwise and counterclockwise atoms.

(58) **Chiral parity**
Given two maps P, Q, the chiral parity of the graph of their mappings is the difference between the number of clockwise atoms and the number of counterclockwise atoms in that graph.

Note that the chiral parity of all single-node graphs is 0, and the CIT is equivalent to the claim that the chiral parity of all finite graphs is 0.

Let us now introduce an operation on graphs called BREAKING. Let $(a, b) \in P$ be a P-arrow. Let a', b' be strings not in Σ^*; they can be constructed, for example, by appending to a and b some symbol not in Σ.

The operation of breaking, applied to a graph and the mapping (a, b), removes that mapping and introduces two new mappings, (a, a') and (b', b). In other words, breaking results in a map $Br(P, (a, b)) = P \setminus \{(a, b)\} \cup \{(a, a'), (b', b)\}$, visually shown in (59) below.

(59) Before breaking: $a \longrightarrow b$

After breaking: $a \to a' \quad b' \to b$

Note, crucially, that the only way that the nodes a', b' participate in the graph $Br(P, (a, b))$ is as outputs and inputs of a, b, respectively. Because they do not belong to the set Σ^*, they are not in the input or output set of any other map Q. The node b' is in the input set of P

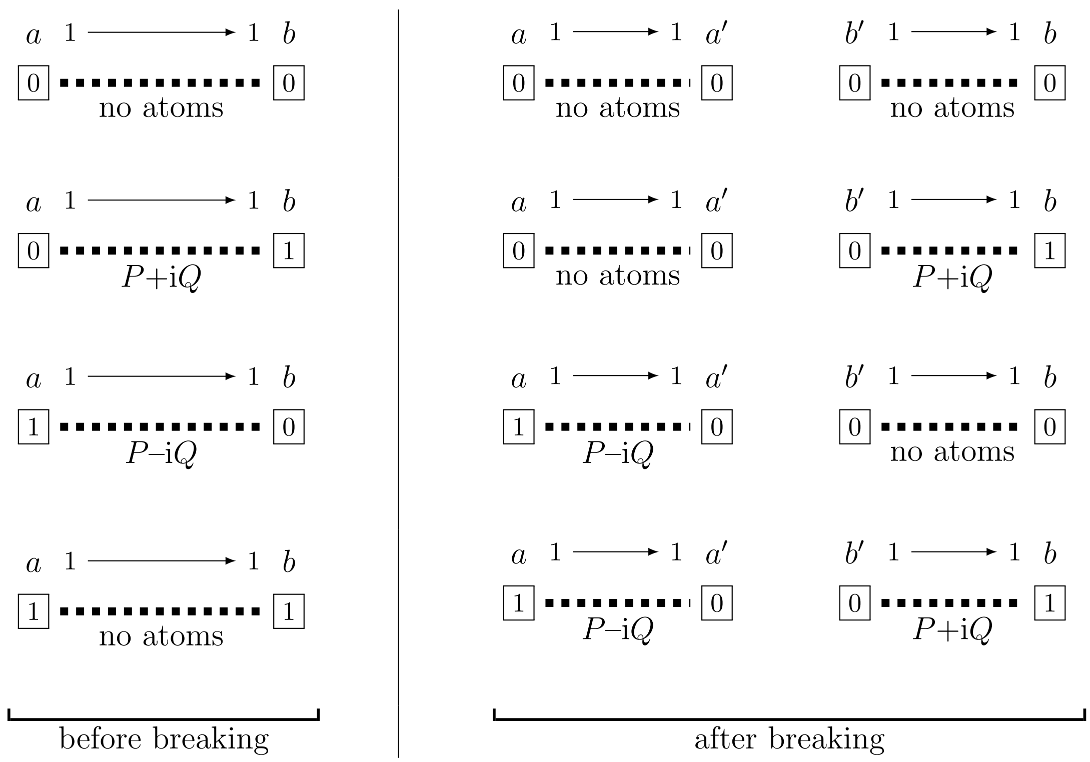
only by virtue of the pair (b', b), and the node a' is in the output set of P only by virtue of the pair (a, a').

Clearly, any finite graph can be reduced via successive breakings to a collection of single-node graphs. With the following theorem, the proof sketch of the CIT will be complete.

Theorem: Breaking does not affect the chiral parity of a finite graph.

Proof sketch. Consider a graph with maps P, Q, and a P-arrow (a, b) of that graph. That arrow may or may not contain some atoms, depending on the comparison of ranks at the edges of that arrow. The four possibilities for the comparison between input-ranks for Q are shown below. There are the same four possibilities for the comparison of output-ranks of Q. Note that the nodes a', b' have Q-input-ranks of 0, because they are not in the input set of Q by assumption.

For each of these situations, we examine the fate of the arrow's atoms upon breaking.

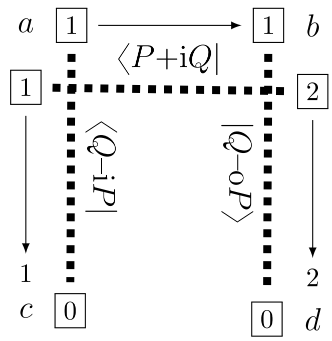


In the first three cases, the breaking operation does not change the atom composition of the map, preserving the original atom if there had been one. In the fourth case, breaking introduces two atoms which are necessarily of opposite chirality, differing only in the + vs. – sign. Likewise, the same four possibilities hold for the comparison of output-ranks across the same P-arrow, and the same outcome of breaking: it either does not change the composition of atoms, or adds two atoms of opposite chirality. Breaking a P-arrow thus preserves the chiral parity of a graph. *Mutatis mutandis* for breaking a Q-arrow.

Because any finite graph can be reduced via a series of breaking operations to a collection of single-node graphs, its original chiral parity is equal to the chiral parity of that collection. By Lemma 3, each single node graph's chiral parity is 0, and thus the original graph's chiral parity is also necessarily 0. This completes the proof. □

The Weak CIC

The following configuration is an example of a graph containing a non-limited map that obeys the Weak CIC but not the Strong CIT, containing one clockwise atom and two counter-clockwise atoms.



We have not been able to discover or construct phonologically plausible examples of such configurations.

The Weak CIC is stated as a conjecture at present. No counterexample is known to us, but prospects of a proof also remain elusive.

However, in practice, phonologically reasonable graphs have properties that guarantee that they will obey the weak version of the theorem. In particular, one sufficient condition for the satisfaction of the weak CIC is that the graph P, Q contain a disjoint subgraph that is limited, and therefore satisfies the Strong CIT. For maps instantiating phonological processes, this property is nearly guaranteed: it is difficult to imagine a rule that would apply more than once to all or nearly all of its inputs.

Another sufficient condition for the satisfaction of the Weak CIC is the existence of an active node with the following property: every rank (input-rank or output-rank) at that node is greater or equal than the corresponding rank across a P (or Q) arrow. The reason this condition is sufficient is that the operation of breaking can be applied to all arrows originating or ending in such a node without loss or gain of atoms at that node, resulting in a single active node which, by Lemma 3, satisfies the Strong CIT (a more detailed demonstration of this claim is omitted here for reasons of space).