A set-theoretic typology of phonological map interaction*

Eric Baković, UC San Diego
with *Lev Blumenfeld, Carleton University*

Background assumptions

(1) Theories of generative phonology assume that, in general:
  a. morphemes have unitary underlying representations, and
  b. systematic variation in the surface pronunciation of morphemes in different morphological contexts results from the application of a complex, context-sensitive transformation — a *phonological grammar* — to those underlying representations.

(2) A phonological grammar is thus a *complex map* from underlying to surface representations.
  a. Theories differ on the details of what the phonological grammar is made of (rules and ordering/composition (Chomsky & Halle 1968, Kaplan & Kay 1994), constraints and ranking/lenient composition (Prince & Smolensky 1993, Karttunen 1998), etc.), but
  b. it is commonly assumed that it can be broken down into a set of simpler maps — *individual phonological processes* — that make particular changes in particular contexts.

Our question

(3) What are the possible ways in which the individual processes that constitute a phonological grammar can non-trivially affect each other's mappings — can *interact* with one another?

(4) Pairs of processes P and Q interact iff there is *overlap* between (Baković 2012, 2013)
  a. P's and Q's *inputs* (= sets of substrings that match their structural descriptions),
  b. P's and Q's *outputs* (= sets of substrings that match their structural changes), and/or
  c. the input of one process and the output of the other.

(5) Save for some early work on e.g. formal inclusion relations between these components of phonological rules (Anderson 1969, Kiparsky 1973b, Koutsoudas et al. 1974, among others), we believe that the formal consequences of (4) have not been adequately explored.

Our contribution

(6) This work begins to fill this gap in our understanding of phonological grammars by fleshing out a complete, theory-independent typology of possible pairwise process interaction types.
  ☞ 'theory-independent' = without prior commitment to whether an 'individual phonological process' corresponds to a(n ordered) phonological rule, a $\mathbb{M} \gg \mathbb{F}$ ranking, etc.

(7) Formal analysis reveals this typology to be relatively small, consisting of just six process interaction types. We present this typology and our formal analysis of it in detail.

(8) We also propose formal characterizations of 'underapplication' and 'overapplication', familiar from work on phonological opacity, which cross-classify process interaction types.

*Both authors give thanks to Jeff Heinz, Jason Riggle, and Eric Meinhardt, and to other members of audiences at the LSA, Carleton, San Diego, and MOT. Today's presenter gives special thanks to his co-author.

Basic notions: inputs, outputs, maps

(9) A phonological grammar is a function from a set of input strings I to a set of output strings O. These sets are drawn from the set of all possible strings, Σ^*: $\{I \subseteq \Sigma^*\} \mapsto \{O \subseteq \Sigma^*\}$.

a. Entire phonological grammars, individual phonological processes, or subsets of phonological processes within a phonological grammar are all functions in this sense.

b. E.g., the function defined by a process of intervocalic voicing contains input-output pairs like {(apa, aba), (ata, ada), (aka, aga), (pa, pa), (ada, ada), (ana, ana), ...}.

c. We are specifically interested in sets of non-identical i-o pairs. We ignore *vacuous application*, as exhibited by the pairs {(pa, pa), (ada, ada), (ana, ana)} in this case.

d. We will use the term *map* to refer to sets of non-vacuous i-o pairs. The map for intervocalic voicing thus only contains pairs like {(apa, aba), (ata, ada), (aka, aga), ...}.

A *map* P defines a set of non-vacuous input-output string pairs.
The set of all first members of P's i-o pairs is P's *input* (= $I(P)$).
The set of all second members of P's i-o pairs is P's *output* (= $O(P)$).

(10) Some further details that we won't dwell on here:

a. Each i-o pair must also be equipped with a *correspondence relation* between their respective representational elements (segments, features, ...). This is glossed over here.

b. Deletion (and insertion) maps complicate the picture, in large part because they disrupt the *successor relation* between corresponding strings (Baković & Blumenfeld 2017). But we cannot completely side-step such maps, and will call attention to them as they arise.

c. The map defined by intervocalic voicing technically contains (apa, aba) but not e.g. (kapa, kaba) because *k* is irrelevant to the map's *target*. Establishing relevant relations between map targets requires careful formalization; see Baković & Blumenfeld (2017).

d. We only consider individual processes *qua* maps P such that $I(P) \cap O(P) = \varnothing$. This restriction is distinct from the requirement that a map may contain only non-vacuous i-o pairs (9d): $\{(x,y),(y,z)\}$ contains only non-vacuous i-o pairs, but $I \cap O = \{y\}$.[1]

Feeding and bleeding

(11) We can now formalize the classical rule interaction notions of *feeding* and *bleeding* (Kiparsky 1968) in terms of what we call *input provision* and *input removal* relations between maps.

(12) **Feeding = i-provision.** Two maps P and Q are in a *feeding relationship*, *P potentially feeding Q*, if there exists an i-o pair $(x,y) \in P$ such that $x \notin I(Q)$ and $y \in I(Q)$.

↳ In such a case, we say that (x,y) is a *p-feeding pair* and that P *i(nput)-provides* Q.

☞ If $P > Q$, then P FEEDS Q; if $Q > P$, then P COUNTERFEEDS Q.

(13) **Bleeding = i-removal.** Two maps P and Q are in a *bleeding relationship*, *P potentially bleeding Q*, if there exists an i-o pair $(x,y) \in P$ such that $x \in I(Q)$ and $y \notin I(Q)$.

↳ In such a case, we say that (x,y) is a *p-bleeding pair* and that P *i(nput)-removes* Q.

☞ If $P > Q$, then P BLEEDS Q; if $Q > P$, then P COUNTERBLEEDS Q.

[1] In other words, maps do not apply to their own outputs, in the way made clear by Kaplan & Kay (1994: 346ff) in their demonstration that a set of ordered rules respecting this requirement corresponds to a regular relation.

Examples of p-feeding/i-provision

(14) Feeding, based on Turkish (Underhill 1976)

a. P: [−son, −cont] ⟶ [−voi] / — $]_\sigma$ (devoice stops syllable-finally)
Q: [−son] ⟶ [−voi] / [−voi] — (devoice obstruents after voiceless consonants)

b. (rengden, renkden) $\in P$ is a p-feeding pair: rengden $\notin I(Q)$, renkden $\in I(Q)$.

☞ $P > Q$, so P feeds Q.

	/reng+den/	(*feeding*)		/reng+den/	(*counterfeeding*)
P	renkden		Q	—	
Q	renkten		P	renkden	
	[renkten]	‘color (ABL)’	*	[renkden]	

(15) Counterfeeding, based on Bedouin Arabic (Al-Mozainy 1981, McCarthy 1999)

a. P: [−cons] ⟶ [+syll] / C — # (vocalize word-final glides in clusters)
Q: [+low] ⟶ [+high] / — CV (raise low vowels in open syllables)

b. (badw, badu) $\in P$ is a p-feeding pair: badw $\notin I(Q)$, badu $\in I(Q)$.

☞ $Q > P$, so P counterfeeds Q.

	/badw/	(*counterfeeding*)		/badw/	(*feeding*)
Q	—		P	badu	
P	badu		Q	bidu	
	[badu]	‘Bedouin’	*	[bidu]	

(16) Bleeding, based on Lamba (Kenstowicz & Kisseberth 1979)

a. P: V ⟶ [−high] / [−high, −low] C_0 — (lower vowels after mid vowels)
Q: s ⟶ ʃ / — [+high, −back] (palatalize *s* before *i*)

b. (kosila,kosela) $\in P$ is a p-bleeding pair: kosila $\in I(Q)$, kosela $\notin I(Q)$.

☞ $P > Q$, so P bleeds Q.

	/kos+il+a/	(*bleeding*)		/kos+il+a/	(*counterbleeding*)
P	kosela		Q	koʃila	
Q	—		P	koʃela	
	[kosela]	‘be strong (APP)’	*	[koʃela]	

(17) Counterbleeding, based on Polish (Kenstowicz & Kisseberth 1979)

a. P: [−son] ⟶ [−voi] / — # (devoice obstruents word-finally)
Q: o ⟶ u / — [−nas, +voi] (raise *o* before voiced non-nasals)

b. (ʒwob,ʒwop) $\in P$ is a p-bleeding pair: ʒwob $\in I(Q)$, ʒwop $\notin I(Q)$.

☞ $Q > P$, so P counterbleeds Q.

	/ʒwob/	(*counterbleeding*)		/ʒwob/	(*bleeding*)
Q	ʒwub		P	ʒwop	
P	ʒwup		Q	—	
	[ʒwup]	‘crib’	*	[ʒwop]	

Beyond inputs: fleshing out the typology

(18) Outputs get some love, too...

a. P *i-provides* Q iff there exists an i-o pair $(x, y) \in P$ s.t. $x \notin I(Q)$ and $y \in I(Q)$.
   $\Rightarrow (x, y) \in P$ is an *i-provision pair* for Q.

b. P *i-removes* Q iff there exists an i-o pair $(x, y) \in P$ s.t. $x \in I(Q)$ and $y \notin I(Q)$.
   $\Rightarrow (x, y) \in P$ is an *i-removal pair* for Q.

c. P *o-provides* Q iff there exists an i-o pair $(x, y) \in P$ s.t. $x \notin O(Q)$ and $y \in O(Q)$.
   $\Rightarrow (x, y) \in P$ is an *o-provision pair* for Q.

d. P *o-removes* Q iff there exists an i-o pair $(x, y) \in P$ s.t. $x \in O(Q)$ and $y \notin O(Q)$.
   $\Rightarrow (x, y) \in P$ is an *o-removal pair* for Q.

(19) Classical p-feeding involves o-provision in addition to i-provision.

a. Turkish feeding (14)
   P: [−son, −cont] ⟶ [−voi] / — $]_\sigma$
   Q: [−son] ⟶ [−voi] / [−voi] —

b. Bedouin Arabic counterfeeding (15)
   P: [−cons] ⟶ [+syll] / C — #
   Q: [+low] ⟶ [+high] / — CV

c. In each case, there is both an i-provision pair and an o-provision pair:
   i. Turkish i-provision: (rengden, renkden) $\in P$; rengden $\notin I(Q)$, renkden $\in I(Q)$.
      Bedouin Arabic i-provision: (badw, badu) $\in P$; badw $\notin I(Q)$, badu $\in I(Q)$.
   ii. Turkish o-provision: (rengten, renkten) $\in P$; rengten $\notin O(Q)$, renkten $\in O(Q)$.
      Bedouin Arabic o-provision: (bidw, bidu) $\in P$; bidw $\notin O(Q)$, bidu $\in O(Q)$.

(20) Classical p-bleeding involves o-removal in addition to i-removal.

a. Lamba bleeding (16)
   P: V ⟶ [−high] / [−high, −low] C_0 —
   Q: s ⟶ ʃ / — [+high, −back]

b. Polish counterbleeding (17)
   P: [−son] ⟶ [−voi] / — #
   Q: o ⟶ u / — [−nas, +voi]

c. In each case, there is both an i-removal pair and an o-removal pair:
   i. Lamba i-removal: (kosila,kosela) $\in P$; kosila $\in I(Q)$, kosela $\notin I(Q)$.
      Polish i-removal: (ʒwob, ʒwop) $\in P$; ʒwob $\in I(Q)$, ʒwop $\notin I(Q)$.
   ii. Lamba o-removal: (koʃila, koʃela) $\in P$; koʃila $\in O(Q)$, koʃela $\notin O(Q)$.
      Polish o-removal: (ʒwub, ʒwup) $\in P$; ʒwub $\in O(Q)$, ʒwup $\notin O(Q)$.

(21) The following "bubble diagrams" illustrate map interaction.[2]

Figure 1: ***Feeding*** *as full provision*

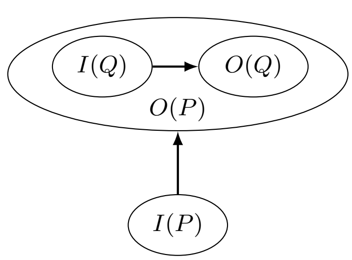
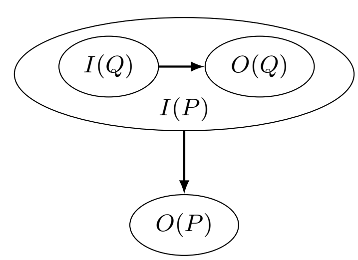


Figure 2: ***Bleeding*** *as complete removal*



[2]These diagrams are meant as convenient visual aids, not as formal objects — note, for example, that both $I(Q)$ and $O(Q)$ are inexactly represented as *proper subsets* of $O(P)$ in Figure 1 and of $I(P)$ in Figure 2.

Formal analysis

(22) $(3 \times 3 \times 3 \times 3) - 1 = 80$ logically possible non-trivial combinations!

- P may i-provide or i-remove Q (= 3 possibilities, including neither).
- P may o-provide or o-remove Q (= 3 possibilities, including neither).
- Q may i-provide or i-remove P (= 3 possibilities, including neither).
- Q may o-provide or o-remove P (= 3 possibilities, including neither).

Of these 80 logical possibilities, how many and which are *basic* (= not further reducible)?

Claimed result: **there are six (not seven!) basic interactions.**[3]

(23)

	interaction type	*formal characterization*
1.	feeding	P i-provides & o-provides Q
2.	bleeding	P i-removes & o-removes Q
3.	self-destructive feeding	P i-provides Q, Q o-removes P
4.	mutual feeding	mutual i-provision & o-removal
5.	merger	mutual o-provision
6.	divergent mutual bleeding	mutual i-removal

(24) We now justify our claim that these six interactions exhaust the logically possible ones.

a. We first assume that P either i-provides, i-removes, o-provides, or o-removes Q, and from there determine the other possible relations that must hold between P and Q.

b. **Caveat**: these interaction types do not apply directly to P and Q, but to sets of i-o pairs in P and Q. Thus it is possible for P and Q to display more than one interaction.[4]

Case 1 of 4: P i-provides Q

(25) Let $(x, y) \in P$ be the i-provision pair. Because $y \in I(Q)$, there exists $(y, z) \in Q$.
There are three possibilities with respect to z's place in P:

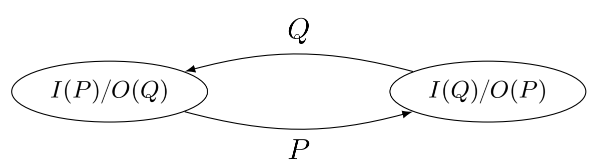
a. $z \in I(P)$ (and thus $\notin O(P)$) **mutual feeding**

b. $z \in O(P)$ (and thus $\notin I(P)$) **'regular' feeding**

c. $z \notin I(P), z \notin O(P)$ **self-destructive feeding**

(26) **Subcase (25a)**: $z \in I(P)$: P and Q mutually i-provide & o-remove.[5]

Figure 3: ***Mutual feeding*** *as mutual i-provision & o-removal*



[3]The previously-advertised seventh interaction, *convergent mutual bleeding*, appears to be a 'compound' case involving *merger* (23-5) and *divergent mutual bleeding* (23-6), and so is not basic in the sense we mean here.

[4]Baković & Blumenfeld (2017) begin to explore how maps that describe *SPE*-style rules may exhibit such 'compound' relations; see the last page of this handout for a sneak peek of a couple of examples.

[5]The given conditions $(x, y) \in P$, $(y, z) \in Q$, and $z \in I(P)$ define mutual i-provision, but the requirement (10d) that $I(Q) \cap O(Q) = \varnothing$ further entails that $z \notin O(P)$ and $y \notin O(Q)$, therefore also defining mutual o-removal.

(27) Mutual feeding example, based on Nootka (Kenstowicz & Kisseberth 1979)

a. P: [+dor] $\longrightarrow$ [+round] / [+round] — (labialize dorsals after round vowels)
Q: [+dor] $\longrightarrow$ [−round] / — $]_\sigma$ (delabialize dorsals syllable-finally)

b. (moq, moq^w) $\in P$ is an i-provision pair: moq $\notin I(Q)$, moq^w $\in I(Q)$.
(moq, moq^w) $\in P$ is also an o-removal pair: moq $\in O(Q)$, moq^w $\notin I(Q)$.

c. (moq^w, moq) $\in Q$ is an i-provision pair: moq^w $\notin I(P)$, moq $\in I(P)$.
(moq^w, moq) $\in Q$ is also an o-removal pair: moq^w $\in O(P)$, moq $\notin O(P)$.

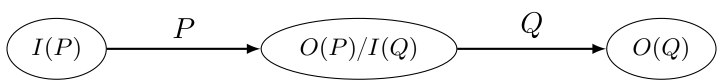
☞ $P > Q$, so delabialization prevails.

	/moq/	(*feeding*)		/moq^w/	(*also feeding*)
P	moq^w		Q	moq	
Q	moq		P	moq^w	
	[moq]	‘ten on top’	*	[moq^w]	

(28) **Subcase (25b)**: $z \in O(P)$: P i-provides & o-provides Q.
↳ This is ‘regular’ feeding; see (14), (15), (19), and Figure 1 above.

(29) **Subcase (25c)**: $z \notin I(P), z \notin O(P)$: P i-provides Q & Q o-removes P.

Figure 4: ***Self-destructive feeding*** *as i-provision & o-removal*



(30) Self-destructive feeding with deletion, based on Turkish (Kenstowicz & Kisseberth 1979)

a. P: [+cont] $\longrightarrow \varnothing$ / C — (delete continuants after consonants)
Q: k $\longrightarrow \varnothing$ / V — +V (delete *k* between vowels)

b. (ajaksɯ, ajakɯ) $\in P$ is an i-provision pair: ajaksɯ $\notin I(Q)$, ajakɯ $\in I(Q)$.

c. (ajakɯ, ajaɯ) $\in Q$ is an o-removal pair: ajakɯ $\in O(P)$, ajaɯ $\notin O(P)$.

☞ $P > Q$, so self-destructive feeding.

	/ajak+sɯ/	(*self-destructive feeding*)		/ajak+sɯ/	(*counterfeeding*)
P	ajakɯ		Q	—	
Q	ajaɯ		P	ajakɯ	
	[ajaɯ]	‘foot (POSS)’	*	[ajakɯ]	

(31) Hypothetical example of self-destructive feeding without deletion, based on Turkish

a. P: [−son, −cont] $\longrightarrow$ [−voi] / — [−son, +voi] (devoice stops before voiced obstruents)
Q: [−son] $\longrightarrow$ [−voi] / [−voi] — (devoice obstruents after voiceless consonants)

b. (rengden, renkden) $\in P$ is an i-provision pair: rengden $\notin I(Q)$, renkden $\in I(Q)$.

c. (renkden, renkten) $\in Q$ is (now) an o-removal pair: renkden $\in O(P)$, renkten $\notin O(P)$.

	/rengden/	(*self-destructive feeding*)		/rengden/	(*counterfeeding*)
P	renkden		Q	—	
Q	renkten		P	renkden	
	[renkten]		*	[renkden]	

(32) Different example with deletion, based on Lomongo (Kenstowicz & Kisseberth 1979)[6]

a. P: [−low] ⟶ [−syll] / — V (glide non-low vowels before vowels)

Q: [−son, +voi] ⟶ ∅/ V — V (delete voiced obstruents between vowels)

b. (obina, oina) $\in P$ is an i-provision pair: obisa $\notin I(Q)$, oisa $\in I(Q)$.

c. (oina, wina) $\in Q$ is an o-removal pair: oisa $\in O(P)$, wisa $\notin O(P)$.

☞ $Q > P$, so counterfeeding.

	/o+bina/	(*counterfeeding*)		/o+bina/	(*self-destructive feeding*)
Q	—		P	oina	
P	oina		Q	wina	
	[oina]	'you (SG) dance'	*	[wisa]	

Case 2 of 4: P i-removes Q

(33) Let $(x, y) \in P$ be the i-removal pair. Because $x \in I(Q)$, there exists $(x, z) \in Q$.
There are three possibilities with respect to z's place in P:

a. $z \in I(P)$ **'regular' bleeding**

b. $z \in O(P)$ **convergent mutual bleeding**

c. $z \notin I(P), z \notin O(P)$ **divergent mutual bleeding**

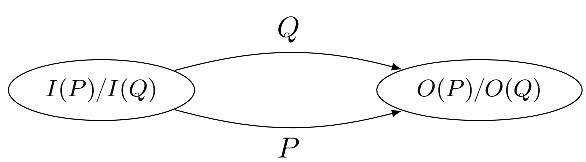
(34) **Subcase (33a)**: $z \in I(P)$: P i-removes & o-removes Q.

↳ This is 'regular' bleeding; see (16), (17), (20), and Figure 2 above.

(35) **Subcase (33b)**: $z \in O(P)$: P and Q mutually i-remove & Q o-provides P.

↳ Only three-quarters of the way toward convergent mutual bleeding, since we don't know whether P also o-provides Q. But let's play along, shall we?

Figure 5: ***Convergent mutual bleeding*** *as mutual i-removal & o-provision*



(36) Possible example of convergent mutual bleeding, based on Mushunguli (Hout 2012)[7]

P: V ⟶ [+long] / — σ# (lengthen penultimate vowels)

Q: [−high, −low]$_1$ μ_2 ⟶ [+long]$_{1,2}$ (lengthen compensatorily)

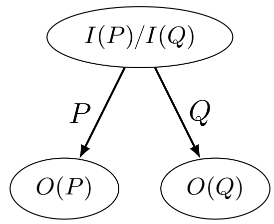
	/ka+iva/	(*bleeding*)		/ka+iva/	(*also bleeding*)
	ke$^\mu$va	(*coalescence*)		ke$^\mu$va	(*coalescence*)
P	keːva		Q	keːva	
Q	—		P	—	
	[keːva]	's/he heard'		[keːva]	

[6]Given the $Q > P$ counterfeeding order in the analysis of this example, it had not been previously identified as a case of self-destructive feeding (under the $P > Q$ order). Thanks to Lev Blumenfeld for discussion leading to this.

[7]Thanks to Kati Hout for pointing out the relevance of this example here.

(37) **Subcase (33c)**: $z \notin I(P), z \notin O(P)$: P and Q mutually i-remove.

Figure 6: ***Divergent mutual bleeding*** *as mutual i-removal*



(38) Example of divergent mutual bleeding, based on two German dialects (Kiparsky 1971)

a. P: [−son] ⟶ [−voi] / — # (devoice obstruents word-finally)
Q: g ⟶ ∅ / [+nas] — (delete *g* after nasals)

b. (laŋg, laŋk) ∈ P is an i-removal pair: laŋg ∈ $I(Q)$, laŋk ∈ $I(Q)$.

c. (laŋg, laŋ) ∈ Q is an i-removal pair: laŋg ∈ $I(P)$, laŋ ∉ $I(P)$.

	/laŋg/	(*bleeding*)		/laŋg/	(*also bleeding*)
P	laŋk		Q	laŋ	
Q	—		P	—	
	[laŋk]	'long (M.)'		[laŋ]	'long (M.)'

Case 3 of 4: P o-provides Q

(39) Let $(x, y) \in P$ be the o-provision pair. Because $y \in O(Q)$, there exists $(z, y) \in Q$. There are three possibilities with respect to z's place in P:

a. $z \in I(P)$ **convergent mutual bleeding**

b. $z \in O(P)$ **'regular' feeding**

c. $z \notin I(P), z \notin O(P)$ **merger**

(40) **Subcase (39a)**: $z \in I(P)$: P and Q mutually o-provide & Q i-removes P.

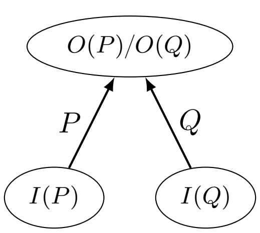
↳ This goes the other three-quarters of the way toward convergent mutual bleeding (recall subcase (33b), (36), and Figure 5 above). Lev and I are now trying to figure this out.

(41) **Subcase (39b)**: $z \in O(P)$: P i-provides & o-provides Q.

↳ This is 'regular' feeding again (recall (14), (15), (19), (25b), and Figure 1 above).

(42) **Subcase (39c)**: $z \notin I(P), z \notin O(P)$: P and Q mutually o-provide.

Figure 7: ***Merger*** *as mutual o-provision*



(43) Plausible-ish examples of merger (ensuring distinctness from convergent mutual bleeding):

	P	Q	$\in P$	$\in Q$
a.	[−cont] ⟶ [−voi]	[−voi] ⟶ [−cont]	(d, t)	(s, t)
b.	[−cont] ⟶ [−nas]	[−cont] ⟶ [+voi]	(n, d)	(t, d)
c.	σσ́σ# ⟶ σ́σσ#	#σ́CVC# ⟶ #σ́CV.Cə#	(atánə, átanə)	(átan, átanə)

Case 4 of 4: P o-removes Q

(44) Let $(x, y) \in P$ be the o-removal pair. Because $x \in O(Q)$, there exists $(z, x) \in Q$. There are three possibilities with respect to z's place in P:

a. $z \in I(P)$ **'regular' bleeding**

b. $z \in O(P)$ **mutual feeding**

c. $z \notin I(P), z \notin O(P)$ **self-destructive feeding**

(45) **Subcase (44a)**: $z \in I(P)$: P i-removes & o-removes Q.
↳ This is 'regular' bleeding again (recall 16), (17), (20), (33a), and Figure 2 above).

(46) **Subcase (44b)**: $z \in O(P)$: P and Q mutually i-provide & o-remove.
↳ This is mutual feeding again (recall (27), (25a), and Figure 3 above).

(47) **Subcase (44c)**: $z \notin I(P), z \notin O(P)$: P i-provides Q & Q o-removes P.
↳ This is self-destructive feeding again (recall (30), (31), (32), (25c), and Figure 4 above).

Summary of bubble diagrams

Figure 1: ***Feeding*** *as full provision*

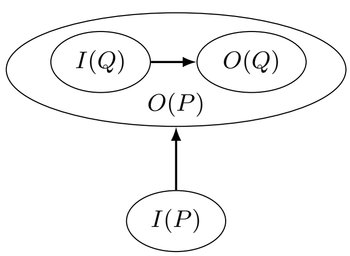


Figure 2: ***Bleeding*** *as complete removal*

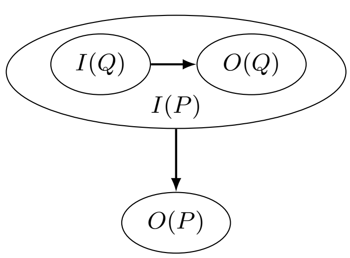


Figure 3: ***Mutual feeding*** *as mutual i-provision & o-removal*

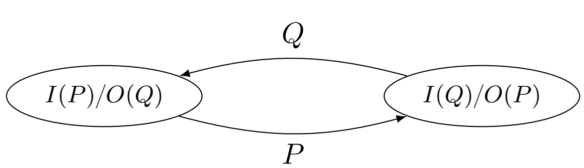


Figure 5: ***Convergent mutual bleeding*** *as mutual i-removal and o-provision*

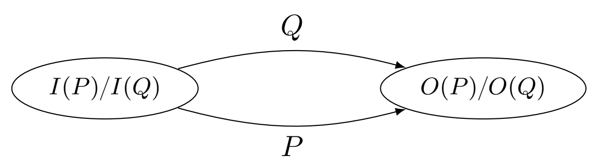


Figure 6: ***Divergent mutual bleeding*** *as mutual i-removal*

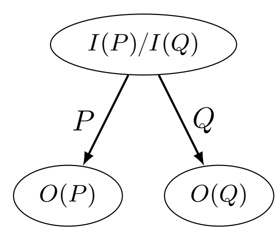


Figure 7: ***Merger*** *as mutual o-provision*

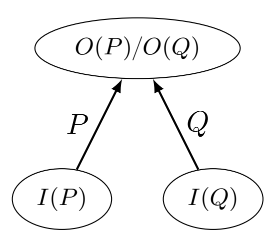
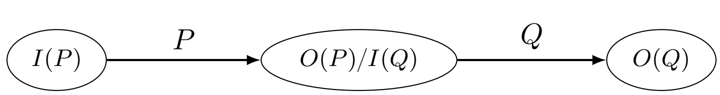


Figure 4: ***Self-destructive feeding*** *as i-provision and o-removal*



Underapplication and overapplication

(48) Opacity is classically understood as underapplication or overapplication of processes, attributed to counterfeeding or counterbleeding, respectively (but see Baković 2007).

a. **Counterfeeding *qua* underapplication**: it looks like Q should've applied, but didn't.

↳ e.g. in Bedouin Arabic, /badw/ ⟶ [badu] doesn't undergo open-syllable raising.

b. **Counterbleeding *qua* overapplication**: it looks like Q shouldn't've applied, but did.

↳ e.g. in Polish, /ʒwob/ ⟶ [ʒwup] undergoes raising, but before a [−voi] sound.

(49) Two ways of understanding under-/overapplication:

a. **Classically**. As relations between a process and the ultimate output of the phonological grammar: are there SRs that contradict Q? (Kiparsky 1971, 1973a)

b. **Here**. As relations between pairs of processes: is there a process P that in some way masks the application of Q? (Logically distinct from (49a); see Baković 2011: §3.4.)[8]

(50) Process-and-grammar opacity: a process Q of the form $a \longrightarrow b \ / \ c \_\_ d$

a. **underapplies** if there are surface sequences [cad].

b. **overapplies** if there are surface [b]s derived from /a/ by Q not in context [$c \_\_ d$].

(51) Pair-of-processes opacity: a process P of the form $a \longrightarrow b \ / \ c \_\_ d$

a. **underapplies** to the extent there are sequences [cad] at some point in the derivation to which Q does not apply (because another process P creates those sequences).

b. **overapplies** if there are [b]s derived from /a/ by Q not in context [$c \_\_ d$] at some point in the derivation later than Q (because another process P affects those contexts).

(52) First approach to set-theoretic conditions on pair-of-processes under-/overapplication:

a. Q underapplies if P i-provides Q and $(P \circ Q) \cap P \neq \varnothing$.

b. Q overapplies if P o-removes Q and $(P \circ Q) \cap Q = \varnothing$ and $(P \circ Q) \cap P \neq \varnothing$.

Summary comments

(53) The apparent formal basis of classical feeding and bleeding — input-provision and input-removal, respectively — are not sufficient to characterize possible map interactions.

↳ The full typology also requires output-provision and output-removal.

(54) There are six predicted basic interaction types: feeding, bleeding, self-destructive feeding, mutual feeding, divergent mutual bleeding, and merger.

↳ Whether one or both of the $\frac{3}{4}$-of-convergent-mutual-bleeding cases count as 'basic' or as compounds (of divergent mutual bleeding and merger) remains to be seen.

(55) Under-/overapplication are given formal characterizations in terms of our primitives.

↳ The primitives ({i, o}-{provision, removal}) are independent of particular theories of generative phonology and phonological grammars, as broadly defined in (1) & (2).

(56) **Ongoing/future work**: What is the set of possible/attested compound interactions? Are there other interactions not captured by the full typology? (How) are particular theories of generative phonology and phonological grammars able to model these interactions?

[8] Kiparsky (2015) cites a similarly local definition: Q is opaque w.r.t. x if $P(Q(x)) = P, Q(x)\ (\neq Q(P(x)))$ — that is, the output of the order $Q > P$ is not distinct from the output of the two rules applying simultaneously. This definition describes counterfeeding and counterbleeding, but not self-destructive feeding nor mutual feeding. These both involve opacity under either order of the rules, but only one order is non-distinct from simultaneous application.

A sneak peek at one of the compound relations

(57) **Ambivalence** (*Suggested pronunciation:* [ˌæmbɪˈveɪln̩s]): more than one classical relation can hold between two processes, depending on the derivation (Baković & Blumenfeld 2017).

(58) Ambivalence, based on Karok (Kenstowicz & Kisseberth 1979)

a. P: V ⟶ ∅ / V — (delete vowels after vowels)

Q: s ⟶ ʃ / [+high, −back] (C) — (palatalize *s* after *i*)

b. (niuksup,niksup) $\in P$ is a p-feeding pair: niuksup $\notin I(Q)$, niksup $\in I(Q)$.

c. (ʔuiksah,ʔuksah) $\in P$ is a p-bleeding pair: ʔuiksah $\in I(Q)$, ʔuksah $\notin I(Q)$.

☞ $P > Q$, so P feeds Q in some derivations and bleeds Q in others.

	/ni+uksup/	(*feeding*)		/ʔu+iksah/	(*bleeding*)
P	niksup		P	ʔuksah	
Q	nikʃup		Q	—	
	[nikʃup]	'point (1SG)'		[ʔuksah]	'laugh (3SG)'

(59) Ambivalence, based on Yokuts (Kenstowicz & Kisseberth 1979)

a. P: [+long] ⟶ [−high] (lower long high vowels)

Q: [αhigh] ⟶ [+round] / [αhigh, +round] C_0 — (round *i* after *u* and *a* after *o*)

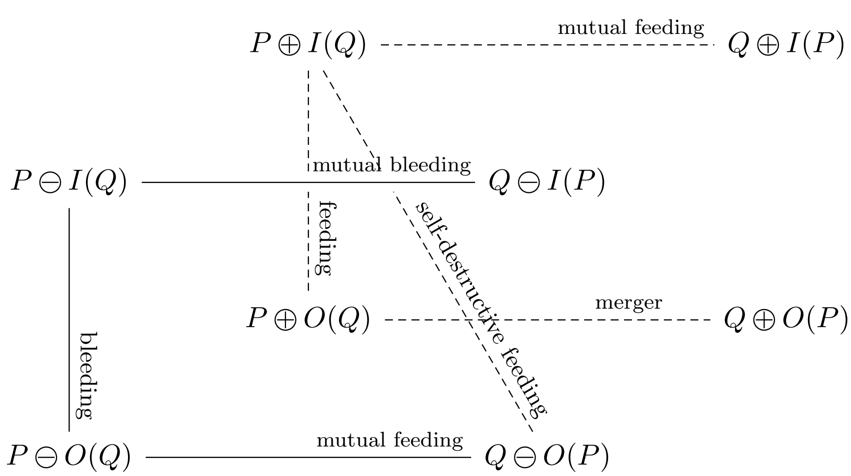
b. (ʔuːtal,ʔoːtal) $\in P$ is a p-feeding pair: ʔuːtal $\notin I(Q)$, ʔoːtal $\in I(Q)$.

c. (ʔuːtit,ʔoːtit) $\in P$ is a p-bleeding pair: ʔuːtit $\in I(Q)$, ʔoːtit $\notin I(Q)$.

☞ $Q > P$, so P counterfeeds Q in some derivations and counterbleeds Q in others.

	/ʔuːt+al/	(*counterfeeding*)		/ʔuːt+it/	(*counterbleeding*)
Q	—		Q	ʔuːtut	
P	ʔoːtal		P	ʔoːtut	
	[ʔoːtal]	'steal (DUB)'		[ʔoːtut]	'steal (AOR PASS)'

The provision/removal "cube"



Legend: ⊕ = provision, ⊖ = removal

References

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

Anderson, Stephen R. 1969. *West Scandinavian Vowel Systems and the Ordering of Phonological Rules*. Cambridge, MA, MIT Doctoral dissertation.

Baković, Eric. 2007. A revised typology of opaque generalisations. *Phonology* 24. 217–259. ROA-850, Rutgers Optimality Archive, http://roa.rutgers.edu.

Baković, Eric. 2011. Opacity and ordering. In John A. Goldsmith, Jason Riggle & Alan C. L. Yu (eds.), *The Handbook of Phonological Theory*, 40–67. Oxford: Wiley-Blackwell 2nd edn.

Baković, Eric. 2012. Breaking down rule interactions. Talk presented at the Seventh North American Phonology Conference, Concordia University, Montréal.

Baković, Eric. 2013. Defining phonological interactions with string set intersection. Talk presented at the Workshop on Opacity in Grammar, University of Leipzig.

Baković, Eric & Lev Blumenfeld. 2017. Compound relations between phonological maps: ambivalence and reciprocity. Manuscript, UC San Diego and Carleton University.

Chomsky, Noam & Morris Halle. 1968. *The Sound Pattern of English*. New York: Harper and Row.

Hout, Katherine. 2012. *The Vocalic Phonology of Mushunguli*. Columbus, OH, The Ohio State University B.A. thesis. http://kb.osu.edu/dspace/handle/1811/53373.

Kaplan, Ronald & Martin Kay. 1994. Regular models of phonological rule systems. *Computational Linguistics* 20(3). 331–378.

Karttunen, Lauri. 1998. The proper treatment of optimality theory in computational phonology. In *Proceedings of the International Workshop on Finite State Methods in Natural Language Processing*, 1–12. Stroudsburg, PA: Association for Computational Linguistics. http://www.aclweb.org/anthology/W98-1301.

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. 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. Abstractness, opacity, and global rules. In Osamu Fujimura (ed.), *Three Dimensions of Linguistic Theory*, 57–86. Tokyo: TEC.

Kiparsky, Paul. 1973b. "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. 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.

Koutsoudas, Andreas, Gerald Sanders & Craig Noll. 1974. On the application of phonological rules. *Language* 50. 1–28.

McCarthy, John J. 1999. Sympathy and phonological opacity. *Phonology* 16. 331–399.

Prince, Alan & Paul Smolensky. 1993. *Optimality Theory: Constraint Interaction in Generative Grammar*. Rutgers University Center for Cognitive Science: Technical Report RuCCS-TR-2. ROA-537, Rutgers Optimality Archive, http://roa.rutgers.edu.

Underhill, Robert. 1976. *Turkish Grammar*. Cambridge, MA: MIT Press.