On the absence of crucially-simultaneous phonological interactions in natural language[*]

Alma Frischoff and Ezer Rasin
MIT and Tel-Aviv University

Draft of September 13, 2024

Abstract

Theories of phonology should be able to generate attested types of interactions between phonological processes – including opaque interactions – and at the same time explain why certain conceivable types of interactions are unattested. We draw attention to three universals regarding unattested opaque interactions, which have been identified and defended in previous literature. These universals are expected in a rule-based theory of phonology where rules must apply serially and can never apply simultaneously. We propose to unify the three universals under a general universal called No Simultaneity, which states that there are no crucially-simultaneous phonological interactions in natural language. We then argue that this universal has important implications for theories of phonology, by showing that certain phonological theories that aim to generate opaque interactions in parallel are too permissive and over-generate unattested interactions, contrary to recent proposals that opacity is not as tied to serialism as previously thought.

1 Introduction

A fundamental property of natural-language phonology is that phonological processes can interact with each other. Consider, for example, two processes in Bedouin Hijazi Arabic (Al-Mozainy 1981): palatalization, which changes the velar stop /k/ to [k^j] before the front vowel /i/, and vowel deletion, which omits a high vowel (/i/ or /u/) in an open syllable. Both processes are stated in rule notation in (1).

(1) Two rules of Bedouin Hijazi Arabic
  a. Palatalization: $k \rightarrow k^j$ / __ i
  b. Vowel deletion: $V_{[+high]} \rightarrow \emptyset$ / __ CV

The independent application of palatalization and vowel deletion can be observed in (2a) and (2b) respectively, and their interaction is exemplified in (2c), where both processes apply.

(2) Rule interaction in Bedouin Hijazi Arabic

a.	/ħaːkim/	→	[ħaːk^jim]	(Palatalization applies)	‘ruling (m.sg.)’
b.	/tħakum-iːn/	→	[tħakmiːn]	(Vowel deletion applies)	‘they rule’
c.	/ħaːkim-iːn/	→	[ħaːk^jmiːn]	(Both rules apply)	‘ruling (m.pl.)’

[*]Acknowledgments: to be added.

In (2c), palatalization applies even though its environment of application is not present on the surface, as the vowel /i/ has been omitted by vowel deletion. This is an instance of *opacity*, as defined by Kiparsky (1971): a phenomenon where a process loses support on the surface, either because the process has applied but its conditioning environment is not present on the surface (as in (2c)), or because the process has failed to apply even though its conditioning environment is present, typically as a result of an interaction with another process (see also Baković 2007b, 2011).

Opacity receives a straightforward account under serial rule-based phonology (Chomsky and Halle 1968), which can explain the interaction in (2c) by ordering palatalization before vowel deletion. The outcome of this ordering is shown in the derivation in (3): palatalization applies first, and only then vowel deletion destroys its conditioning environment by removing /i/. (Here and below, "UR" stands for underlying representation and "SR" for surface representation.)

(3) A serial rule-based analysis of opacity

UR	/ħaːkim-iːn/
Palatalization	ħaːkʲimiːn
Vowel deletion	ħaːkʲmiːn
SR	[ħaːkʲmiːn]

The success of serial rule application in accounting for this and other kinds of opacity has led to the long-standing belief among phonologists that serialism plays a crucial role in the explanation of opacity, and therefore in phonological systems (see Kiparsky 2000 and McCarthy 2007b for a couple of representative examples).

Nevertheless, it has also been pointed out since as early as Koutsoudas et al. (1974) that certain opaque rule interactions could be derived by applying the rules *simultaneously* rather than serially. An informal definition of simultaneous rule application is the following: first, examine all rules against the input (in no particular order) and circle the targets that the rules should rewrite. Then, change all circled targets at once. (In case multiple rules make contradictory rewrite statements, simultaneous application is undefined and the contradictory changes are not made.) To see how simultaneous application works, consider again the opaque interaction in the derivation of [ħaːkʲmiːn], now assuming simultaneous rather than serial application of palatalization and vowel deletion, as shown in (4). The rules are examined against the input to the derivation, /ħaːkim-iːn/, where the environments of both rules are satisfied: /k/ is circled and is marked for undergoing palatalization because it precedes an /i/ in the input, and /i/ is circled and is marked for deletion because it is followed by a consonant-vowel sequence. Palatalization and vowel deletion apply at once, in a single derivational step, yielding the correct output [ħaːkʲmiːn].

(4) A simultaneous rule-based analysis of opacity

UR	/ħaːkim-in/
Palatalization, Deletion	ħaːkʲ∅m-in
SR	[ħaːkʲm-in]

The success of the simultaneous analysis of this example shows that serialism is not a necessary component of any theory of opacity. Under the simultaneous analysis, the reason that palatalization applies in (2c) even though its environment is destroyed is not that the destroying process applies *later*, but rather that the environment of palatalization is present *in the input*, and a rule's ability to apply to the input determines whether it applies. This is what we will call the *input-based* characterization of opacity (Ettlinger 2008, Kiparsky 2015, Pruitt 2023).

The difference between the serial and input-based characterizations of opacity is one manifestation of the long-standing debate in phonological theory regarding whether phonological computation is serial or parallel. This debate is still alive today, as new theories of opacity keep emerging within non-rule-based phonological frameworks. Within Optimality Theory (OT; Prince and Smolensky 1993/2004), the classical parallel version is known to face an unresolved opacity challenge, and competing parallel and serial extensions to the theory have been proposed to try to address the problem (e.g., Bermúdez-Otero 1999, Kiparsky 2000, Goldrick 2000, McCarthy 2000, 2003a,b, 2007b, Moreton and Smolensky 2002, Jarosz 2014, Trommer and Zimmermann 2014). Outside of OT, parallel input-based theories of opacity have been proposed within the computational finite-state literature (Chandlee, Heinz, and Jardine 2018). In a recent state-of-the-art paper on opacity in phonological theory, Pruitt (2023) discusses the success of parallel, input-based theories in accounting for a variety of attested kinds of opacity, as well as the failure of certain serial versions of OT to capture opacity in full. On this basis, she concludes that the input-based characterization of opacity is more appropriate than the serial one, and that opacity is not as tied to serialism as previously thought.

In this paper we aim to shed new light on the proper characterization of opacity and its relation to the serialism-parallelism debate. Much of the literature on opacity has focused on showing that parallel theories are sufficiently expressive to generate *attested* kinds of opaque interactions. In this paper we bring into the debate a generalization regarding opaque phonological interactions that are *unattested*, and ask whether competing theories are sufficiently restrictive to exclude them. The unattested interactions correspond to interactions that can be generated by simultaneous rule application but not by serial rule application. We will refer to such interactions as ***crucially-simultaneous interactions***.

As an example of a hypothetical crucially-simultaneous interaction, based on Wolf 2011, consider the two rules in (5a)-(5b): h-DELETION, which omits an /h/ before a consonant, and GLIDE VOCALIZATION, which changes a syllable-final glide /w/ to the vowel [u] after a consonant. Consider also the input-output mapping in (5c).

(5) Example: a hypothetical crucially-simultaneous rule interaction

a. h-DELETION: $h \rightarrow \emptyset$ / __ C

b. GLIDE VOCALIZATION: $w \rightarrow u$ / C __ $]_\sigma$

c. /gahw/ → [gau]

In (5c), both rules apply and neither's environment of application is surface-apparent, because each rule destroys the other's environment. This mapping cannot be obtained by applying the two rules serially (in any order), as shown in (6): if h-DELETION applies first, it would prevent GLIDE VOCALIZATION from applying and the final output would be the incorrect *[gaw]. And if GLIDE VOCALIZATION applies first, h-DELETION would not be able to apply and the output would be *[gahu].

(6) a. h-DELETION before GLIDE VOCALIZATION:

UR	/gahw/
h-DELETION	gaw
GLIDE VOCALIZATION	-
SR	*[gaw]

b. GLIDE VOCALIZATION before h-DELETION:

UR	/gahw/
GLIDE VOCALIZATION	gahu
h-DELETION	-
SR	*[gahu]

However, if the rules apply simultaneously, the correct output is derived, as shown in (7). This is because the context for both rules is met in the input, so both rules get to apply.

(7) Simultaneous application:

UR	/gahw/
h-DELETION, GLIDE VOCALIZATION	gau
SR	[gau]

Importantly, interactions of this kind have been argued to be unattested in natural language (Wolf 2011, following Baković 2007a). When considering the space of possible rule interactions schematized in the Venn

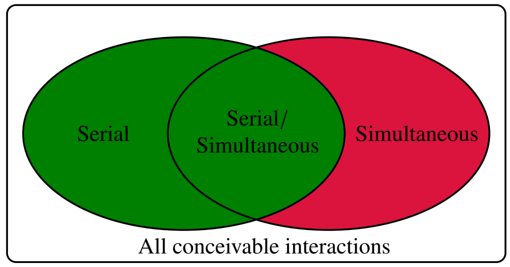


Figure 1: Space of conceivable rule interactions (green = attested, red = unattested).

diagram in Figure 1, crucially-simultaneous interactions are distinguished from crucially-serial interactions, which can only be generated by serial rule application, and from serial-or-simultaneous interactions, which can be generated by either serial or simultaneous rule application (the Bedouin Hijazi Arabic interaction in (2) is an example of the latter.) Differently from crucially-simultaneous interactions, the other two kinds of interactions are widely attested in natural languages.

We refer to this typological asymmetry as the No-Simultaneity Universal. A first statement of the universal, to be updated later on, is given in (8).

(8) No-Simultaneity Universal (to be updated in (44) below)
Crucially-simultaneous interactions are unattested in natural language.

The No-Simultaneity Universal is a new label connecting three previous observations made by Chomsky and Halle (1968), Johnson (1972), and Baković (2007a), each a generalization regarding one kind of a crucially-simultaneous interaction, as we discuss in section 2. Despite their importance, these observations have not figured prominently in the literature: Chomsky and Halle's 1968 observation was made in a footnote (p. 19, ft. 5), Johnson's observation at the very end of chapter 5 out of seven chapters of a full-length book (pp. 76–79), and Baković's in an online blogpost. To our knowledge, Wolf (2011) was the first to connect two of these generalizations and defend them against counterexamples. The first, modest contribution of this paper is therefore to show that three existing observations together form one broad typological generalization.

The second contribution of this paper is to show that the No-Simultaneity Universal can be used as a test case for competing theories of opacity, and that it provides new evidence concerning the serialism-parallelism debate. As we will see, even though the universal is stated in terms of interactions between rules, we can ask whether non-rule-based theories can generate the unattested interactions, by examining their ability to generate the phonological patterns that motivate those interactions. Concretely, we will demonstrate the significance of the universal by discussing several theories of opacity within three different phonological frameworks – rule-based phonology, OT, and finite-state phonology – and show that they explain the universal asymmetry exactly when their account of opacity is serial. Non-serial theories that follow the input-based characterization of opacity can generate the unattested crucially-simultaneous interactions as easily as they can generate attested kinds of interactions, and thus fail to capture the universal. The theories discussed are summarized in Table 1. Contrary to Pruitt's (2023) conclusion, our result provides a new argument that serialism is needed to account for opaque phonological interactions, and in particular to successfully distinguish between attested and unattested kinds of opaque interactions.

Theory	Serial?	Explains the asymmetry?
Rule-based phonology:		
Serial rule application	Yes	Yes
Simultaneous rule application	No	No
Optimality Theory:		
Comparative Markedness	No	No
Sympathy Theory	No	No
Stratal OT	Yes	Yes
Finite-state phonology:		
Input Strictly Local Maps	No	No

Table 1: Summary of the theories discussed in the paper in relation to the NO-SIMULTANEITY UNIVERSAL.

The paper is structured as follows. First, in section 2 we will present the generalizations regarding unattested rule interactions using rule-based phonology, and introduce the NO-SIMULTANETY UNIVERSAL. Then, in section 3 we evaluate different versions of OT with respect to the universal, showing that two parallel theories of opacity within OT fail to derive it, while a serial version of OT succeeds. In section 4 we discuss Input Strictly Local Maps, showing that they can easily generate the unattested opaque interactions. In section 5 we discuss a few questions that arise from this work that we leave open for future research. Section 6 concludes.

2 Crucially-simultaneous rule interactions

In this section we classify a range of interactions between phonological rules into three types, depending on whether they can be generated by serial rule application, simultaneous rule application, or both:

(9) Three types of rule interactions

1. ***Crucially-serial interactions***: interactions that can be generated by serial rule application only.
2. ***Serial-or-simultaneous interactions***: interactions that can be generated by either serial or simultaneous rule application.
3. ***Crucially-simultaneous interactions***: interactions that can be generated by simultaneous rule application only.

We start, in section 2.1, by briefly reviewing the classical typology of pairwise rule interactions, which includes *feeding*, *bleeding*, *counterfeeding*, and *counterbleeding* interactions (Kiparsky 1968, 1971, Newton 1971), all of which are widely attested in natural languages. These interactions have been defined in terms of serial rule ordering, and can trivially be generated by serial rule application. We will see that a subset of those interactions – more accurately, the input-output mappings that they define – can be generated by simultaneous rule application as well (see, e.g., Postal 1968, Koutsoudas et al. 1974, and Kenstowicz and Kisseberth 1979, chapter 8). Thus, the four classical interactions fall into the first two types in (9). Of course, these are not the only interactions belonging to these two types – see Baković and Blumenfeld (2024) for a comprehensive survey. (Readers familiar with the literature on simultaneous rule application may wish to skip section 2.1). In section 2.2, we will turn to the third type in (9) and discuss three less familiar crucially-simultaneous interactions, which cannot be generated by serial rule application. These include *mutual counterbleeding*, *mutual counterfeeding*, and *crucially-noniterative* rules, the latter concerning interactions between two applications of the same rule rather than applications of different rules. As these kinds of interactions have been argued to be unattested, we will conclude the section by establishing a version of the generalization expressed by the No-SIMULTANEITY UNIVERSAL in (8).

2.1 Classification of the basic pairwise rule interactions

The four basic pairwise interactions can be informally defined as follows (definitions adapted from McCarthy 2007a and Baković 2011):

(10) Given two rules $\mathbb{P}$, $\mathbb{Q}$ such that $\mathbb{P}$ precedes $\mathbb{Q}$ in a given derivation,

1. $\mathbb{P}$ *feeds* $\mathbb{Q}$ if $\mathbb{P}$ creates additional inputs to $\mathbb{Q}$.
2. $\mathbb{P}$ *bleeds* $\mathbb{Q}$ if $\mathbb{P}$ removes potential inputs to $\mathbb{Q}$.
3. $\mathbb{Q}$ *counterfeeds* $\mathbb{P}$ if $\mathbb{Q}$ would have fed $\mathbb{P}$ under the reverse order of application.
4. $\mathbb{Q}$ *counterbleeds* $\mathbb{P}$ if $\mathbb{Q}$ would have bled $\mathbb{P}$ under the reverse order of application.

A simple feeding interaction is the interaction between VOWEL DELETION and VOICING ASSIMILATION in Tangale, stated in (11) (Kidda 1985; presentation following Kenstowicz 1994, pp. 95–97).

(11) Two rules of Tangale

a. VOWEL DELETION: $V \rightarrow \emptyset$ / __ +

b. VOICING ASSIMILATION: [-son] $\rightarrow$ [αvoice] / $C_{[\alpha\text{voice}]}$ __

The rules and their interaction are illustrated by the data in (12). The nouns in the last two columns ('tooth' and 'bag') have a final vowel (appearing in row a) that is lost when a suffix is added (rows b–d). The consonant-initial suffixes (rows c–d) match the preceding consonant in voicing, even when a stem-final vowel has been deleted.

(12) Tangale (Chadic, Nigeria)

	'meat'	'window'	'berry'	'tooth'	'bag'
a. 'N'	loo	bugat	tugat	wudo	lutu
b. 'the N'	loo-i	bugat-i	tugad-i	wud-i	lut-i
c. 'your N'	loo-go	bugat-ko	tugad-go	wud-go	lut-ko
d. 'her N'	loo-do	bugat-to	tugad-do	wud-do	lut-to

The interaction of the two rules in words such as [lutko] is a feeding interaction, because VOICING ASSIMILATION crucially applies to the representation resulting from VOWEL DELETION, as shown in the serial derivation in (13).

(13) Serial application (successful):

UR	/lutu-go/
VOWEL DELETION	lut-go
VOICING ASSIMILATION	lut-ko
SR	[lut-ko]

In contrast, simultaneous application of the rules is unsuccessful, as shown in (14). VOICING ASSIMILATION is predicted not to apply, because its environment of application is not present in the input. As a result, the two adjacent consonants incorrectly mismatch in their voicing values on the surface.

(14) Simultaneous application (unsuccessful):

UR	/lutu-go/
VOWEL DELETION, VOICING ASSIMILATION	lut-go
SR	*[lut-go]

The problem of simultaneous application goes beyond this particular example of feeding. A typical feeding interaction between a "feeding" rule *P* and another rule *Q* is one where *P* can apply to the input but *Q* cannot. *Q* nevertheless applies to a derived representation created by the application of *P*. Simultaneous application cannot capture this behavior because it does not allow any rule to apply to derived representations, and thus fails on feeding interactions in general. To clarify, since "feeding" (like the other classical interactions) is defined within a serial rule-based theory, the statement that "simultaneous application fails on feeding interactions" does not mean that simultaneous application fails to generate a serial derivation. This is trivial. What it means is that simultaneous application fails to account for *phonological patterns* whose most natural account under a serial rule-based theory would involve a feeding interaction, as is the case in Tangale. We therefore classify feeding as a *crucially-serial* interaction.[1]

For similar reasons, *bleeding* is a crucially-serial interaction as well. As a simple example, consider the interaction between DEVOICING and SCHWA EPENTHESIS in the English plural in (15). The rules are stated in an oversimplified form in (16).

(15) English

	'bag'	'cat'	'quiz'	'glass'
sg.	bæg	kʰæt	kwɪz	glæs
pl.	bæg-z	kʰæt-s	kwɪz-əz	glæs-əz

(16) a. SCHWA EPENTHESIS: $\emptyset \rightarrow$ ə / [+strident] __ [+strident]

b. DEVOICING: [-son] $\rightarrow$ [-voice] / [-voice] __

The interaction of the rules in the derivation of words like [glæs-əz] 'glasses' can be obtained by a serial application of the rules, as shown in (17). Even though the environment of DEVOICING is present in the input, SCHWA EPENTHESIS applies first and inserts a schwa that disrupts the linear adjacency between the two stridents. As a result, DEVOICING is bled and fails to apply.

(17) Serial application (successful):

UR	/glæs-z/
SCHWA EPENTHESIS	glæs-əz
DEVOICING	-
SR	[glæs-əz]

[1] One response to the claim that simultaneous application fails on feeding interactions is that alternative rules could be stated within the simultaneous theory to capture the desired patterns, by building the context of the "feeding" rule *P* into the "fed" rule *Q*. For example, if the VOICING ASSIMILATION rule of Tangle in (11b) is restated as the more complex rule [-son] $\rightarrow$ [αvoice] / $C_{[\alpha \text{voice}]}$ (V+) __, it would apply voicing assimilation not only between adjacent consonants but also between consonants separated by a morpheme-final vowel. Applying this rule simultaneously with VOWEL DELETION in (14) would yield the correct output. Such a response, however, would be short lived. First, because in a series of feeding relations between more than two rules, each rule would have to be complicated so as to take into account the contexts of *every subset* of preceding rules, leading to massive complexity in the grammar (Postal 1968, pp. 140–152; Chomsky and Halle 1968, pp. 348–349). Worse, this response will not work when the "feeding" rule *P* is an optional rule, because *Q* would be incorrectly predicted to apply even if *P* has not. Optional phonological rules that feed other rules are attested (see, e.g., Purnell 2017, pp. 158–159).

As shown in (18), simultaneous application fails here. Since the environment of Devoicing is present in the input, simultaneous application incorrectly predicts that it should apply.

(18) Simultaneous application (unsuccessful):

UR	/glæs-∅z/
Schwa epenthesis, Devoicing	glæs-əs
SR	*[glæs-əs]

As with the feeding case, the problem of simultaneous application with bleeding interactions is general, because the application of the "bled" rule should be determined on the basis of a derived representation rather than the input. This is not possible under simultaneous application, according to which all rules apply only to the input.[2]

With both feeding and bleeding, simultaneous application incorrectly generates an opaque rule interaction which corresponds to the opposite order of application: counterfeeding and counterbleeding, respectively. In the case of feeding in Tangale, the outcome was opaque underapplication of voicing assimilation (14), and in the bleeding case of English, opaque overapplication of devoicing. Opacity is therefore the natural state of affairs under simultaneous application, and counterfeeding and counterbleeding interactions fall under the *serial-or-simultaneous* type of interactions in our classification in (9). Let us see this using attested examples of counterfeeding and counterbleeding, which will serve us in later sections of this paper.

Consider the following counterfeeding interaction between two rules in Catalan (Faust and Torres-Tamarit 2017, Mascaró 1976), which we have simplified greatly: a rule of nasal deletion, which deletes nasal consonants word-finally, and a rule of cluster simplification, which deletes word-final stops after a nasal, both stated in (19).

(19) Two rules of Catalan

a. Nasal deletion: $\mathrm{N} \rightarrow \emptyset$ / __ #

b. Cluster simplification: $\mathrm{T} \rightarrow \emptyset$ / N __ #

The two tables in (20) and (21) illustrate the application of the two rules, respectively. In the masculine examples in (21), a nasal consonant appears word-finally despite there being a rule in the grammar that deletes nasals in this position. These are not isolated exceptions in the language: whenever a nasal consonant becomes word-final as a result of Cluster simplification, the rule of Nasal deletion does not apply.

(20)

	'cousin'	'plan'
sg.	kuzi	pla
pl.	kuzin-s	plan-s

(21)

	'hot'	'white'
masc.	kəlen	blaŋ
fem.	kəlent-ə	blaŋk-ə

A serial rule-based account of this interaction would order Nasal deletion before Cluster simplification, as shown in (22). As a result, by the time Cluster simplification applies and makes a nasal word-final, it is too late for Nasal deletion to apply, and the nasal remains on the surface. In this interaction, Cluster simplification *counterfeeds* Nasal deletion, because under the opposite order of application, Cluster simplification would have created a new input for Nasal deletion (/kəlent/ → |kəlen|), causing Nasal deletion to apply (|kəlen| → *[kəle]).

[2]The same qualifications about feeding from ft. 1 also hold for bleeding, in that an attempt to account for bleeding simultaneously by restating the rules would be short lived. See the references in ft. 1, as well as Kenstowicz and Kisseberth (1979, p. 311) for how an optional bleeding rule challenges rule restatement under simultaneous application.

(22) Serial application (successful):

UR	/kəlent/
Nasal deletion	-
Cluster simplification	kəlen
SR	[kəlen]

By applying the rules simultaneously, the same output is obtained: Nasal deletion does not apply, because its environment is not present in the input, as shown in (23).

(23) Simultaneous application (successful):

UR	/kəlent/
Nasal deletion, Cluster simplification	kəlen
SR	[kəlen]

The success of the simultaneous account of counterfeeding, and thus of the input-based characterization of opacity, shows that serialism is not required to generate counterfeeding interactions. What matters is that the "counterfed" rule (in the Catalan case above, Nasal deletion) applies to the input and is not ordered after the "counterfeeding" rule (Cluster simplification above). Counterfeeding is thus a *serial-or-simultaneous* interaction according to the classification in (9).

For counterbleeding, we have already seen a counterbleeding interaction in Bedouin Hijazi Arabic (2), and we saw that it could be generated by either serial or simultaneous application. We will return to that example in our theoretical discussion in the following sections.

The table in (24) summarizes the discussion in this section. Feeding and bleeding are crucially-serial interactions, whereas counterfeeding and counterbleeding can be generated by either serial or simultaneous application. In the next section we turn to the third logical option: interactions that can be generated by simultaneous application only.

(24)

Interaction ↓ Theory →	Serial	Simultaneous
Feeding	✓	✗
Bleeding	✓	✗
Counterfeeding	✓	✓
Counterbleeding	✓	✓

2.2 Three kinds of crucially-simultaneous interactions

2.2.1 Mutual-counterbleeding

The first crucially-simultaneous interaction, *mutual-counterbleeding*, is the one already discussed in the introduction. Here we repeat the example and discuss it in more detail. This interaction, which was introduced by Baković (2007b), is between two rules that can destroy each other's environment of application. When the environment of both processes is met, both of them apply.

Consider again the hypothetical interaction from the introduction, based on Wolf (2011), between h-deletion and Glide vocalization, stated again in (25).

(25) Two rules of a hypothetical language

a. h-deletion: h → ∅ / __ C

b. Glide vocalization: w → u / C __ $]_\sigma$

It is easy to construct a concrete hypothetical language where each process would receive independent support from alternations. Consider the paradigms in (26), where each row represents a different noun and the columns represent inflections in four different grammatical cases: Nominative, Accusative, Genitive, and Dative.

(26) A hypothetical language exhibiting mutual-counterbleeding

	Nom.	Acc.	Gen.	Dat.
a.	ma-t	ma-e	ma-u	ma-w
b.	dar-t	dar-e	dar-u	dar-u
c.	ga-t	gah-e	gah-u	ga-u

The vowel-final noun /ma/ in the first row and the consonant-final noun /dar/ in the second row together reveal an alternation in the realization of the dative suffix /-w/, which is realized as [-u] after [dar]. That the alternation changes /w/ to [u] rather than the reverse could be determined by observing the genitive suffix /-u/, which is realized consistently as [-u] after both vowel-final and consonant-final roots. The h-final noun /gah/ in the third row loses its /h/ before the consonantal nominative suffix /-t/. This is a case of deletion rather than epenthesis, because otherwise an [h] would have also been inserted before the accusative and genitive suffixes in the first row. Finally, the interaction between deletion and vocalization is observed in the dative form in the third row, whose UR can be inferred to be /gah-w/ on the basis of the reasoning presented up to this point. Here, the context for both deletion and vocalization is met in the input, and both processes apply, yielding the output form [gau].

The interaction between Glide vocalization and h-deletion in the derivation /gah-w/ → [gau] is doubly opaque, as both processes apply even though neither's environment is present on the surface: Glide vocalization destroys the environment for h-deletion and vice versa. As we have seen in the introduction, this interaction cannot be generated by applying the two rules serially, but it can be generated if the rules apply simultaneously. The two failed serial derivations and the successful simultaneous derivation are repeated in (27) and (28). In the serial derivation in (27a), Glide vocalization counterbleeds h-deletion, and in (27b) the opposite is true. This interaction was therefore named *mutual-counterbleeding* by Wolf (2011) (Baković 2007b calls it "mutually-assured destruction"), and we classify it as a crucially-simultaneous interaction.

(27) a. h-deletion before Glide vocalization:

UR	/gahw/
h-deletion	gaw
Glide vocalization	-
SR	*[gaw]

b. Glide vocalization before h-deletion:

UR	/gahw/
Glide vocalization	gahu
h-deletion	-
SR	*[gahu]

(28) Simultaneous application:

UR	/gahw/
h-deletion, Glide vocalization	gau
SR	[gau]

Aside from the interaction required to generate the data, there is nothing unusual about the hypothetical language in (26). As noted by Wolf (2011), the rules of Glide vocalization and h-deletion which are responsible for the morphophonological alternations are natural rules found in attested languages. There is also nothing unusual about the coexistence of two rules that can destroy each other's environment in a single language. When such rules interact, only one rule applies, corresponding to a serial application with that rule ordered first. Such interactions have been referred to as "mutual bleeding" in the literature. Examples include the interaction between devoicing and cluster simplification in German (Kiparsky 1971, p. 66), two deletion rules in Ukranian and Russian (Kenstowicz and Kisseberth 1979, pp. 311–313), epenthesis and deletion in Lardil (Baković 2011, p. 62), epenthesis and degemination in a variety of Singapore English (Baković 2011, p. 63), and others. Nevertheless, the interaction corresponding to the simultaneous application of the rules, mutual-counterbleeding, has been argued to be unattested by Wolf (2011), following Baković (2007b), making it an unattested crucially-simultaneous interaction. In the next section we turn to the feeding counterpart of mutual-counterbleeding, which constitutes another crucially-simultaneous interaction.

2.2.2 Mutual-counterfeeding

The second kind of crucially-simultaneous interactions, *mutual-counterfeeding*, occurs when two rules can feed each other. Whenever one of the rules applies and creates a new environment for the other rule, the other rule fails to apply.

Consider the following hypothetical language, which is also a simplified version of an example presented by Wolf (2011). In this language, there are two rules, h-DELETION, as in the previous section, and SCHWA SYNCOPE, a rule that deletes a schwa in a two-sided open-syllable environment (VC __ CV). The rules are stated in (29). Consider also the data in (30).

(29) Two rules of a hypothetical language

a. h-DELETION: $\mathrm{h} \rightarrow \emptyset$ / __ C

b. SCHWA SYNCOPE: ə → $\emptyset$ / VC __ CV

(30)

	NOM.	ACC.	GEN.
a.	gah-e	ga-t	ga-rəmu
b.	ma-e	ma-t	ma-rmu
c.	ahp-e	ahəp-t	ahəp-rəmu

The application of h-DELETION is demonstrated in row (a), where the h-final root /gah/, which surfaces without change in the nominative case, loses its /h/ before the consonant-initial accusative and genitive suffixes. (An h-insertion analysis would face the challenge of explaining why there is no insertion in the accusative after the root /ma/ in row (b).) The application of SCHWA SYNCOPE is illustrated in two places: the first is the schwa-zero alternation in the root /ahəp/ in row (c), where a schwa appears before the consonant-initial suffixes of the accusative and genitive, but disappears before the vocalic nominative suffix. The second place is the alternation in the genitive suffix /-rəmu/, where a schwa appears when the root is consonant-initial, as in row (c), but disappears after the vowel-final root in row (b). The presence of the schwa in the genitive suffix in row (a) is an example of the interaction between h-DELETION and SCHWA SYNCOPE. From the reasoning up to this point we can conclude that the UR of this form is /gah-rəmu/. Here, h-DELETION applies, and as a result the environment of SCHWA SYNCOPE is met. Nevertheless, SCHWA SYNCOPE does not apply. At the same time, the environment of SCHWA SYNCOPE is met in the inferred UR of the nominative form in row (c), /ahəp-e/. SCHWA SYNCOPE applies, and as a result the context for h-DELETION is met, but this time h-DELETION underapplies.[3]

The interactions between h-DELETION and SCHWA SYNCOPE in the derivations /gah-rəmu/ → [ga-rəmu] and /ahəp-e/ → [ahp-e] are both opaque counterfeeding interactions, which together pose an ordering paradox

[3]Additional examples could be introduced to support SCHWA SYNCOPE over an alternative schwa insertion rule, as well as the precise environments of application of both h-DELETION and SCHWA SYNCOPE. Such examples were omitted to keep the presentation simple.

for serial rule application. In order for h-Deletion to counterfeed Schwa syncope in the derivation /gah-rəmu/ → [ga-rəmu], Schwa syncope should be ordered before h-Deletion. But in order for Schwa syncope to counterfeed h-Deletion in the derivation /ahəp-e/ → [ahp-e], the reverse order is necessary. The result, as shown in (31), is that no matter what order is chosen, one of the derivations would involve a feeding interaction rather than counterfeeding, which would lead to incorrect outputs.

(31) a. Schwa syncope before h-Deletion:

UR	/gah-rəmu/	/ahəp-e/
Schwa syncope	-	ahpe
h-Deletion	garəmu	ape
SR	[garəmu]	*[ape]

b. h-Deletion before Schwa syncope:

UR	/gah-rəmu/	/ahəp-e/
h-Deletion	garəmu	-
Schwa syncope	garmu	ahpe
SR	*[garmu]	[ahpe]

Differently from serial application, simultaneous application can generate this interaction without any difficulties, since under simultaneous application the rules apply to the input and no rule can apply to the output of any other rule. A successful simultaneous derivation is shown in (32). Mutual-counterfeeding is therefore a crucially-simultaneous interaction.

(32) Simultaneous application:

UR	/gah-rəmu/	/ahəp-e/
Schwa syncope, h-Deletion	garəmu	ahpe
SR	[garəmu]	[ahpe]

As in the case of mutual counterbleeding, there is nothing particularly unusual about the hypothetical language in (30) apart from the interaction between the rules. As Wolf (2011) notes, both rules are familiar rules that occur in attested languages. In addition, the coexistence of two rules that can feed each other in a single language is also attested, often giving rise to interactions discussed in the literature under names such as *fed-counterfeeding* (Kavitskaya and Staroverov 2010) and *Duke-of-York* (Pullum 1976, McCarthy 2003b). Consider, for example, the fed-counterfeeding interaction in Lardil (see Kenstowicz and Kisseberth 1979, pp. 112-113; data from Hale 1973), involving the rules of Apocope, which deletes word-final vowels in nouns that have more than two syllables, and Consonant deletion, which deletes word-final non-apical consonants. The two rules are stated in (33) and their interaction is demonstrated in (34).

(33) Two deletion rules in Lardil

a. Apocope: $V \rightarrow \emptyset$ / VC_1VC_1__ #

b. Consonant deletion: $C_{\text{[-apical]}} \rightarrow \emptyset$ / __ #

(34) Rule interaction in Lardil

a.	/wiwala/	→	[wiwal]	(Apocope applies)	'bush mango'
b.	/thuraraŋ/	→	[thurara]	(Consonant deletion applies)	'shark'
c.	/ŋawuŋawu/	→	[ŋawuŋa]	(Both rules apply)	'termite'

The derivation in (34a) illustrates a simple application of Apocope, which deletes the final vowel of the word and makes the apical consonant /l/ word-final. Example (34b) shows an application of Consonant deletion, which deletes the non-apical consonant /ŋ/ and makes the preceding vowel word-final. This example shows that Apocope does not apply to vowels that have been made final by Consonant deletion, suggesting an ordering of Apocope before Consonant deletion. In (34c), Apocope applies and feeds Consonant

DELETION, which deletes the non-apical consonant /w/ and thus creates a new environment for the application of APOCOPE. As predicted by the ordering of APOCOPE before CONSONANT DELETION, APOCOPE underapplies. The two Lardil rules have the property that they can feed each other. If applied simultaneously, the interaction between the rules would have been mutual counterfeeding: CONSONANT DELETION would have counterfed APOCOPE in the derivation /thuraraŋ/ → [thurara] (as is the case in actual Lardil), and APOCOPE would have counterfed CONSONANT DELETION in the hypothetical derivation /ŋawuŋawu/ → [ŋawuŋaw]. Examples of other cases of fed-counterfeeding include the interaction of debuccalization and apocope in Tundra Nenets (Kavitskaya and Staroverov 2010) and the interaction of syncope and schwa epenthesis in Judeo-Baghdadi Arabic (Bistry et al. 2023). In all these cases, two rules that can feed each other interact as predicted by the serial rule-based theory. The same is true for Duke-of-York derivations (see McCarthy 2003b for a variety of examples).

Chomsky and Halle (1968, p. 19, fn. 5) were the first to claim (using different terminology) that mutual-counterfeeding is unattested in natural language, and used its absence as an argument that rules should apply serially rather than simultaneously. One potential exception they mentioned involved "feature exchange" rules (sometimes called *polarity* rules in the literature), which replace an underlying /-/ feature with [+] and an underlying /+/ feature with [-] in the same environment. Such cases were handled using the alpha-variable notation, such that the change required by a feature exchange rule would be stated as [αF] → [-αF]. Such cases, according to them, therefore do not constitute real cases of mutual-counterfeeding. Wolf (2011), who named mutual-counterfeeding, revisited Chomsky & Halle's claim and concluded, after reviewing potential counterexamples, that no convincing cases of mutual-counterfeeding are attested. Mutual-counterfeeding is therefore another kind of an unattested crucially-simultaneous interaction.

Having introduced both mutual-counterbleeding and mutual-counterfeeding as (unattested) crucially-simultaneous interactions, we can expand the table of interactions in (24) to include them as well. The updated table is given in (35). This table distinguishes between the basic crucially-serial interactions (feeding and bleeding), serial-or-simultaneous interactions (counterfeeding and counterbleeding), and crucially-simultaneous interactions (mutual-counterfeeding and mutual-counterbleeding). All of these interactions are between applications of different rules. But the question of serial versus simultaneous application also arises in the case of interactions between multiple applications of the same rule. There, as well, crucially-simultaneous interactions have been argued to be unattested, as we discuss in the next section.

(35)

Interaction ↓ Theory →	Serial	Simultaneous
Feeding	✓	✗
Bleeding	✓	✗
Counterfeeding	✓	✓
Counterbleeding	✓	✓
Mutual-counterfeeding	✗	✓
Mutual-counterbleeding	✗	✓

2.2.3 Crucially-noniterative interactions

The environment of a single rule could be met more than once in the UR or in the course of the derivation. When one application can create or destroy the context for another application of the same rule, two applications of one rule can interact with each other. A question then arises as to whether a single rule applies serially or simultaneously to its targets, though the details are different from the case of the relative application of multiple rules (see Kenstowicz and Kisseberth 1977, chapter 5 for extensive discussion). Here we present Johnson's 1972 generalization that the crucially-simultaneous application of a single rule to multiple targets is unattested.

To set the stage for Johnson's generalization, consider first the application of progressive rounding vowel harmony (36) in two dialects of Crimean Tatar (McCollum and Kavitskaya 2018).

(36) Vowel harmony
V → [+round] / [+round] C_0__

According to McCollum and Kavitskaya (2018), the two dialects differ in whether harmony applies more than once, as illustrated in (37). In the southern dialect (37a), harmony causes rounding to spread rightwards until it reaches the end of the word. In contrast, in the central dialect (37b), rounding spreads only to one following vowel.

(37) Application of Vowel harmony in two dialects of Crimean Tatar
   a. Southern dialect: /tuz-lɯɣ-ɯ/ → [tuz-luɣ-u]
   b. Central dialect: /tuz-lɯɣ-ɯ/ → [tuz-luɣ-ɯ]

In terms of rule application, the difference between the two dialects could reflect a difference between the iterative and simultaneous application of Vowel harmony, as shown in (38).

(38) a. Southern dialect: self-feeding

UR	/tuz-lɯɣ-ɯ/
Vowel harmony	tuz-luɣ-ɯ
Vowel harmony	tuz-luɣ-u
SR	[tuz-luɣ-u]

b. Central dialect: self-counterfeeding

UR	/tuz-lɯɣ-ɯ/
Vowel harmony	tuz-luɣ-ɯ
SR	[tuz-luɣ-ɯ]

In the southern dialect (38a), Vowel harmony can be said to apply iteratively, each time creating a new context for itself to apply, an example of *self-feeding*. In the central dialect, however, even though the environment for another application of harmony is met, harmony does not reapply, a case of underapplication opacity sometimes called *self-counterfeeding*. This behavior could be generated by applying Vowel harmony simultaneously, where simultaneous application of a single rule could be defined similarly to the simultaneous application of multiple rules (Chomsky and Halle 1968, p. 344): first identify all segments in the input that satisfy the environment of the rule, then make all changes at once. As in the case of the application of multiple rules, we will assume that all identified segments are circled. In (38b), this means that only the second vowel in the input should be circled and rounded, because it is the only non-round vowel that is immediately preceded by a round vowel (ignoring the intervening consonants). Some of the early generative literature proposed that a phonological rule at least has the option of applying simultaneously: Chomsky and Halle (1968) in fact proposed that a rule *must* apply simultaneously to its input, whereas other proposals took iterative versus simultaneous application to be a rule-specific parameter (e.g., Anderson 1974).

Taking a different line, Johnson (1972) proposed that differences such as that reflected in (38) are not due to iterative versus simultaneous application. Rather, he proposed that all rules apply iteratively, and that differences like (38) are due to the direction of iterative application: a rule can apply to the input string either from left to right or from right to left; in (38a), self-feeding corresponds to the left-to-right application of the vowel harmony rule, and what seems like simultaneous application in (38b) is in fact derived by applying the rule in the opposite direction, from right to left. Starting from the end of the word, the right-to-left rule would not apply to the final vowel of the word, because at this point the preceding vowel is non-round. By the time the rule has reached the preceding vowel and has changed it to a round vowel, it must proceed leftwards and cannot go back to change the final vowel.

The effect of simultaneous application can be obtained by applying the rule iteratively in the opposite direction also in *self-bleeding* and *self-counterbleeding* interactions, which can result from rules that destroy their own environment rather than create one. Kenstowicz and Kisseberth (1979, pp. 319–321) provide an example of the Rhythmic Law in (39), which shortens a long vowel in case the preceding vowel is long.

(39) RHYTHMIC LAW
$V \rightarrow$ [-long] / V:C_0 __

In Gidabal (citing Geytenbeek and Geytenbeek 1971), the rule seems to apply in a self-bleeding fashion: as illustrated in (40a), in an underlying sequence of three long vowels, only the middle long vowel is shortened, as would be expected if the rule applies iteratively from left to right. The rule bleeds itself in the sense that shortening the middle vowel removes the context for shortening the final vowel, which remains long on the surface. In Slovak, however, such a sequence is resolved by shortening all vowels except the first, a case of overapplication opacity illustrated in (40b). The rule counterbleeds itself in the sense that its application to the final vowel occurs while the environment for this application is no longer present on the surface, as a result of the application of the rule to the penultimate vowel. This interaction is the excepted outcome if the Rhythmic Law applies simultaneously to the input, as indicated by the circled vowels in (40b). But the same outcome could be obtained by applying the rule from right to left: in a right-to-left application, the rule would first shorten the final vowel, then proceed leftwards to shorten the penultimate vowel.

(40) a. Gidabal: self-bleeding

UR	/djalum-baː-daːŋ-beː/
Rhythmic Law	djalum-baː-daŋ-beː
Rhythmic Law	-
SR	[djalum-baː-daŋ-beː]
Gloss	‘is certainly right on the fish’

b. Slovak: self-counterbleeding

UR	/piːs-aː-v-aː/
Rhythmic Law	piːs-av-a
SR	[piːs-av-a]
Gloss	‘write.3.sg’

Johnson (1972) claims that the success of directional-iterative application makes simultaneous application redundant. Importantly, he further observes that there are hypothetical interactions that can be generated by simultaneous application and which cannot be reduced to left-to-right iterative or right-to-left iterative application. We will refer to such interactions as *crucially-noniterative* (where “noniterative” is used synonymously with “simultaneous” in the context of a single rule):

(41) *Crucially-noniterative interaction*
An interaction between two applications of the same rule that cannot be derived by left-to-right or right-to-left iterative application of the rule.

Crucially-noniterative interactions can result from rules with a two-sided context, as exemplified in (42).

(42) Hypothetical example: a crucially-noniterative interaction

a. $V \rightarrow \emptyset$ / VC __ CV

b. /VC$\underline{V}$C$\underline{V}$CV/ $\rightarrow$ [VCCCV]

The rule in (42a) is a familiar rule that deletes a vowel in a two-sided open-syllable environment. In the input-output mapping in (42b) the rule applies twice to the input, which contains two vowels in the right context (underlined). This interaction is crucially-noniterative because applying the rule iteratively would delete only one vowel. If the rule applies from right to left, the result would be *[VCVCCV], and if it applies from left to right, the result would be *[VCCVCV]. A summary of the different interactions between multiple applications of the same rule is given in Table (43)

(43)

Interaction ↓ Theory →	Serial (iterative)	Simultaneous
Self-feeding	✓	✗
Self-bleeding	✓	✗
Self-counterfeeding	✓	✓
Self-counterbleeding	✓	✓
Crucially-noniterative	✗	✓

Johnson's generalization (Johnson 1972, pp. 76–79) is that, despite the attestedness of left-to-right and right-to-left iterative rules in natural language, crucially noniterative rules are unattested.[4] This is again not because there is something unusual about the rules themselves or about the conditions required to test their mode of application. For example, considering the deletion rule in (42), Yadav (1996, pp. 51–52) suggests that a right-to-left application of a two-sided open-syllable schwa syncope rule is attested in Maithili, at least for some speakers (/nikə̲l-ə̲t-ah/ → [nikəltah] 'will come out.3.HONORIFIC'; cf. /nikə̲l-ət/ → [niklət] 'will come out'). Gouskova (2003, p. 128), citing Phelps (1975), discusses a left-to-right application of a similar rule deleting /a/ in Tonkawa (/we-ya̲ka̲pa-oʔ/ → [weykapoʔ] 'he hits them'; cf. /yaka̲pa-oʔ/ → [yakpoʔ] 'he hits it').

Johnson used his generalization as a typological argument against theories that allow a single rule to apply simultaneously to its target. Unless counterexamples to this generalizations are found, crucially-nonoiterative interactions form a third kind of crucially-simultaneous interactions that are unattested, in this case interactions between multiple applications of a single rule rather than interactions between different rules.

2.3 Interim conclusion: No Simultaneity

Given the discussion up to this point, the No-Simultaneity Universal from the introduction can be updated as follows:

(44) No-Simultaneity Universal (final version)
Crucially-simultaneous interactions (between two applications of the same rule or between different rules) are unattested in natural language.

The formulation in (44) makes it clear that, at least until strong counterexamples are found, the absence of crucially-simultaneous interactions is a general property of natural-language phonology, which holds both for interactions between different rules and for interactions between applications of the same rule.

The No-Simultaneity Universal does not seem like an accident. The phonological literature since the 1960s is filled with crucially-serial (e.g., feeding, bleeding) and serial-or-simultaneous (e.g., counterfeeding, counterbleeding) interactions, which can be found in phonology textbooks, language grammars, research articles, and so on. But if the observations due to Chomsky and Halle (1968), Baković (2007a), Wolf

[4] This generalization is different from that made by Kaplan (2008), who argued that reported examples of self-counterfeeding could be reanalyzed and that true self-counterfeeding does not exist. However, true cases of self-counterfeeding are attested in a variety of languages (McCollum and Kavitskaya 2022).

(2011), and Johnson (1972) are correct, there is not a single convincing example of a crucially-simultaneous interaction. This stark asymmetry must be explained by something. We take this to mean that the No-Simultaneity Universal provides evidence in favor of serial rule application over theories incorporating simultaneous rule application, as such theories fail to capture the typological asymmetry expressed by the universal.

In what follows, we will examine a range of non-rule-based theories of opacity and use the No-Simultaneity Universal as a test case. For each theory, we will ask whether the asymmetry between crucially-simultaneous interactions and attested opaque interactions can be captured. We start, in the next section, by examining theories of opacity embedded within Optimality Theory.

3 No-Simultaneity and Optimality Theory

3.1 The opacity problem for classic Optimality Theory

Unlike rule-based phonology, attested kinds of opacity pose a well-known generative challenge to classic OT (Prince and Smolensky 1993/2004), which relies on surface-oriented markedness constraints to trigger phonological processes and apply them in parallel (in the context of OT, which lacks the notion of "process" as a primitive, parallel application means that multiple phonological changes are made in one derivational step that maps the input to the output). In response to the opacity challenge, a variety of parallel and serial extensions to the classic theory have been proposed. In this section, we review three extensions that provide a successful general account for certain types of opacity – Comparative Markedness (McCarthy 2003a), Sympathy Theory (McCarthy 1999, 2003b), and Stratal OT (Bermúdez-Otero 1999, Kiparsky 2000) – and discuss their ability to generate input-output mappings that correspond to crucially-simultaneous interactions. Importantly, since OT differs from rule-based phonology on multiple dimensions, there is no immediate correlation between how an interaction is classified within rule-based phonology and whether it can be generated by a parallel or serial version of OT. For example, as is well known, OT can easily generate feeding and bleeding interactions – which are crucially-serial – in a fully parallel fashion. As for crucially-simultaneous interactions, the details of each OT extension turn out to matter, as we will show in this section.

To see the challenge opacity poses for the basic version of OT, consider again the counterbleeding interaction between palatalization and vowel syncope in Bedouin Hijazi Arabic. As shown by McCarthy (2007b, pp. 24–25), a simple Parallel-OT attempt to account for the behavior of palatalization in Bedouin Hijazi Arabic could use markedness constraints that trigger palatalization and i-deletion (here, *ki and *iCV respectively) and rank them over the corresponding faithfulness constraints that militate against palatalization and deletion (here, Ident[back] and Max respectively). Consider the ranking in the tableau in (45).

(45) A failed Parallel-OT attempt to derive [ħaːkʲm-iːn]

		/ħaːkim-iːn/	*ki	*iCV	Ident[back]	Max
a.		ħaːkimiːn	*!	*		
b.		ħaːkʲimiːn		*!	*	
c.	☞	ħaːkmiːn				*
d.	☹	ħaːkʲmiːn			*!	*

The desired candidate (45d), in which both palatalization and syncope have applied, loses to candidate (45c), in which only syncope has applied. The reason is that deleting the vowel /i/ satisfies both markedness constraints, so there is no surface motivation to also palatalize the /k/. Candidate (45d) is therefore harmonically bounded by candidate (45c) and loses under any ranking of these constraints. No other choice of basic markedness and faithfulness constraints would work either.[5]

Consider now the example of counterfeeding in Catalan, which involved Nasal deletion and Cluster simplification. Relevant constraints are the markedness constraints *N# and *NT#, which trigger final-nasal deletion and cluster simplification respectively, and the faithfulness constraints Max[N] and Max[T], which penalize deletions of nasals and stops. In order to enforce final-nasal deletion, the markedness constraint *N# must outrank the faithfulness constraint Max[N], as shown in tableau (46) for the mapping /kuzin/→[kuzi].

(46)

/kuzin/	*NT#	Max[T]	*N#	Max[N]
a. kuzin			*!	
b. ☞ kuzi				*

The same ranking, however, predicts that a nasal should be deleted even when it becomes word-final as a result of cluster simplification. This is shown in tableau (47) for the mapping /kəlent/→[kəlen]. The only way to change the ranking so as to make the correct candidate (47b) win is by reversing the ranking of *N# and Max[N]. But this will lead to the incorrect faithful candidate winning in (46) – a ranking paradox. No other basic markedness or faithfulness constraint could help either.

(47)

/kəlent/	*NT#	Max[T]	*N#	Max[N]
a. kəlent	*!			
b. ☹ kəlen		*	*!	
c. ☞ kəle		*		*

Overall, classic OT fails on opacity because of its over-reliance on surface-oriented constraints. In the next section we turn to Comparative Markedness, a version of OT that tries to amend the problem by introducing a new type of markedness constraint that looks at the input as well.

3.2 Comparative Markedness

Comparative Markedness (McCarthy 2003a) is an extension to classic OT that aims to provide a general account of certain types of opacity, including counterfeeding. This extension has been compared to other versions of OT, and some advantages and disadvantages have already been pointed out in the literature (see, e.g., Blumenfeld 2003, McCarthy 2003c). Here we show that the No-Simultaneity Universal provides a new test case for Comparative Markedness: the theory tries to generate counterfeeding in parallel, without intermediate representations, by referencing the input. As a result, the theory over-generates mutual-counterfeeding interactions.

We will introduce Comparative Markedness by showing how it succeeds on simple cases of counterfeeding, using Catalan as an illustrative example. Recall from tableau (47), repeated here as (48), that the problem for classic OT is to select the correct candidate (48b) in which a nasal appears word-finally, given that *N# needs to outrank Max[N] in order to delete nasals that occur in final position in the input.

(48)

/kəlent/	*NT#	Max[T]	*N#	Max[N]
a. kəlent	*!			
b. ☹ kəlen		*	*!	
c. ☞ kəle		*		*

[5] Similar cases of counterbleeding have been accounted for using Max[F] constraints or through coalescence mechanisms. See McCarthy (2007b, pp. 33–34) for arguments against these approaches to counterbleeding.

Comparative Markedness deals with this problem by revising the standard interpretation of markedness. It assumes that markedness violations are determined in comparison to the fully faithful candidate (FFC), and it distinguishes between two types of markedness violations: an "old" markedness violation is a markedness violation that is shared with the FFC, whereas a "new" markedness violation is one that is not shared with the FFC. Traditional markedness constraints are accordingly split into two: every markedness constraint M is replaced in Comparative Markedness by two markedness constraints, $_{\mathrm{O}}$M and $_{\mathrm{N}}$M, which penalize "old" and "new" markedness violations respectively, and which are ranked separately.

Evaluating markedness in this way is relevant for counterfeeding opacity because in simple cases of counterfeeding the environment for the opaque process is present in the output but not in the input. In Catalan, final-nasal deletion applies to nasals that are already word-final in the UR (e.g., /kuzin/→[kuzi]) but fails to apply when the nasal was not final in the UR (e.g., /kəlent/→[kəlen]). Comparative Markedness can make the correct distinction between these cases by ranking the "old" markedness constraint $_{\mathrm{O}}$*N# high enough to penalize only word-final nasals that were final in the input. The "new" markedness constraint $_{\mathrm{N}}$*N# can be ranked lower than faithfulness to ensure the survival of nasals that become final as a result of cluster simplification, as in candidate (48b). The crucial ranking is given in (49).

(49) $_{\mathrm{O}}$*N# ≫ Max[N] ≫ $_{\mathrm{N}}$*N#

As shown by the tableaux in (50), this ranking solves the problem for classic OT, as nasal deletion can be triggered in isolation (50a) and blocked when it interacts with cluster simplification (50b).

(50) a.

/kuzin/	*NT#	Max[T]	$_{\mathrm{O}}$*N#	Max[N]	$_{\mathrm{N}}$*N#
a. kuzin			*!		
b. ☞ kuzi		*		*	

b.

/kəlent/	*NT#	Max[T]	$_{\mathrm{O}}$*N#	Max[N]	$_{\mathrm{N}}$*N#
a. kəlent	*!				
b. ☞ kəlen		*			*
c. kəle		*		*!	

More generally, Comparative Markedness deals with counterfeeding interactions using the ranking schema in (51), where "old" markedness outranks faithfulness, which in turn outranks "new" markedness.

(51) Ranking schema for counterfeeding opacity
$_{\mathrm{O}}$M ≫ Faith ≫ $_{\mathrm{N}}$M

We can see that this account of counterfeeding follows the input-based characterization of opacity, because the ranking schema in (51) compels the opaque process to apply only when its environment is present in the input. The problem is that the same ranking schema can be used to generate mutual-counterfeeding interactions as well, because mutual-counterfeeding involves two counterfeeding interactions, and what determines whether each process applies is whether its environment is present in the input.

Consider again the hypothetical example of mutual-counterfeeding from section 2.2.2, summarized in (52).

(52) Mutual-counterfeeding in a hypothetical language

a. Rules:

- h-deletion: h → ∅ / __ C

- SCHWA SYNCOPE: ə → ∅ / VC __ CV

b. Mappings:
- /gah-rəmu/ → [ga-rəmu] (SCHWA SYNCOPE underapplies)
- /ahəp-e/ → [ahp-e] (h-DELETION underapplies)

Each of the two counterfeeding interactions could be generated by ranking the relevant constraints according to the schema in (51). Assuming that the markedness constraint triggering h-deletion is *hC and the one triggering schwa syncope is *VCəCV, the crucial rankings are the following:

(53) Ranking statements for mutual-counterfeeding
a. $_{O}$*hC ≫ MAX[h] ≫ $_{N}$*hC
b. $_{O}$*VCəCV ≫ MAX[ə] ≫ $_{N}$*VCəCV

A combination of these ranking statements correctly generates the two counterfeeding interactions without trouble, as shown in the tableaux in (54).[6]

(54) a.

	/gah-rəmu/	$_{O}$*hC	$_{O}$*VCəCV#	MAX[h]	MAX[ə]	$_{N}$*hC	$_{N}$*VCəCV#
a.	gah-rəmu	*!					
b. ☞	ga-rəmu			*			*
c.	ga-rmu			*	*!		

b.

	/ahəp-e/	$_{O}$*hC	$_{O}$*VCəCV#	MAX[h]	MAX[ə]	$_{N}$*hC	$_{N}$*VCəCV#
a.	ahəp-e		*!				
b. ☞	ahp-e				*	*	
c.	ap-e			*!	*		

For Comparative Markedness, generating mutual-counterfeeding is not meaningfully different from generating two unrelated counterfeeding interactions. All that is needed is to rank two "old" markedness constraints high enough, respecting the counterfeeding ranking schema in (51). As in classic OT, the constraints can be freely ranked with respect to each other, and nothing prevents mutual-counterfeeding from being included in the factorial typology generated by the theory. For the serial rule-based theory, which accounts for counterfeeding through ordering, mutual-counterfeeding posed an ordering paradox. In Comparative Markedness there is no ordering, and no paradox arises. Overall, by relying on an input-based characterization of counterfeeding, Comparative Markedness fails to create an asymmetry between mutual-counterfeeding and attested cases of counterfeeding.

3.3 Sympathy Theory

3.3.1 A Sympathy analysis of opacity

Another attempt to deal with the opacity problem within parallel OT is Sympathy Theory (McCarthy 1999, 2003b), which introduces a new kind of faithfulness constraints that are defined not with respect to the input but with respect to some designated output candidate. That designated candidate often resembles what would have been the intermediate representation in a serial derivation. This allows Sympathy Theory to

[6]An additional ranking requirement is that each of the "old" markedness constraints must outrank both "new" markedness constraints.

try to simulate intermediate representations in a fully parallel derivation, and successfully deal with cases of both counterfeeding and counterbleeding opacity. As we will see, however, by generating opacity in parallel in this way, Sympathy Theory also generates unattested crucially-simultaneous interactions: mutual-counterbleeding, and under certain conditions also mutual-counterfeeding.

To see how Sympathy Theory works, consider again the counterbleeding problem for OT discussed in section 3.1.

(55) A failed Parallel-OT attempt to derive [ħaːkʲm-iːn]

		/ħaːkim-iːn/	*ki	*iCV	Ident[back]	Max
a.		ħaːkimiːn	*!	*		
b.		ħaːkʲimiːn		*!	*	
c.	☞	ħaːkmiːn				*
d.	☹	ħaːkʲmiːn			*!	*

The problem for classic OT was that there was no surface motivation to palatalize the /k/ given that the following /i/ has been deleted. McCarthy's proposal is that palatalization in the desired candidate (55d) can be enforced if this candidate is required to be faithful not only to the UR, but also to the candidate in which palatalization is motivated: candidate (55b), [ħaːkʲimiːn], in which the /i/ has not been deleted. The tableau in (56) shows that a constraint "Ident[back]$_{\text{[Cand b]}}$", which requires faithfulness in [back] to candidate (56b) rather than the UR, correctly rules out the problematic candidate (56c) when ranked above the regular faithfulness constraint Ident[back].

(56) A special faithfulness constraint eliminates the incorrect candidate [ħaːkm-iːn]

		/ħaːkim-iːn/	*ki	*iCV	**Ident[back]$_{\text{[Cand b]}}$**	Ident[back]	Max
a.		ħaːkimiːn	*!	*	*		
b.		ħaːkʲimiːn		*!		*	
c.		ħaːkmiːn			***!**		*
d.	☞	ħaːkʲmiːn				*	*

Of course, constraints like "Ident[back]$_{\text{[Cand b]}}$" are not available in classic OT and are a new addition to the theory. They come with an additional component that selects the designated candidate in a principled way. Informally, candidate (56b) can be selected as the special candidate for evaluation using its following property: it is the optimal candidate among the candidates that satisfy Max. The candidates that satisfy Max are the candidates in which /i/ has not been deleted: candidates (56a) and (56b). Among them, (56b) is optimal because it satisfies the highest ranked markedness constraint *ki whereas (56a) does not.

Formally, the Sympathy analysis requires the following ingredients:

(57) Sympathy: main ingredients

1. A *selector constraint F* (marked with ✯), required to be a faithfulness constraint.
2. A *sympathetic candidate* (marked with ❀$_F$), defined as the optimal candidate among the candidates that satisfy the selector constraint F.
3. A *sympathy constraint* (also marked with ❀$_F$), which requires faithfulness to the sympathetic candidate ❀$_F$ rather than the UR.
4. Assumption: sympathy constraints are ignored for the selection of the sympathetic candidate.

The final tableau, using Sympathy Theory's notation, is given in (58), where the selector constraint MAX is indicated by the symbol ✯, the sympathetic candidate it selects by ❀$_{\text{MAX}}$, so as its corresponding sympathetic constraint IDENT[back]$_{\text{MAX}}$.

(58) A sympathy analysis of counterbleeding

	/ħaːkim-iːn/	*ki	*iCV	❀ IDENT[back]$_{\text{MAX}}$	IDENT[back]	✯ MAX
a.	ħaːkimiːn	*!	*	*		
b. ❀$_{\text{MAX}}$	ħaːkʲimiːn		*!		*	
c.	ħaːkmiːn			*!		*
d. ☞	ħaːkʲmiːn				*	*

Overall, Sympathy Theory successfully generates simple cases of counterbleeding opacity in a fully parallel fashion, by simulating an intermediate representation that transfers its phonological properties to the output through faithfulness.

3.3.2 Mutual-counterbleeding

However, as we will now show, this way of generating counterbleeding opacity is too powerful, as it can also generate mutual-counterbleeding. First, as already noted by McCarthy (1999, section 7), the theory must allow for the possibility of a single language with multiple selector constraints, sympathetic candidates, and sympathy constraints, in order to account for multiple unrelated instances of opacity in the same language. The grammar keeps track of the relationship between selectors and their sympathetic candidates and constraints through the indexing presented above (e.g., ❀$_F$ is a sympathy constraint/candidate that corresponds to the selector constraint F).

Consider now again our example of mutual-counterbleeding from section 2.2.1, summarized in (59).

(59) Mutual-counterbleeding in a hypothetical language

a. Rules:
   - h-DELETION: h → ∅ / __ C
   - GLIDE VOCALIZATION: w → u / C __ $]_\sigma$

b. Mapping:
   - /gahw/ → [gau]

To generate mutual-counterbleeding within Sympathy, it is possible to simulate two intermediate representations that would transfer their properties to the output simultaneously: the candidate [gah-u] can transfer its final vowel [u], while [ga-w] can transfer its absence of [h].

This is shown in tableau (60). The markednsess constraints assumed are *hC, which penalizes a preconsonantal [h] (the trigger for h-deletion), and *Cw$]_\sigma$, which penalizes the glide [w] syllable-finally after a consonant (the trigger for glide vocalization). The basic faithfulness constraints are IDENT[voc] (short for the feature [vocalic]), DEP[h], and MAX[h].

(60) A Sympathy tableau for mutual-counterbleeding

	/gah-w/	*hC	*Cw$]_\sigma$	❀IDENT[voc]$_{\text{MAX[h]}}$	❀DEP[h]$_{\text{IDENT[voc]}}$	✯MAX[h]	✯IDENT[voc]
a.	gah-w	*!	*	*	*	✓	✓
b. ❀$_{\text{IDENT[voc]}}$	ga-w			*!		*	✓
c. ❀$_{\text{MAX[h]}}$	gah-u				*!	✓	*
d. ☞	ga-u					*	*

The first selector constraint is ⍟MAX[h], which selects candidate (60c), [gah-u], as its sympathetic candidate. This is because there are two candidates that satisfy ⍟MAX[h] – the faithful candidate (60a) and candidate (60c) – and among them candidate (60c) is more optimal as it satisfies both markedness constraints. The sympathetic constraint that corresponds to the selector ⍟MAX[h] is ❀IDENT[voc]$_{\text{MAX[h]}}$, which penalizes any glide-vowel alternations between [gah-u] and other candidates. Candidate (60b), [ga-w], has such an alternation and incurs a critical violation of ❀IDENT[voc]$_{\text{MAX[h]}}$. The second selector constraint is ⍟IDENT[voc], which selects candidate (60b), [ga-w], as its sympathetic candidate, following a similar reasoning as before. In this case, the sympathetic constraint that corresponds to the selector is ❀DEP[h]$_{\text{IDENT[voc]}}$, which assigns a critical violation to candidate (60c), [gah-u], because it has an extraneous [h] compared to [ga-w]. Ultimately, the optimal candidate is candidate (60d), [ga-u], which preserves both the [vocalic] feature of [gah-u] and the h-lessness of [ga-w], and mutual-counterbleeding is generated.

The problem for Sympathy Theory is that it tries to simulate intermediate representations without serialism. The peculiar consequence of this theory is the possible coexistence of two "intermediate representations" that differ from the UR in one minimal way and can affect the output (in the example above, the UR was /gah-w/ and the "intermediate representations" under Sympathy were [ga-w], to which h-DELETION applied, and [gah-u], to which GLIDE VOCALIZATION applied). In this way, Sympathy avoids the ordering paradox that mutual-counterbleeding poses for serial theories of opacity, and ends up overgenerating typologically.

3.3.3 Mutual-counterfeeding

In addition to counterbleeding, Sympathy can also generate cases of underapplication opacity resulting from simple counterfeeding interactions. In what follows, we will show that Sympathy can also generate mutual-counterfeeding interactions, albeit in specific circumstances in which one of the interacting processes can be described as a "Do-Something-Except-When" process, using Baković's (2011) terminology.

Let us first see how Sympathy Theory accounts for simple counterfeeding interactions, using the Catalan example once again. As tableau (61) shows, it is possible to designate the faithful candidate (61a) as the sympathetic candidate by choosing MAX[T] as the selector, because candidate (61a) is trivially the optimal candidate in which /t/ surfaces. The sympathetic constraint ❀MAX[N]$_{\text{MAX[T]}}$, when ranked above *N#, requires that the nasal of the faithful candidate should surface even when word-final. This constraint eliminates the problematic candidate (61c), [kəle], and the correct candidate (61b) wins.

(61)

	/kəlent/	*NT#	⍟MAX[T]	❀MAX[N]$_{\text{MAX[T]}}$	*N#	MAX[N]
a. ❀$_{\text{MAX[T]}}$	kəlent	*!				
b. ☞	kəlen		*		*	
c.	kəle		*	*!		*

Tableau (62) shows that this ranking still enforces nasal deletion in isolation. When the input does not contain a stop, the selector constraint ⍟MAX[T] is irrelevant, and the sympathy mechanism is not at play. The regular ranking of markedness over input-output faithfulness *N# ≫ MAX[N] correctly causes deletion.

(62)

	/kuzin/	*NT#	⍟MAX[T]	❀MAX[N]$_{\text{MAX[T]}}$	*N#	MAX[N]
a.	kuzin				*!	
b. ❀$_{\text{MAX[T]}}$ ☞	kuzi					*

Consider now mutual-counterfeeding. For reasons explained below, here we use Wolf's (2011) original example of mutual-counterfeeding, which we have slightly modified in section 2.2.2. The difference is that in Wolf's example, schwa syncope does not apply in a two-sided open-syllable environment. Rather, it is described as follows:

(63) SCHWA SYNCOPE (revised)
Generalization: Delete schwa unless the outcome is a triconsonantal sequence.[7]
Ranking: *CCC ≫ *ə

Other than the conditions of application of schwa syncope, the mutual-counterfeeding example remains the same. There is a process of h-DELETION, and the interaction leads to the same mappings as before: whenever SCHWA SYNCOPE creates the context for h-DELETION, the latter does not apply, and vice versa (64).

(64) Mappings for mutual-counterfeeding

- /gah-rəmu/ → [ga-rəmu] (SCHWA SYNCOPE underapplies)
- /ahəp-e/ → [ahp-e] (h-DELETION underapplies)

Sympathy can generate these mappings as if they were two independent counterfeeding interactions, as shown in tableaux (65)-(66). There are two selector constraints: ✯MAX[h] and ✯MAX[ə], and two corresponding sympathetic constraints, ❀MAX[ə]$_{\text{MAX[h]}}$ and ❀MAX[h]$_{\text{MAX[ə]}}$, respectively.

(65) SCHWA SYNCOPE counterfeeds h-DELETION: /ahəp-e/ → [ahp-e]

	/ahəp-e/	❀MAX[h]$_{\text{MAX[ə]}}$	❀MAX[ə]$_{\text{MAX[h]}}$	*CCC	*ə	*hC	✯MAX[h]	✯MAX[ə]
a. ❀$_{\text{MAX[ə]}}$	ahəp-e				*!			
b. ❀$_{\text{MAX[h]}}$ ☞	ahp-e					*		*
c.	ap-e	*!					*	*

In this tableau, the selector constraint ✯MAX[ə] selects the faithful candidate (66a) as its sympathetic candidate. The corresponding sympathy constraint ❀MAX[h]$_{\text{MAX[ə]}}$ then penalizes h-deletion with respect to this candidate. The main role of this constraint is to eliminate the transparent candidate (65c), [ap-e], in which SCHWA SYNCOPE feeds h-DELETION. Between the faithful candidate (65a) and the desired candidate (65b), candidate (65b) is more optimal because it satisfies the markedness constraint *ə while the faithful candidate violates it.

(66) h-DELETION counterfeeds SCHWA SYNCOPE: /gah-rəmu/ → [ga-rəmu]

	/gah-rəmu/	❀MAX[h]$_{\text{MAX[ə]}}$	❀MAX[ə]$_{\text{MAX[h]}}$	*CCC	*ə	*hC	✯MAX[h]	✯MAX[ə]
a. ❀$_{\text{MAX[h]}}$	gah-rəmu				*	*!		
b. ❀$_{\text{MAX[ə]}}$ ☞	ga-rəmu				*		*	
c.	ga-rmu		*!				*	*
d.	gah-rmu		*!	*		*		*

In this tableau, the selector constraint ✯MAX[h] is the one that selects the faithful candidate (66a) as its sympathetic candidate. The corresponding sympathy constraint ❀MAX[ə]$_{\text{MAX[h]}}$ penalizes schwa deletion with respect to this candidate, and eliminates the transparent candidate (66c), [ga-rmu], in which h-DELETION feeds SCHWA SYNCOPE (candidate (66d) would have lost even without the sympathy constraint). Between the faithful candidate (66a) and the desired candidate (66b), candidate (66b) is more optimal because the two candidates are tied with respect to the markedness constraint *ə, but candidate (66b) satisfies the markedness constraint *hC while the faithful candidate violates it. Since Sympathy can generate both counterfeeding interactions, it can generate mutual-counterfeeding.

Sympathy Theory's ability to generate mutual-counterfeeding in this example crucially relies on a decomposition of the markedness constraints that trigger one of the processes. Specifically, the markedness

[7] Wolf offers the following SPE-style rule to capture this generalization: ə → ∅ / {V, #}(C) __ (C){V, #}.

constraints that trigger Schwa syncope are the general markedness constraint *ə, and the constraint *CCC which outranks it and blocks syncope in specific circumstances. We can think of syncope in this case as a "Do-Something-Except-When" process (Baković 2011), a process obtained from a combination of two markedness constraints: a "trigger" markedness constraint, and a higher-ranking "blocker" constraint that prevents the trigger from taking effect in case the result is too marked. In tableau (66), this decomposition played a crucial role because the faithful candidate (66a) violated *ə even though it does not meet the context for syncope. Without this violation, *hC would have had to outrank *ə in order for the desired candidate (66b) to win, in contradiction with the opposite ranking required in tableau (65).[8]

Our rather nuanced conclusion in this section is therefore that Sympathy Theory can generate mutual-counterfeeding in specific circumstances, if the required decomposition of markedness is possible. Together with its general ability to generate mutual-counterbleeding interactions, our general conclusion is that Sympathy generates attested opaque interactions in a parallel fashion at the expense of overgenerating unattested interactions, and does not capture the No-Simultaneity Universal.

3.4 Stratal OT

Stratal OT is another extension to OT that aims to deal with opacity by incorporating serialism into OT directly (Bermúdez-Otero 1999, 2018; Kiparsky 2000, 2015). As we will see, by using serialism to derive opacity, Stratal OT can generate many attested cases of opacity, while also successfully excluding crucially-simultaneous interactions.

In Stratal OT there are multiple constraint rankings, each corresponding to a morphological or a syntactic domain, such as the stem , the word, and the phrase. Within each domain – also called stratum – the constraint ranking operates in a fully parallel fashion, as a classic Parallel-OT grammar. However, the different strata interact serially, starting from the smallest domain and going outwards, as schematized in (67).

(67) Schematic illustration of phonological computation in Stratal OT

$$[_{\varphi_3} \cdots [_{\varphi_2} \cdots [_{\varphi_1} \cdots] \cdots] \cdots]$$

φ_1: First application of a Parallel-OT grammar
φ_2: Second application of a Parallel-OT grammar
φ_3: Third application of a Parallel-OT grammar
… where each φ_i denotes a morphological or a syntactic domain.

With multiple strata that interact serially, Stratal OT can generate many kinds of opacity. Consider, once again, the counterbleeding interaction between Palatalization and Syncope in Bedouin Hijazi Arabic from section 1. The two processes can be assigned to two different strata, according to the rankings in (68). The derivation of the UR /ħaːkim-iːn/ is given in tableaux (69)–(70). In the first stratum (69) palatalization applies, followed by syncope in the second stratum (70).

[8]Concretely, suppose that we merge the two markedness constraints that account for syncope, say by replacing them with the single markedness constraint *VCəCV. Instead of tableau (66), we would have gotten the tableau below, in which the fully faithful candidate incorrectly wins. Fixing this problem by ranking *VCəCV lower than *hC, would have created a ranking paradox with tableau (65), which requires the very opposite ranking.

	/gah-rəmu/	❀Max[h]$_{\mathrm{Max[ə]}}$	❀Max[ə]$_{\mathrm{Max[h]}}$	*VCəCV	*hC	✯Max[h]	✯Max[ə]
a. ❀$_{\mathrm{Max[h]}}$ ☞	gah-rəmu				*		
b. ❀$_{\mathrm{Max[ə]}}$ ☹	ga-rəmu			*!		*	
c.	ga-rmu		*!			*	*
d.	gah-rmu		*!		*		*

(68) A Stratal OT analysis of counterbleeding: PALATALIZATION before SYNCOPE

a. Stratum I: *ki, MAX ≫ IDENT[back], *iCV
(PALATALIZATION active, SYNCOPE inactive)

b. Stratum II: *iCV ≫ MAX
(SYNCOPE active)

(69) Stratum I: Palatalization applies

		/ħaːkim-iːn/	*ki	MAX	*iCV	IDENT[back]
a.		ħaːkimiːn	*!		*	
b.	☞	ħaːkʲimiːn			*	*
c.		ħaːkmiːn		*!		
d.		ħaːkʲmiːn		*!		*

(70) Stratum II: Syncope applies

		/ħaːkʲimiːn/	*ki	*iCV	MAX	IDENT[back]
a.		ħaːkʲimiːn		*!		
b.	☞	ħaːkʲmiːn			*	

Given that each Parallel-OT grammar faces an opacity challenge, Stratal OT can generate opacity across strata but not within strata (whether intra-stratal cases of opacity exist that pose a problem for the theory is a matter of ongoing debate, which we will not discuss in this paper). Importantly, this means that Stratal OT can generate opacity only serially, and is unable to generate crucially-simultaneous interactions.

We will illustrate this point using the example of mutual-counterbleeding between h-deletion and schwa syncope from section 2.2.1, which includes derivations like /gah-w/→ [gau]. A first, trivial observation is that a single Parallel-OT grammar cannot generate the doubly-opaque interaction between the two processes, as shown in (71). The problem, as in simple counterbleeding, is that a single change suffices to satisfy both markedness constraints, so the desired candidate with overapplication – candidate (71d) – is harmonically bounded (differently from regular counterbleeding, here it is harmonically bounded by two different candidates).

(71) Mutual-counterbleeding cannot be generated by a classic Parallel-OT grammar

		/gah-w/	*hC	$*\text{Cw}]_\sigma$	MAX[h]	IDENT[voc]
a.		gah-w	*	*		
b.		ga-w			*	
c.		gah-u				*
d.	☹	ga-u			*	*

Stratal OT can try to generate this opaque interaction by assigning h-deletion and glide vocalization to different strata, but no such assignment will work, for the same reason that serial rule application failed on mutual-counterbleeding. If h-deletion applies in an earlier stratum, glide vocalization will fail to apply in the later stratum, and the final output would be the incorrect *[gaw]. The opposite assignment of glide vocalization to an earlier stratum and h-deletion to a later stratum will result in the incorrect output *[gahu].

To see this concretely, consider the first option, of h-deletion preceding glide vocalization. One possible grammar is given in (72): in the first stratum, the ranking is such that h-deletion is active and glide vocalization is inactive, and the ranking of the second stratum guarantees that glide vocalization will apply but not h-deletion.

(72) A Stratal OT attempt at mutual-counterbleeding: h-deletion before glide vocalization

a. Stratum I: *hC, IDENT[voc] ≫ MAX[h], $^*\mathrm{Cw}]_\sigma$
(h-deletion active, glide vocalization inactive)

b. Stratum II: $^*\mathrm{Cw}]_\sigma$ ≫ MAX[h] ≫ IDENT[voc], *hC
(glide vocalization active, h-deletion inactive)

As can be seen in tableaux (73)-(74), the output of the first stratum under this ranking is [gaw], which serves as the input to the second stratum. Since [gaw] already satisfies both markedness constraints, there is simply no motivation to apply glide vocalization in the second stratum. The correct output candidate [gau] incurs a spurious faithfulness violation and loses to the faithful candidate at that stratum, *[gaw].

(73) Stratum I: h-deletion applies

/gah-w/	*hC	IDENT[voc]	MAX[h]	$^*\mathrm{Cw}]_\sigma$
a. gah-w	*!			*
b. ☞ ga-w			*	
c. gah-u		*!		
d. ga-u		*!	*	

(74) Stratum II: no motivation for glide vocalization

/ga-w/	$^*\mathrm{Cw}]_\sigma$	MAX[h]	IDENT[voc]	*hC
a. ☞ ga-w				
b. ☹ ga-u			*!	

Similarly, if the ranking is changed such that glide vocalization is active in an earlier stratum and h-deletion in a later stratum, the output of the first stratum would be *[gahu], and there will be no surface motivation to delete the /h/ in the second stratum. The desired output [gau] will therefore incur a spurious violation of MAX[h] and lose to the incorrect *[gahu], which will be the final output.

Overall, since serialism is the only mechanism for deriving counterbleeding (and opacity more generally) in Stratal OT, mutual-counterbleeding poses an ordering paradox for the theory. A similar point can be made about mutual-counterfeeding, which poses another ordering paradox for Stratal OT. Since these interactions are unattested, Stratal OT is properly restrictive in this respect.

3.5 Interim conclusion

To conclude the section about OT, we have evaluated three versions of OT with respect to the NO-SIMULTANEITY UNIVERSAL: Comparative Markedness, Sympathy Theory, and Stratal OT. All three versions improve OT's ability to account for opaque interactions, but they do so in different ways. Comparative Markedness and Sympathy Theory are extensions that maintain the full parallelism of classic OT, and do not create the observed asymmetry between attested and unattested opaque interactions. In contrast, Stratal OT proposes to deal with opacity through a serial mechanism, and the observed asymmetry is correctly derived. This result suggests that even within a constraint-based phonological framework, serialism is an important component of a successful theory of opacity.

4 Input Strictly Local Maps

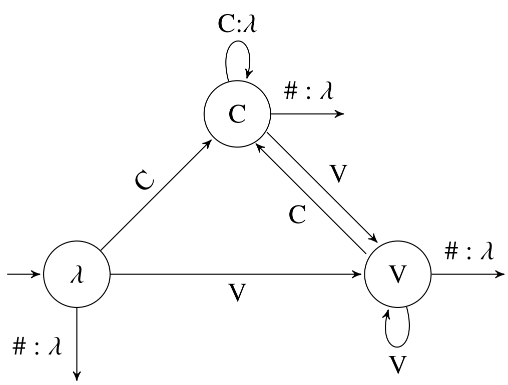
4.1 Background

The computational phonology literature has recognized a class of functions called Input Strictly Local (ISL) Maps (Chandlee 2014, Chandlee and Heinz 2018). ISL maps are input-output maps that satisfy the property that the output of each symbol in the input string depends only on the current input symbol and the preceding $k-1$ input symbols, for some fixed positive integer k. Consider, for example, the map defined by the SPE rule in (75), which deletes a consonant after another consonant.

(75) Example: SPE rule corresponding to an ISL map for $k = 2$
C $\rightarrow \emptyset$ / C __

Assuming that the input string is read from left to right, the decision whether to delete a consonant depends only on the current input symbol (whether it is a consonant) and on the immediately preceding symbol (whether it is also a consonant). The map defined by the rule in (75) is therefore ISL for $k = 2$. As shown by Chandlee (2014), ISL maps correspond to a subclass of deterministic finite-state transducers with the following properties: the states correspond to all possible sequences up to length $k-1$ over the transducer's alphabet, and the transitions ensure that each state represents the most recently read $k-1$ input symbols. A finite-state transducer that defines the same map as the SPE rule in (75) is given in (76).[9]

(76) ISL transducer ($k = 2$) for C $\rightarrow \emptyset$ / C __, with $\Sigma = \{C, V\}$



We follow Chandlee and Heinz's 2018 notation and conventions for writing ISL transducers: the empty string is represented by the symbol λ, each transition between states indicates the input and output separated by a colon, except when the input and output are identical, in which case they are indicated by the same symbol (without a colon). The initial state is indicated by an incoming arrow, and each state has an outgoing arrow indicating the output for the end-of-string marker #. In the ISL transducer in (76), the alphabet Σ contains two symbols: "C" and "V". Each input consonant always leads to state "C" and each vowel to state "V". The only input-output change occurs when a consonant is read in state "C", as indicated by the "C:λ" label on the looping transition, meaning that a consonant will be deleted whenever it follows another input consonant. This transducer is ISL for $k = 2$ because every input of length $k - 1 = 1$ always leads to the same state.

[9] Specifically, the map corresponds to the simultaneous or right-to-left application of the rule.

Chandlee (2014) claims that ISL is a computationally interesting property because it captures the notion of locality in phonology, in the sense that attested local phonological process are ISL maps (see also Chandlee and Heinz 2018). Chandlee et al. (2018) have extended this claim from individual processes to process interaction. They go over a range of attested opaque interactions and show that each could be modeled by an ISL transducer. ISL transducers can capture interactions between multiple phonological processes because they are not limited to a single change, as we will see below. Given their result, Chandlee et al. (2018) conclude that the ISL class provides a properly restrictive characterization of process interaction, and they compare it to rule-based phonology and OT in terms of generative power and empirical predictions.

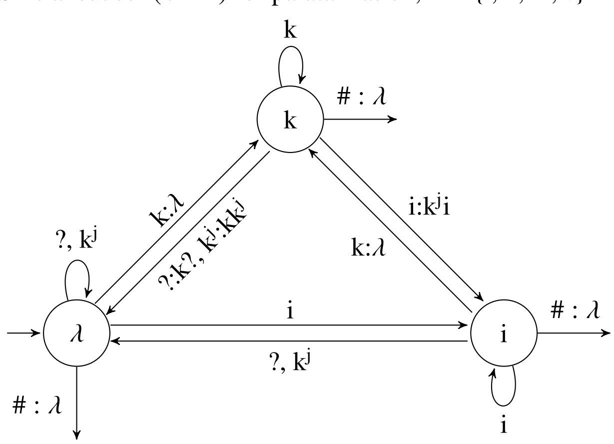
Our new contribution in this paper is to show that while attested opaque interactions are ISL, the unattested crucially-simultaneous interactions are also ISL, and can be captured by ISL transducers as easily as attested interactions. The ISL class therefore fails to capture the No-Simultaneity Universal.

Before turning to ISL characterizations of process interactions, it will be useful to understand how maps with a right context (rather than a left context) can be characterized as ISL. Even though ISL maps are defined with respect to *preceding* symbols, maps in which changes are made with respect to a following context of bounded length are also ISL. Consider, for example, the palatalization rule from the introduction, repeated in (77):

(77) Palatalization, ISL for $k = 2$
$k \rightarrow k^j$ / __ i

Once a /k/ in the input string is read, it is not yet known whether it should be palatalized, because palatalization depends on the symbol that comes after. The map defined by this rule can nevertheless be generated by an ISL transducer, by "postponing" the decision to palatalize an input /k/ until the next input symbol is read. This can be done by deleting any /k/, and restoring it as either [k] or [k^j] depending on whether the following symbol is /i/, as shown in (78).

(78) ISL transducer ($k = 2$) for palatalization, $\Sigma = \{i, k, k^j, ?\}$



In this transducer, the state "k" is reached when reading an input /k/, which gets deleted, as indicated by the "k:λ" label on the transitions to this state from states "λ" and "i". If the input at state "k" is /i/, which should trigger palatalization, the output is the sequence "k^ji", which brings back the deleted /k/ as palatalized. This is indicated by the label "i:k^ji" on the transition from state "k" to state "i". However, if the next input symbol at state "k" is any symbol other than /i/ (or /k/) – indicated by the question mark "?" – the deleted /k/ is brought back, as indicated by the label "?:k?" on the transition from state "k" to state "λ". The remaining transitions are self-explanatory. Overall, since each symbol in the alphabet always leads to the same state, this transducer is ISL for k = 2.

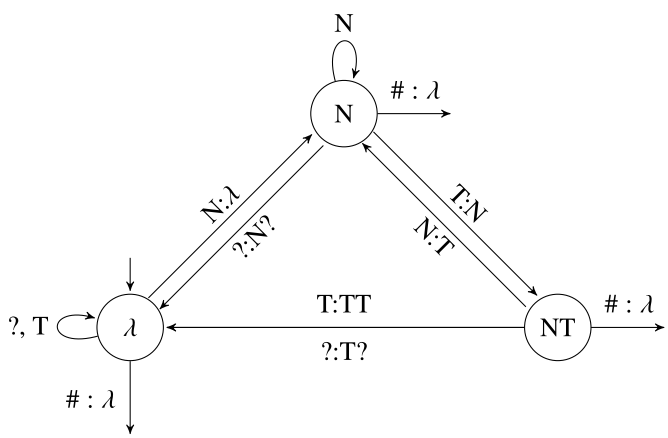
We now turn to introduce Chandlee et al.'s 2018 claim that simple opaque interactions are also ISL. As an example of an ISL transducer for counterfeeding opacity, consider again the Catalan example from section 2.1, summarized in (79). The ISL transducer generating this language is given in (80).

(79) Counterfeeding in Catalan

a. Rules:
- NASAL DELETION: N $\rightarrow \emptyset$ / __ #
- CLUSTER SIMPLIFICATION: T $\rightarrow \emptyset$ / N __ #

b. Mappings:
- /kuzin/ → [kuzi] (NASAL DELETION applies)
- /kəlent/ → [kəlen] (CLUSTER SIMPLIFICATION applies, NASAL DELETION underapplies)

(80) ISL transducer (k = 3) for Catalan counterfeeding, with Σ = {N, T, ?}



The transition tables in (81)-(82) illustrate how the counterfeeding interaction is derived. As long as a non-nasal input symbol is read, the transducer stays in the initial state and makes no changes to the string. Once an input nasal /n/ is reached, it is deleted while transitioning to state "N". Then, if the input string contains no further symbols, as in /kuzin/, the computation ends without restoring the deleted nasal. This is how final-nasal deletion applies without any interaction.

(81) Transition table for /kuzin/ → [kuzi]

input		k		u		z		i		n		#	
state	λ	→	λ	→	λ	→	λ	→	λ	→	N	→	END
output		k		u		z		i		λ		λ	

In contrast, when the input nasal is followed by any alphabet symbol, the deleted nasal is restored. If the following symbol is a stop, as in /kəlent/, the nasal is restored through a transition from state "N" to state

"NT", which deletes the stop in anticipation of the following context. When no symbol follows the stop, the deleted stop is not restored and the computation ends with a nasal-final output.

(82) Transition table for /kəlent/ → [kəlen]

input		k		ə		l		e		n		t		#	
state	λ	→	λ	→	λ	→	λ	→	λ	→	N	→	NT	→	END
output		k		ə		l		e		λ		n		λ	

The opaque interaction between Nasal deletion and Cluster simplification is derived without any difficulties (using an ISL transducer where $k = 3$), because the input-output changes of an ISL transducer depend exclusively on the input string. In particular, a nasal is deleted (and not restored) precisely when it is the final symbol of the input. In other words, this is an input-based characterization of opacity, which can easily account for simple cases of counterfeeding interactions, as we have seen above. The ISL account is also a parallel account, in the sense that the input string is mapped to an output string directly; there are no intermediate levels of representation, and no output can serve as an input to further operations. Simple counterbleeding interactions can similarly be generated by ISL transducers in a straightforward manner.

4.2 Crucially-simultaneous interactions are ISL

We now turn to show that unattested (local) crucially-simultaneous interactions are also ISL, and that they can be generated by ISL transducers as easily as attested opaque interactions. Here we will use particularly simplified examples of crucially-simultaneous interactions, to keep the transducers simple.

First, for mutual-counterfeeding, consider the two rules and input-output mappings given in (83).

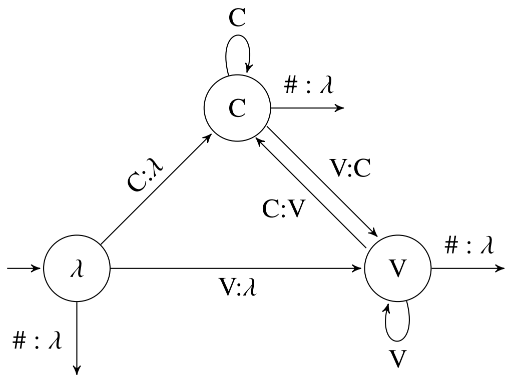
(83) Mutual-counterfeeding: simple example

a. Rules:
- C → ∅ / __ #
- V → ∅ / __ #

b. Mappings:
- CVC → CV
- CVCV → CVC
- [...]

Here there are two rules, one that deletes consonants word-finally and one that deletes vowels word-finally, each corresponding to an ISL map with $k = 2$. Whenever each of these rules creates the context for the other rule, the other rule fails to apply. A simple ISL transducer generating this interaction, still with $k = 2$, is given in (84). Representative transition tables are given in (85)–(86).

(84) ISL transducer for mutual counterfeeding, $k = 2$



(85) Transition table for /CVC/ → [CV]

input		C		V		C		#	
state	λ	→	C	→	V	→	C	→	END
output		λ		C		V		λ	

(86) Transition table for /CVCV/ → [CVC]

input		C		V		C		V		#	
state	λ	→	C	→	V	→	C	→	V	→	END
output		λ		C		V		C		λ	

The reason that mutual-counterfeeding can be generated without a problem using an ISL transducer is that ISL maps depend on the input and follow an input-based characterization of opacity, as mentioned before. Whatever context is present in the input determines whether an input-output change should take place, a property that is perfectly compatible with mutual-counterfeeding interactions. On the assumption that mutual-counterfeeding is unattested in natural language, this is a problem for the ISL characterization of the typology of process interactions.

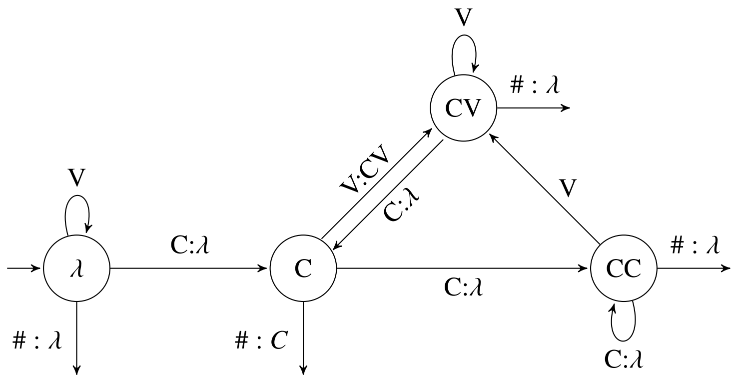
A similar reasoning applies to mutual-counterbleeding. A simple example of the interaction is given in (87), followed by an ISL transducer (88) and two transition tables (89)–(90).

(87) Mutual-counterbleeding: simple example

a. Rules:
- C → ∅ / __ C
- C → ∅ / C __

b. Mappings:
- VCCV → VV
- VCV → VCV

(88) ISL transducer for mutual-counterbleeding, $k = 3$



(89) Transition table for /VCV/ → [VCV]

input		V		C		V		#	
state	λ	→	λ	→	C	→	CV	→	END
output		V		λ		CV		λ	

(90) Transition table for /VCCV/ → [VV]

input		V		C		C		V		#	
state	λ	→	λ	→	C	→	CC	→	CV	→	END
output		V		λ		λ		V		λ	

Again, generating a mutual-counterbleeding interaction with an ISL transducer is straightforward, as input-output changes take place whenever the right context is met in the input. The fact that each change can remove the context for the other is insignificant from the ISL perspective. As with mutual-counterfeeding, the unattestedness of mutual-counterbleeding is surprising under the ISL characterization of process interaction. The problem is not with ISL as a computational property in general. It may still be a useful property for understanding the complexity of single processes. However, as a model of opaque process interaction, it fails to create an asymmetry between attested and unattested opaque interactions, because it makes multiple changes in parallel without intermediate representations, depending solely on the input.

5 Open issues

In this section, we discuss a few issues that we leave unresolved in this paper, and that we believe future research could shed light on. We first introduce two additional crucially-simultaneous interactions that to our knowledge have not been previously discussed in the literature. We then discuss the open question of the role of abstractness in phonology, and how answering that question could affect the conclusions that can be drawn from the typology of process interactions. Finally, we comment on the status of crucially-simultaneous interactions that involve more than two processes.

5.1 Additional interactions not discussed in the literature

5.1.1 Mutual-countershifting

In addition to feeding and bleeding, the literature has recognized a third kind of a basic rule interaction that Rasin (2022) calls *shifting*. In shifting interactions, a rule does not feed or bleed the next rule to apply, but rather causes it to apply in a different way. An abstract example involves the two rules in (91). The first rule deletes a vowel word-finally and the second assigns penultimate stress. The two possible interactions between these rules are shown in (92).

(91) Two rules that can stand in shifting and countershifting relations:

1. $V \rightarrow \emptyset / \_\_ \ \#$
2. Assign penultimate stress

(92) Example: shifting and countershifting interactions

a. Shifting:

Input	/CVCVCV/
Deletion	CVCVC
Stress	CV́CVC
Output	[CV́CVC]

b. Countershifting:

Input	/CVCVCV/
Stress	CVCV́CV
Deletion	CVCV́C
Output	[CVCV́C]

In the first derivation, Deletion precedes Stress. As a result of vowel deletion, stress is assigned to the first syllable of the word. Here, Deletion does not feed or bleed Stress, because stress would have been assigned regardless of the application of Deletion. Nevertheless, the application of Deletion changes location to which stress is assigned from the second syllable to the first syllable of the word, an instance of shifting. If the rules had been applied in the opposite order, as shown in the second derivation, Stress would have been opaque, as the final syllable of the word would have been stressed on the surface. This is a *countershifting* interaction, involving what Baković and Blumenfeld (2024) refer to as misapplication opacity (rather than under- or over-application). Shifting and countershifting interactions are attested in a variety of languages (Rasin 2022; Baković and Blumenfeld 2024; Pruitt 2023).

The mutual-opaque version of countershifting, which is parallel to mutual-counterfeeding and mutual-counterbleeding, involves two rules that can countershift each other. An example is the two rules in (93). The first rule reduplicates the initial CV of the root, and the second rule assigns high tone to the initial syllable of the word.

(93) Two rules that can countershift each other:

1. CV-Reduplication: $C_1V_2C_3V_4 \rightarrow C_1V_2\text{-}C_1V_2C_3V_4$
2. Initial high tone

Applying these rules serially to a bisyllabic input results in a countershifting interaction regardless of the ordering, as shown in (94).

(94) Both orders of application lead to countershifting

a. Countershifting:

Input	/gabu/
Reduplication	ga-gabu
Initial tone	gá-gabu
Output	[gá-gabu]

b. Countershifting:

Input	/gabu/
Initial tone	gábu
Reduplication	gágábu
Output	[gágábu]

In the derivation in (94a), reduplication applies before initial tone assignment. The result is a mismatch between the reduplicated phonological material and the root, a case of misapplication opacity (this discussion assumes that reduplication in this hypothetical language copies tones, as is the case in Chichewa; Hyman and Mtenje 1999). In the reverse order of application, as shown in (94b), initial tone is opaque because it assigned a high tone to a non-initial syllable, again an instance of misapplication. In contrast, if the rules apply simultaneously to the input rather than serially, a different output is derived, as shown in (95).

(95) Mutual-countershifting resulting from simultaneous application

Input	/gabu/
Initial tone, Red	gagábu
Output	[gagábu]

Here, the rules apply to the input together, and both end up being opaque, because a non-initial syllable has received a high tone and there is a mismatch between the reduplicant and the root.

We are unaware of attested cases of mutual-countershifting interactions, but at present it is unclear how meaningful this observation is. Countershifting and misapplication opacity have only recently entered the literature on process interactions as new categories, and more work should be done to understand the range of processes that can participate in such interactions. With such an understanding, it would be possible to look for potential cases systematically.

5.1.2 *Vipratiṣedha*-resolution

Any definition of simultaneous rule application should address the situation where two rules that apply simultaneously make contradictory rewrite statements. In the Sanskrit literature, this situation is known as *vipratiṣedha*. An example of two conflicting rules and a relevant input is given in (96).

(96) Example: *Vipratiṣedha*

- $\text{k} \rightarrow \text{k}^{\text{j}}$ / __ i
- k → tʃ / i __
- Input: /iki/

The context for both rules is met in the input, but the first rule requires the underlying /k/ to change into [k^{j}] while the second requires [tʃ] on the surface. How does simultaneous application resolve the conflict? One possibility is to define a meta-rule that gives priority to one of the conflicting rules in some principled manner (see Rajpopat 2023 for a historical overview of such ideas from the Sanskrit literature). A more straightforward alternative – already mentioned in the introduction – is to leave simultaneous application undefined in the case of rule conflict. This interpretation of simultaneous rule application predicts a special kind of rule blocking: two rules apply regularly when their contexts are met separately from one another, but when they interact, they are both blocked. This hypothetical interaction is illustrated by the mappings in (97).

(97) *Vipratiṣedha*-blocking (hypothetical example)

- /ki/ → [k^ji]
- /ik/ → [itʃ]
- /iki/ → [iki]

This is a crucially-simultaneous interaction because no conflict arises under serial rule application, where the rule that is ordered first would win (resulting in a *mutual-bleeding on focus* interaction). To our knowledge, such cases of blocking have not been reported in the literature, though here again a more systematic survey of rule interaction is in order.

5.2 Abstractness

The abstractness of phonological representations has been a matter of debate at least since Kiparsky's 1968 paper "How abstract is phonology?". After several decades of research, it remains a central open question in the field (Heinz 2015, for example, has presented it as a "Hilbert problem for phonology" for the 21st century). Answering the abstractness question can affect the kinds of conclusions that can be drawn from comparing theories using the typology of process interaction. To explain why, consider again the mutual-counterbleeding mapping /gah-w/ → [gau] from section 2.2.1 involving h-DELETION and GLIDE VOCAILIZATION. To generate this mapping, a serial rule-based grammar could be supplemented with four rules in addition to h-DELETION and GLIDE VOCAILIZATION, as shown in (98).

(98)

UR	/gah-t/	/dar-w/	/gah-w/
$\emptyset \rightarrow$ X / h __ w $]_\sigma$	-	-	gahXw
h $\rightarrow \emptyset$ / __ X	-	-	gaXw
w $\rightarrow$ u / X __	-	-	gaXu
X $\rightarrow \emptyset$ / __	-	-	gau
h-DELETION	ga-t	-	-
GLIDE VOCALIZATION	-	dar-u	-
SR	[ga-t]	[dar-u]	[gau]

The derivation /gah-w/ → [gau] in (98) begins with a special initial rule that inserts the abstract symbol "X" precisely where the context for both h-DELETION and GLIDE VOCALIZATION is met. This X triggers the application of two other rules, which are special versions of h-DELETION and GLIDE VOCALIZATION that apply in the presence of X rather than in the contexts in which the rules apply in isolation. After the two changes are made, a fourth rule removes every X from the derivation, and the regular versions of the rules get to apply. The sequence of rules in (98) successfully mimics the simultaneous application of h-DELETION and GLIDE VOCALIZATION, but it does so using a highly abstract Duke-of-York derivation (Pullum 1976), involving an extended intermediate alphabet with a symbol that is absent from both URs and SRs and is inserted only to trigger some changes and disappear.

As mentioned, the existence of this and other abstract derivations has been debated in the generative literature, for reasons that have nothing to do with the issue of crucially-simultaneous interactions. Positions in the literature range from grammatical conditions that ban such derivations completely, to conjectures that the degree of abstractness manifested in these derivations is too difficult to learn in the course of language acquisition. Here we will have nothing to contribute to this debate. What matters for the purposes of this paper is the observation that even if serial rule application can in principle generate input-output mappings that correspond to crucially-simultaneous interactions, it does so in indirect, convoluted ways. Importantly, serial rule application creates a clear asymmetry between attested phonological interactions and the unattested crucially-simultaneous interactions: the attested interactions can be generated straightforwardly, while the unattested interactions require complex abstract derivations, which may be unavailable completely or pose a learning challenge because of their abstractness (the same point holds for Stratal OT). The parallel theories discussed in this paper do not create the same asymmetry, as they can generate the unattested interactions as easily as the attested interactions. The absence of crucially-simultaneous interactions therefore remains helpful in distinguishing between theories of phonology, but what remains open is whether this absence is better explained by a categorical exclusion from the space of possible grammars, or by a softer ban, such as a difficulty in the learning process caused by some theories. Until progress on the abstractness questions is made, we believe that it is better to speak of a typological asymmetry created by a theory, as we have done in multiple places throughout this paper, rather than talking about non-generation categorically.

5.3 Beyond pairwise rule interactions

This paper follows much of the literature on phonological rule interaction in focusing on pairwise rule interactions (examples involving three rules, which are exceptions to this, are *Duke-of-York* derivations, *rule sandwiching* (Bye 2002), and *surface-true counterfeeding* (Baković 2011)). But crucially-simultaneous interactions can arise between more than two rules. Consider, as an abstract example, a language where /A/ becomes [B] in the environment X __ Y, /B/ becomes [C] in the same environment, and /C/ becomes [A] (again, in the same environment). This interaction, sometimes called a *circular chain shift* in the literature (Moreton 2003), can be generated by simultaneous rule application using three straightforward rules:

(99) Rules for a circular chain shift

- A → B / X __ Y
- B → C / X __ Y
- C → A / X __ Y

For serial rule application, this example poses an ordering paradox, because no matter how the rules are ordered, an incorrect output is derived. For example, if the ordering of the rules follows their order of presentation in (99), /B/ would incorrectly surface as [A] rather than [C] in the relevant environment, because the third rule would apply to the output of the second one. Circular chain shifts therefore provide potential candidate for a crucially-simultaneous interaction that goes beyond pairwise rule interaction. However, Moreton (2003), Lubowicz (2011), and Papillon (2017) have argued that there are no convincing cases of circular chain shifts that are attested (see also Wolf 2011 for relevant discussion), making the empirical picture consistent with No Simultaneity. Further research could tell whether any additional kinds of crucially-simultaneous interactions that necessitate more than two rules are possible, and whether they are attested.

6 Conclusion

We proposed a new generalization called the No-Simultaneity Universal, which groups together three typological generalizations previously made in the literature regarding unattested opaque rule interactions, stated again in (100).

(100) NO-SIMULTANEITY UNIVERSAL (final version)
Crucially-simultaneous interactions (between two applications of the same rule or between different rules) are unattested in natural language.

We suggested that this generalization can be used as a test case for competing phonological theories, and showed that some theories generate unattested interactions as easily as they generate attested interactions, and therefore do not explain the observed typological asymmetry. This was true for theories in three different phonological frameworks – rule-based phonology, OT, and finite-state phonology – all of which attempt to account for opacity in parallel without intermediate representations, typically referencing the input to the derivation. Our findings therefore weaken recent theoretical proposals that follow the input-based characterization of opacity rather than the traditional, serial characterization.

Of course, the theoretical comparison we have offered in this paper includes no formal proof that any successful account of the typological asymmetry will involve serialism, or that any parallel theory of opacity will fail. Indeed, Wolf (2011) – whose lead we have followed in this paper – has shown that two different serial versions of OT with Candidate Chains (McCarthy 2007a) can either generate mutual counterbleeding and mutual counterfeeding or not, depending on different auxiliary assumptions made within a serial framework. And while we currently are not aware of a fully parallel theory of opacity that can account for the asymmetry, it is conceivable that one could be discovered, in which case the conclusions to be drawn from the NO-SIMULTANEITY UNIVERSAL would be more nuanced. Our theoretical comparison also clarifies what is at stake if a convincing counterexample to NO-SIMULTANEITY is found: theories that are currently at a disadvantage because of their generation of crucially-simultaneous interactions would gain a new advantage over theories that cannot generate such interactions. Our hope is that further studies on opacity, and on process interaction in phonology more generally, could help advance our understanding of the right architecture of phonology.

References

Al-Mozainy, Hamza Qublan. 1981. Vowel alternations in a Bedouin Hijazi Arabic dialect: Abstractness and stress. Doctoral Dissertation, University of Texas at Austin.

Anderson, Stephen. 1974. *The organization of phonology*. New York: Academic Press.

Baković, Eric. 2007a. Mutually-assured destruction. Post on Phonoloblog, February 19, 2007. Available online at https://quote.ucsd.edu/phonoloblog/2007/02/19/mutually-assured-destruction/.

Baković, Eric. 2007b. A revised typology of opaque generalisations. *Phonology* 217–259.

Baković, Eric. 2011. Opacity and ordering. In *The handbook of phonological theory, second edition*, ed. John Goldsmith, Alan C. L. Yu, and Jason Riggle, 40–67. Malden: Wiley-Blackwell.

Baković, Eric, and Lev Blumenfeld. 2024. A formal typology of process interactions. *Phonological Data and Analysis* 6:1–43.

Bermúdez-Otero, Ricardo. 1999. Constraint interaction in language change: quantity in English and Germanic [opacity and globality in phonological change]. Doctoral Dissertation, University of Manchester and Universidad de Santiago de Compostela.

Bermúdez-Otero, Ricardo. 2018. Stratal phonology. In *The Routledge handbook of phonological theory*, ed. S.J. Hannahs and Anna R. K. Bosch, 100–134. Abingdon: Routledge.

Bistry, Moty, Si Berrebi, Eyal Marco, and Ezer Rasin. 2023. An overview of phonological interactions in Judeo-Baghdadi Arabic. Ms., Tel-Aviv University. Submitted.

Blumenfeld, Lev. 2003. Counterfeeding, derived environment effects, and comparative markedness. *Theoretical Linguistics* 29:89–99.

Bye, Patrik A. 2002. Virtual Phonology: Rule-sandwiching and multiple opacity in North Saami. Doctoral Dissertation, University of Tromsø.

Chandlee, Jane. 2014. Strictly local phonological processes. Doctoral Dissertation, University of Delaware.

Chandlee, Jane, and Jeffrey Heinz. 2018. Strict locality and phonological maps. *Linguistic Inquiry* 49:23–60.

Chandlee, Jane, Jeffrey Heinz, and Adam Jardine. 2018. Input strictly local opaque maps. *Phonology* 35:171–205.

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

Ettlinger, Marc. 2008. Input-driven opacity. Doctoral Dissertation, University of California, Berkeley.

Faust, Noam, and Francesc Torres-Tamarit. 2017. Stress and final /n/ deletion in Catalan: Combining Strict CV and OT. *Glossa: a journal of general linguistics* 2.

Geytenbeek, Brian, and Helen Geytenbeek. 1971. *Gidabal grammar and dictionary*, volume 43 of *Australian Institute of Aboriginal Studies*. Canberra.

Goldrick, Matthew. 2000. Turbind output representations and the unity of opacity. In *Proceedings of the North East Linguistic Society*, ed. Masako Hirotani, Andries Coetzee, Nancy Hall, and Ji-yung Kim, 231–246. Rutgers University: Graduate Linguistic Student Association.

Gouskova, Maria. 2003. Deriving economy: syncope in Optimality Theory. Doctoral Dissertation, University of Massachusetts Amherst.

Hale, Kenneth. 1973. Deep-surface canonical disparities in relation to analysis and change: An Australian example. *Current trends in linguistics* 11:401–458.

Heinz, Jeffrey. 2015. A Hilbert problem for phonology. Post in the online blog "Faculty of Language". Available online at: `http://facultyoflanguage.blogspot.com/2015/01/the-5th-hilbert-question-jeff-heinz.html`.

Hyman, Larry M, and Al Mtenje. 1999. Prosodic morphology and tone: The case of Chichewa. In *The prosody-morphology interface*, ed. René Kager, Harry van der Hulst, and Wim Zonneveld, 90–133. Cambridge: Cambridge University Press.

Jarosz, Gaja. 2014. Serial Markedness Reduction. In *Proceedings of the 2013 Annual Meeting on Phonology*, ed. John Kingston, Claire Moore-Cantwell, Joe Pater, and Robert Staubs. Washington, DC: Linguistic Society of America.

Johnson, Douglas. 1972. *Formal aspects of phonological description*. The Hague: Mouton.

Kaplan, Aaron F. 2008. Noniterativity is an emergent property of grammar. Doctoral Dissertation, University of California, Santa Cruz.

Kavitskaya, Darya, and Peter Staroverov. 2010. When an interaction is both opaque and transparent: the paradox of fed counterfeeding. *Phonology* 27:255–288.

Kenstowicz, Michael. 1994. *Phonology in Generative Grammar*. Blackwell.

Kenstowicz, Michael, and Charles Kisseberth. 1977. *Topics in phonological theory*. New York: Academic Press.

Kenstowicz, Michael, and Charles Kisseberth. 1979. *Generative phonology: Description and theory*. Academic Press.

Kidda, Mairo Elinor. 1985. *Tangale phonology: A descriptive analysis (nigeria)*. University of Illinois at Urbana-Champaign.

Kiparsky, Paul. 1968. How abstract is phonology? Indiana University Linguistics Club.

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

Kiparsky, Paul. 2000. Opacity and cyclicity. *The Linguistic Review* 17:351–367.

Kiparsky, Paul. 2015. Stratal OT: A synopsis and FAQs. In *Capturing phonological shades within and across languages*, ed. Yuchau E. Hsiao and Lian-Hee Wee, 2–44. Cambridge Scholars Publishing.

Koutsoudas, Andreas, Gerald Sanders, and Craig Noll. 1974. The application of phonological rules. *Language* 50:1–28.

Lubowicz, Anna. 2011. Chain shifts. In *The Blackwell companion to phonology*, ed. Marc van Oostendorp, Colin Ewen, Elizabeth Hume, and Keren Rice, 1717–1735. Oxford: Blackwell.

Mascaró, Joan. 1976. Catalan phonology and the phonological cycle. Doctoral Dissertation, MIT.

McCarthy, John. 2000. Harmonic serialism and parallelism. In *Proceedings of the North East Linguistic Society*, ed. Masako Hirotani, Andries Coetzee, Nancy Hall, and Ji-yung Kim, 501–524. Rutgers University: Graduate Linguistic Student Association.

McCarthy, John. 2003a. Comparative Markedness. *Theoretical Linguistics* 29:1–51.

McCarthy, John. 2003b. Sympathy, cumulativity, and the Duke-of-York gambit. In *The syllable in Optimality Theory*, ed. Caroline Fery and Ruben van de Vijver, 23–76. Cambridge: Cambridge University Press.

McCarthy, John. 2003c. What does comparative markedness explain, what should it explain, and how? *Theoretical Linguistics* 29:141–155.

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

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

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

McCollum, Adam, and Darya Kavitskaya. 2018. Non-iterative vowel harmony in Crimean Tatar. In *Proceedings of the 35th West Coast Conference on Formal Linguistics (WCCFL)*, ed. William G. Bennett, Lindsay Hracs, and Dennis Ryan Storoshenko, 259–268. Somerville, MA: Cascadilla Press.

McCollum, Adam G, and Darya Kavitskaya. 2022. On the status of non-iterativity in feature spreading. *Glossa: a journal of general linguistics* 7.

Moreton, Elliot, and Paul Smolensky. 2002. Typological consequences of local constraint conjunction. In *Proceedings of the 21st West Coast Conference on Formal Linguistics*, ed. Line Mikkelsen and Christopher Potts, 306–319. Cambridge, MA: Cascadilla Press.

Moreton, Elliott. 2003. Non-computable functions in Optimality Theory. In *Optimality Theory in phonology: A reader*, ed. John J. McCarthy, 141–163. Malden, MA, and Oxford, UK: Blackwell.

Newton, B. E. 1971. Ordering paradoxes in phonology. *Journal of Linguistics* 7:31–53.

Papillon, Maxime. 2017. Attested tone circles are not a challenge for rule-based phonology as they are not circles. Presentation at Manchester Phonology Meeting 21 (MFM), May 2017, University of Manchester.

Phelps, Elaine. 1975. Iteration and disjunctive domains in phonology. *Linguistic Analysis* 1:137–172.

Postal, Paul Martin. 1968. *Aspects of phonological theory*. Harper & Row New York.

Prince, Alan, and Paul Smolensky. 1993/2004. *Optimality Theory: Constraint interaction in generative grammar*. Wiley-Blackwell: Oxford.

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

Pullum, Geoffrey K. 1976. The Duke of York gambit. *Journal of linguistics* 12:83–102.

Purnell, Thomas. 2017. Rule-based phonology: background, principles and assumptions. In *The Routledge handbook of phonological theory*, ed. S.J. Hannahs and Anna R.K. Bosch, 135–166. Routledge.

Rajpopat, Rishi. 2023. The evolution of conflict-resolution tools in the early Pāṇinian tradition. *Bhasha. Journal of South Asian Linguistics, Philology and Grammatical Traditions* 2:31–58.

Rasin, Ezer. 2022. Shifting interactions and countershifting opacity: A note on opacity in Harmonic Serialism. *Linguistic Inquiry* 53:836–851.

Trommer, Jochen, and Eva Zimmermann. 2014. Generalised mora affixation and quantity-manipulating

morphology. *Phonology* 31:463–510.
Wolf, Matthew. 2011. Limits on global rules in Optimality Theory with Candidate Chains. *Phonology* 28:87–128.
Yadav, Ramawatar. 1996. *A reference grammar of Maithili*, volume 11. Walter de Gruyter.