Consequences of contextual faithfulness constraints in Harmonic Serialism

Ivy Hauser

July 2016

1 Introduction

Faithfulness constraints which demand faithfulness in a specific context have been explored in the phonological literature and can be used to capture various asymmetries relating to specific positions. Positional faithfulness constraints (Beckman, 1997; Lombardi, 1999) provide a specific context for faithfulness corresponding to positions of prosodic prominence in the output. This paper develops a new kind of contextual faithfulness constraint, one which does not necessarily correspond to a position of prosodic or psycholinguistic prominence and has a input defined context. The main finding is that when these types of constraints are used in Harmonic Serialism, they can produce rule ordering and local directional effects. I give analyses of three opaque phenomena: counterbleeding, double counterbleeding, and self-destructive feeding to illustrate how contextual faithfulness constraints deal with rule ordering effects. Local directionality effects can also be controlled because these constraints can specify an input context to the left or right of the targeted position. To illustrate this, I provide analyses of directional assimilation and cluster simplification.

Section 2 of this paper reviews previous work relating to contextual faithfulness constraints. Similar constraints have been used in the literature before, but with some key differences from the constraints I examine here. In Section 3, I review the formalization of the constraints and the restrictions on the system. In Section 4, I explain the differences which arise when contextual faithfulness constraints are used in Harmonic Serialism vs. parallel Optimality Theory. Section 5 contains the opaque analyses, and Section 6 contains the directional analyses. Sections 7-8 conclude and discuss directions for further work.

2 Previous work

Adding context to faithfulness constraints has been explored in detail by Beckman (1997) and Lombardi (1999) with positional faithfulness constraints. These constraints provide faithfulness for prosodically prominent positions in the output and are intended to capture various phonological asymmetries observed in these positions. For example, it is often the case that phonological contrasts will be maintained in these positions while neutralized in others. Segments in prominent positions also often trigger phonological processes. Beckman (1997) provides a list of prominent positions which have some perceptual or processing advantage requiring special faithfulness:

root-initial syllables, stressed syllables, syllable onsets, roots, and long vowels. She also provides phonological diagnostics for positional privilege: contrast maintenance with neutralization elsewhere, triggering of phonological processes, and positional resistance to processes which apply elsewhere.

These phonological asymmetries are captured with positional faithfulness constraints in a grammar where positional faithfulness dominates markedness and general faithfulness. The constraints refer to positions defined in the output and assign violations when a segment in a certain output position is unfaithful to its input correspondent. If the segment in question is not in the privileged output position, the positional faithfulness constraint is vacuously satisfied. Beckman's general form for positional faithfulness constraints is given in (1).

(1) General positional faithfulness (Beckman, 1997)
IDENT-*Position*(F): Let β be an output segment in a privileged position P and α the input correspondent of β. If β is [γF], then α must be [γF].
"Correspondent segments in a privileged position must have identical specifications for [F]."

The position can be any of the previously specified prosodically prominent positions. Ranking a positional faithfulness constraint over markedness, and markedness over general faithfulness will produce positional asymmetries. This general pattern provides a unified explanation for a variety of different phenomena, which can all be analyzed as having specific faithfulness to prominent positions.

The contextual faithfulness constraints that I examine here are similar to positional faithfulness constraints but have a few crucial differences. These constraints specify an input context instead of an output context. Previous approaches to positional faithfulness have specified output contexts in order to refer to prosodic positions, such as onset position, which would not be specified in the input. If syllabification happens during the phonological derivation, onsets can only be referred to in the output. The contextual faithfulness framework proposed here allows faithfulness constraints to refer to contexts in the input, which can provide crucial distinctions not available with output contexts.

For example, Wilson (2001) demonstrates a problem associated with contextual faithfulness constraints that define an output context. There is a cross-linguistic generalization that VCCV clusters are often simplified by deleting the second consonant, not the first. An appropriate positional faithfulness approach would be to have a constraint IDENT-*Onset*. However, deleting either of the consonants would satisfy that constraint because the remaining consonant will be syllabified as an onset regardless of which consonant is deleted. Having an input defined contextual faithfulness constraint can distinguish between deletion of the first consonant and the second. I will return to this analysis in Section 6.2.

Positional faithfulness constraints have previously only been used to demand faithfulness in positions of prosodic prominence. This is meant to reflect the fact that these positions are more prominent for perceptual or other processing reasons. In the contextual faithfulness system, any segmental context can be specified but it must be local to the target segment. Positional faithfulness constraints can only analyze phenomena related to such prosodically prominent positions and the contextual faithfulness system provides a way of analyzing phonological asymmetries which do not have an explanation from prosodic or psycholinguistic prominence. These contextual faithfulness constraints (demanding faithfulness to a context in the input) can be used in a system alongside positional faithfulness constraints (demanding faithfulness to a certain position in the output). The addition of contextual faithfulness expands the possible contexts which can require faithfulness.

3 Constraint formalization

Contextual faithfulness constraints are constructed with input-output correspondence and a specified context in the input. The general form is given below with MAX and IDENT constraints as examples.

(2) IDENT(F)/*Context*: Let β be an output segment in some context C and α the input correspondent of β. If α is [γF] and in context C, then β must be [γF].
"Correspondent segments in the specified input context must have identical specifications for [F]."

MAX(α)/*Context*: Let α be an input segment in the context C. α must have a corresponding segment in the output.
"Segments in the specified context C must have output correspondents."

Any faithfulness constraint can be used with the addition of a specified input context. Some examples are given in (3).

(3) Examples of contextual faithfulness
MAX(C)/_V: Assign one * if a consonant is deleted which precedes a vowel in the input.
IDENT(high)/_#: Assign one * if the value for [high] is changed on a segment which is word-final.
IDENT/V_: Assign one * if the values for any features are changed on a segment which is post-vocalic.

These constraints are always satisfied if the specified context does not exist in the input. Possible contexts that can be specified are restricted by locality. The context in a contextual faithfulness constraint must be local to the targeted segment.[1] The relevant context can either directly precede or follow the segment but cannot be a distant context or some other property of the input. This has an implication for the types of phenomena which can be analyzed with this constraints in terms of directionality: only locally directional processes can be analyzed with contextual faithfulness.

4 Contextual faithfulness in a serial framework

4.1 Harmonic Serialism

Harmonic Serialism is a serial version of Optimality Theory in which GEN is limited to making one change at a time (McCarthy, 2010). Candidates are derived and evaluated in a series of multiple steps. Each candidate must either be identical to the input or differ from the input by only a single operation (gradualness) and the output of each step must be better-formed according to the language hierarchy than the output of the previous step (harmonic improvement). Each output of

[1]Constraints can be defined to be local in terms of the segmental string or on a relevant segmental tier.

GEN and EVAL is submitted as the input for a following pass until no further changes can be made and the derivation converges. Convergence occurs when the output at a step is identical to the input at that step and no further changes can be made.

4.2 Rule ordering effects

Because Harmonic Serialism has multiple passes through GEN and EVAL, it is possible to specify a context in a contextual faithfulness constraint which does not exist until or ceases to exist at some intermediate step. This type of situation allows for the analysis of different types of rule ordering effects. In cases of feeding, bleeding, and phonological opacity, rules are crucially and easily ordered in a rule based framework. Such rule ordering effects have generally posed problems for constraint based grammars in the literature because there is no way to specify in what order different processes apply.

For some derivations, Harmonic Serialism provides a way of ordering processes. Because there are multiple passes through GEN and EVAL, processes can be ordered through constraint ranking. Higher ranked markedness constraints will be satisfied first because satisfaction of a higher ranked constraint produces a more optimal candidate and only one change is allowed at a time. Simply ranking the markedness constraint which is satisfied first over the one that is satisfied second does not suffice to analyze the opaque cases that I present below. In these cases, there is one change which will satisfy both markedness constraints simultaneously. Because of this, opaque processes cannot be analyzed in standard HS or parallel OT. I provide a common example from Bedouin Arabic below.

(4) Bedouin Arabic counterbleeding (Al-Mozainy, 2007)

	/i/ deletion	palatalization	counterbleeding
UR	/ʃaribat/	/ħaːkim/	/ħaːkim-in/
palatalization	-	ħaːkʲim	ħaːkʲimin
deletion	ʃarbat	-	ħaːkʲmin
SR	[ʃaribat]	[ħaːkʲim]	[ħaːkʲmin]

There is a counterbleeding interaction where palatalization precedes syncope. /i/ deletes in open syllables and consonants palatalize preceding /i/. These individual processes are shown in the two transparent cases. In the counterbleeding case, the palatalization of /k/ happens before /i/ deletes. Once /i/ deletes, on the surface there is no evidence for the trigger of palatalization, making the interaction opaque. While the rule based analysis can easily capture the interaction by simply ordering palatalization to apply before syncope, neither a standard OT nor HS analysis can capture this.

(5) Bedouin Arabic counterbleeding in standard HS

Relevant markedness constraints

- *[ki]: Assign one * for each sequence of an unpalatalized voiceless consonant before [i].
- *iCV: Assign one * for [i] in an open syllable.

Step 1: /ħaːkim-in/→ [ħaːkʲmin]

	/ħaːkim-in/	*[ki]	IDENT[bk]	*iCV	MAX
→ 1.	ħaːkʲimin		*	*	
💣 2.	ħaːkmin				*
3.	ħaːkim-in	*		*	

In the tableau in (5), the desired winner is candidate 1, the one which palatalizes the [k] preceding [i]. Then at Step 2, syncope should apply. Candidate 2 instead deletes [i] immediately which simultaneously satisfies both markedness constraints. In order to rule out the candidate and demand palatalization happen first, MAX must outrank *iCV . This would result in a ranking paradox because *iCV must outrank MAX in order for deletion to occur later in the derivation, and in the transparent case, shown in (6). The same problem occurs in parallel OT.

(6) Transparent deletion: /ʃaribat/ → [ʃarbat]

	/ʃaribat/	*ki	ID(bk)	*iCV	MAX
→ 1.	ʃarbat				*
2.	ʃaribat			*W	L

Rule ordering effects like counterbleeding opacity present a problem for standard HS and parallel OT because there is no way to order satisfaction of markedness constraints when simultaneous satisfaction of both markedness constraints is an option. Contextual faithfulness constraints provide a way to derive rule ordering effects in an HS system because they can specify a context which only becomes available at an intermediate step.

The generic problem with counterbleeding in HS is that there are two markedness constraints that should both be satisfied on the surface but they cannot be simultaneously satisfied in the same step of the derivation. One must be satisfied before the other, as in rule ordering. In the case of Bedouin Arabic, the /i/ should delete, but only after palatalization has applied. Candidate 2 in tableau (5) is the incorrectly optimal candidate where /i/ deletion applies too soon.

This can be ruled out with a contextual faithfulness constraint which demands faithfulness for /i/ but only until palatalization has applied, which derives the rule ordering effect by preventing deletion from applying first. This is done by specifying a context for the faithfulness to /i/, which ceases to exist after palatalization has applied. This becomes possible by combining contextual faithfulness constraints with the Harmonic Serialism framework. In parallel OT, it is not possible to refer to an intermediate context or state of the derivation because all changes are made at once. Because of the existence of multiple steps in HS, it is possible to indirectly refer to an intermediate step by defining an appropriate context. Specifying a context which no longer exists after a rule has applied is a way of essentially rendering a constraint inactive after a certain step of the derivation.

Counterbleeding is one of a few different types of rule ordering effects which can be analyzed in the same way with contextual faithfulness: counterbleeding, double counterbleeding, and self-destructive feeding. These are all types of overapplication opacity and they are individually examined in the following section. Contextual faithfulness constraints can be used to analyze overapplication opacity but not underapplication opacity. I return to the difference between overapplication and underapplication opacity in section 5.4.

5 Opaque analyses

5.1 Counterbleeding

Bedouin Arabic counterbleeding can be analyzed by adding a contextual faithfulness constraint to the standard HS constraint set.

(7) MAX(i)/k_: Assign one * if [i] is deleted when preceded by a non-palatalized voiceless consonant in the input.

This constraint derives the rule ordering effect of palatalization preceding deletion by assigning a violation when deletion occurs and palatalization has not yet occurred. Providing a context for the faithfulness demanded by MAX effectively allows reference to this intermediate stage of the derivation. The context will only exist before palatalization has applied, after which the constraint will not be violated because the specified context does not exist. The addition of this constraint can solve the ranking paradox presented in the standard analysis. The contextual faithfulness constraint outranks the relevant markedness constraint, which outranks the general faithfulness constraint.

(8) Derivation path
/ħaːkim-in/ → ħaːkʲimin → ħaːkʲmin → [ħaːkʲmin]

Rankings

- MAX(i)/k_ ≫ *iCV, IDENT[bk]
- *ki ≫ IDENT[bk]
- *iCV≫ MAX

Step 1: palatalization occurs

	/ħaːkim-in/	MAX(i)/k_	*iCV	*ki	IDENT[bk]	MAX
→ 1.	ħaːkʲimin		*		*	
2.	ħaːkmin	*W	L		L	*W
3.	ħaːkimin		*	*W	L	

Step 2: deletion occurs

	ħaːkʲimin	MAX(i)/k_	*iCV	*ki	IDENT[bk]	MAX
→ 1.	ħaːkʲmin					*
2.	ħaːkʲimin		*W			L

Step 3: convergence

	ħaːkʲmin	MAX(i)/k_	*iCV	*ki	IDENT[bk]	MAX
→ 1.	ħaːkʲmin					

The derivation in (8) shows how the contextual faithfulness constraint rules out the candidate which was previously problematic in the standard HS analysis. At Step 1, candidate 2, which deletes the [i], satisfies both relevant markedness constraints but violates the contextual faithfulness constraint demanding faithfulness when /i/ is preceded by an unpalatalized consonant in the input. Ranking MAX(i)/k_ over the relevant markedness constraint, *iCV, prevents deletion from occurring before palatalization has occurred. At Step 2, the context specified in MAX(i)/k_ no longer exists so it is not violated by the deletion of [i]. Instead, general case MAX is violated which is ranked below the markedness constraint, *iCV.

The ranking of CONTEXTUALFAITH ≫ MARKEDNESS ≫ GENERALFAITH has been used in other positional faithfulness analyses and allows for the analysis of various opaque processes without causing problems for the transparent patterns in the same system. The transparent processes in Bedouin Arabic are given below. Because the relevant context is non-existent in the transparent deletion case, the contextual faithfulness constraint does not impact the deletion demanded by *iCV. In the transparent palatalization case, the context is present at Step 1, but adding the contextual faithfulness constraint does not change the system because the winner does not violate it. Either MAX(i)/k_ or MAX outranks IDENT[bk], which rules out the candidate which deletes.

(9) Transparent deletion
Step 1:

/ʃaribat/	MAX(i)/k_	*iCV	*ki	IDENT[bk]	MAX
→ 1. ʃarbat					*
2. ʃaribat				*W	L

(10) Transparent palatalization
Step 1:

/ ħaːkim/	MAX(i)/k_	*iCV	*ki	IDENT[bk]	MAX
→ 1. ħaːkʲim				*	
2. ħaːkim			*W	L	
3. ħaːkm	*W			L	*W

5.1.1 Stepwise deletion

It is sometimes claimed in the HS literature that deleting an entire segment cannot be considered "one change." In this case, all deletion occurs in two steps: reduction/removal of place, then deletion (McCarthy, 2010). If we are working with a model of GEN that cannot delete entire segments at once, these types of patterns can still be analyzed with slight changes to the constraint set. Instead of a constraint MAX(i)/k_, a constraint which punishes removal of place, MAX-PL(i)/k_ , is required. The question of whether deletion is one change or two does not impact the analysis with contextual faithfulness. Simply reformulating the constraint to penalize the removal of place instead of total deletion allows for an analysis with the stepwise deletion model of GEN.

(11) Bedouin Arabic counterbleeding with stepwise deletion

Derivation path
/ħaːkim-in/ → ħaːkʲimin → ħaːkʲəmin → [ħaːkʲmin]

New/revised constraints

- MAX-PL(i)/k_: Assign one * if the place is removed from an /i/ preceded by an unpalatalized voiceless consonant in the input.
- MAX: Assign one * if a segment skeleton is deleted.
- MAX-PL: Assign one * if vowel place is deleted.

- *əCV: Assign one * if a vowel skeleton precedes an open syllable.

An additional markedness constraint, *əCV is needed to compel deletion of the vowel skeleton after the place features have been removed. *iCV only compels removal of place features. The crucial interaction involving this constraint is at Step 3.

Rankings

- Max-Pl(i)/k_ ≫ *iCV, Ident[bk]
- *ki ≫ Ident[bk]
- *iCV≫ *əCV≫ Max-Pl

Step 1: palatalization

/ħaːkim-in/	Max-Pl(i)/k_	*iCV	*əCV	*ki	Ident[bk]	Max-Pl
→ 1. ħaːk^{j}imin		*			*	
2. ħaːkəmin	*W	L	*		L	*W
3. ħaːkimin		*		*W	L	

Step 2: deletion of place

ħaːk^{j}imin	Max-Pl(i)/k_	*iCV	*əCV	*ki	Ident[bk]	Max-Pl
→ 1. ħaːk^{j}əmin			*			*
2. ħaːk^{j}imin		*W	L			L

Step 3: deletion of segmental skeleton

ħaːk^{j}əmin	Max-Pl(i)/k_	*iCV	*əCV	*ki	Ident[bk]	Max
→ 1. ħaːk^{j}min						*
2. ħaːk^{j}əmin			*W			L

Step 4: convergence

ħaːk^{j}min	Max-Pl(i)/k_	*iCV	*əCV	*ki	Ident[bk]	Max-Pl	Max
→ 1. ħaːk^{j}min							

In a system with stepwise deletion, the contextual faithfulness constraint operates in the same way. The same problematic candidate (candidate 2) is ruled out at Step 1 by Max-Pl(i)/k_, ensuring deletion does not happen first. In this and following analyses I provide, the stepwise and full deletion models work in the same way, as in the Bedouin Arabic case. Neither stepwise deletion nor full deletion is crucial for any of the phenomena I present in this paper. All analyses can be done with either model of Gen and corresponding contextual faithfulness constraints. For simplicity of viewing how the contextual faithfulness constraints derive the rule ordering effects, I will use a full deletion model of Gen from this point forwards.

5.1.2 Interim discussion: counterbleeding

Using a faithfulness constraint that specifies a context only available before a certain process has applied can derive rule ordering effects in an HS system. I have shown this with a case of counterbleeding in Bedouin Arabic. With a standard HS model, there is no way to demand that palatalization happen before deletion. The problematic candidate at Step 1 simultaneously satisfies both markedness constraints by deleting /i/. Demanding specific faithfulness for /i/ when following a non-palatalized consonant prevents deletion until after the palatalization has applied, providing an effect akin to rule ordering. This approach captures rule ordering effects with simple faithfulness constraints and does not add additional technology to the system. The only addition is a specified context for faithfulness.

5.2 Double counterbleeding

The same principles in the Bedouin Arabic counterbleeding case can also capture cases of double counterbleeding. This is shown with data from Yawelmani, where there are two individual counterbleeding interactions which combine to create double counterbleeding with an eligible input. Two contextual faithfulness constraints are required, one for each counterbleeding process. There are three relevant processes: rounding harmony, vowel lowering, and closed syllable shortening.

(12) Yawelmani ordered processes (Newman, 1944; McCarthy, 2007)
1. Rounding harmony (RH): suffix vowel matches rounding to the preceding vowel if they match in height
2. Lowering (L): high long vowels become non-high
3. Closed syllable shortening (CSS): long vowels shorten in closed syllables

These three processes combine in two separate counterbleeding interactions and one double counterbleeding interaction (given below). Rounding harmony crucially precedes lowering in /cu:m-it/ → [co:mut]. Because there are no superheavy syllables in the word, closed syllable shortening does not apply. Lowering crucially precedes closed syllable shortening in /mi:k-hin/ → [mekhin]. The stem vowel and suffix vowel already match in lack of rounding so harmony is not applied. In the correct order, all three processes can apply to an input like /cu:m-hin/, creating a double counterbleeding interaction.

(13) Counterbleeding interactions

	/mi:k-hin/	/cu:m-it/	/cu:m-hin/
RH	-	cu:mut	cu:mhun
L	me:khin	co:mut	co:mhun
CSS	mekhin	-	comhun
	[mekhin]	[co:mut]	[comhun]

In this case, two separate contextual faithfulness constraints are needed, one for each individual counterbleeding interaction. When lowering counterbleeds rounding harmony, the appropriate contextual faithfulness constraint is ID(hi)/[αrd][βrd], which demands faithfulness in height when the following segment on the vowel tier has a non-identical specification for [round]. This prevents lowering from happening until after rounding harmony has applied. When closed syllable shortening counterbleeds lowering, the appropriate contextual faithfulness constraint is ID(long)/[hi], which demands faithfulness in the feature [long] to segments which are [high] in the input, preventing vowel shortening from applying until the vowel is no longer [high] and lowering has applied.

(14) Constraints

- ID(long)/[hi]: Assign one * if the value of [long] is changed on a segment which is [high] in the input.

- ID(hi)/[αrd][βrd]: Assign one * if the value of [high] is changed on a segment which precedes a segment with a non-identical specification for [round] on the vowel tier in the input.
- ROUND: Assign one * if the suffix vowel and the preceding root vowel are high and do not match in rounding.[2]
- *[long][hi]: Assign one * for every segment which is both [long] and [hi].
- *[μμμ]$_\sigma$: Assign one * for every syllable containing three moras.
- ID(long): Assign one * if [long] is added or removed from a segment.
- ID(hi): Assign one * if [hi] is added or removed from a segment.
- ID(rd): Assign one * if [round] is added or removed from a segment.

It should be noted that there is no underlying contrast between /i/ and /u/ in Yawelmani suffixes (Kenstowicz, 1994). For simplicity of this example, I am using underlying /i/ in the suffixes (taken to be the "basic" form of the suffix in Kenstowicz (1994)). The way I have formulated the constraint ID(hi)/[αrd][βrd], the analysis can account for the data regardless of whether the suffix is underlyingly round, non-round, or underspecified. The contextual faithfulness constraint prevents change in [high] in a disharmonic rounding context. A full analysis with alternative underlying representations is given in the appendix.

(15) Closed syllable shortening and lowering counterbleeding

Rankings

- ID(long)/[hi] ≫ *[μμμ]$_\sigma$ ≫ ID(long)
- *[long][hi] ≫ ID(hi)

Step 1: lowering

	/mi:k-hin/	ID(long)/[hi]	*[μμμ]$_\sigma$	ID(long)	*[long][hi]	ID(hi)
→ 1.	me:khin		*			*
2.	mikhin	*W	L	*W		L
3.	mi:khin		*		*W	L

Step 2: closed syllable shortening

	me:khin	ID(long)/[hi]	*[μμμ]$_\sigma$	ID(long)	*[long][hi]	ID(hi)
→ 1.	mekhin			*		
2.	me:khin		*W	L		

At Step 1 in (15), vowel lowering needs to apply before closed syllable shortening. Candidate 2 simultaneously satisfies both relevant markedness constraints, *[μμμ]$_\sigma$ and *[long][hi], by shortening at Step 1. In a system without the contextual faithfulness constraint, this candidate would be the winner. The faithfulness constraint, ID(long)/[hi] prevents this candidate from winning by

[2]This is a simplified version of several constraints that capture the rounding harmony. I adopt this here for explanatory simplicity to focus on the contextual faithfulness constraints since I am only considering the counterbleeding cases and not other processes in Yawelmani.

demanding faithfulness in [long] for vowels which are [hi] in the input. Once the lowering applies at Step 1, the context in this faithfulness constraint no longer exists because the vowel in question is no longer high. At Step 2, ID(long)/[hi] is not violated by the winning candidate which shortens.

(16) Lowering and rounding harmony counterbleeding

Rankings

- ID(hi)/[αrd][βrd] ≫ *[long][hi] ≫ ID(hi)
- ROUND ≫ ID(rd)

Step 1: rounding harmony

/cu:m-it/	ID(hi)/[αrd][βrd]	*[long][hi]	ID(hi)	ROUND	ID(rd)
→ 1. cu:mut		*			*
2. co:mit	*W	L	*W		L
3. cu:mit		*		*W	L

Step 2: lowering

cu:mut	ID(hi)/[αrd][βrd]	*[long][hi]	ID(hi)	ROUND	ID(rd)
→ 1. co:mut			*		
2. cu:mut		*W	L		

The same approach is used in the counterbleeding relationship between lowering and rounding harmony. Rounding harmony must apply first but there is a candidate at Step 1 which simultaneously satisfies both relevant markedness constraints, candidate 2, by lowering at Step 1. This candidate is ruled out by the contextual faithfulness constraint preventing lowering in the context where rounding would apply. After rounding has applied, the context no longer exists and the lowering can take place at Step 2.

With these rankings in place, double counterbleeding occurs with an input like /cu:m-hin/. The input will first undergo rounding harmony at Step 1, lowering at Step 2, and closed syllable shortening at Step 3.

(17) Double counterbleeding in Yawelmani

Step 1: rounding harmony

/cu:m-hin/	ID(long)/[hi]	*[μμμ]$_\sigma$	ID(long)	ID(hi)/[αrd][βrd]	*[long][hi]	ID(hi)	ROUND	ID(rd)
→1. cu:mhun		*			*			*
2. cu:mhin		*			*		*W	L
3. co:mhin		*		*W	L	*W		L
4. cumhin	*W	L	*W				*W	L

Step 2: lowering

cu:mhun	ID(long)/[hi]	*[μμμ]$_\sigma$	ID(long)	*[long][hi]	ID(hi)
→ 1. co:mhun		*			*
2. cu:mhun		*		*W	L
3. cumhun	*W	L	*W		

Step 3: closed syllable shortening

	co:mhun	*[μμμ]$_\sigma$	ID(long)
→ 1.	comhun		*
2.	co:mhun	*W	L

The use of contextual faithfulness constraints can be generalized for multiple cases of counterbleeding. When it is possible that two counterbleeding interactions apply in succession with the same input, the contextual faithfulness constraints used in each individual counterbleeding interaction will produce the double counterbleeding interaction. The constraints do not pose problems for cases of layered overapplication opacity.

5.3 Self-destructive feeding

The same principles of contextual faithfulness can be used to analyze another type of opacity, self-destructive feeding (Baković, 2011). Self-destructive feeding is an interaction where one rule feeds another which crucially changes the form such that the application of the first rule is now opaque. Like counterbleeding, this is a form of overapplication opacity. The interaction of elision and deletion in Turkish is an example of such an interaction.

(18) Processes (Kenstowicz, 1994)
1. Elision: [s] and [j] delete after consonants.
2. Deletion: [g] deletes intervocalically

(19) Interactions

	/bebeg + i/	/tʃan + sɯ/	/ajag + sɯ/
elision	-	tʃan ɯ	ajag ɯ
deletion	bebe i	-	aja ɯ
	[bebei]	[tʃansɯ]	[ajaɯ]

With an input like /ajag + sɯ/, elision feeds deletion by providing the intervocalic context for the /g/ to delete. Once the /g/ deletes, elision becomes opaque. On the surface, there is no remnant of the environment in which elision occurred. As in the counterbleeding cases, a contextual faithfulness constraint is required which prevents the application of deletion until after elision has applied. In this case, that constraint is MAX(g)/_C, which demands faithfulness for /g/ when it precedes a consonant. Once elision has applied and that consonant is removed, the specified context no longer exists and the [g] is free to delete in the intervocalic context which remains.

(20) Constraints

- MAX(g)/_C: Assign one * if [g] is deleted when it precedes a consonant in the input.
- *Cs: Assign one * for every consonant sequence of C followed by [s] or [j].
- *VgV: Assign one * for every intervocalic [g].

(21) /ajagsɯ/ → [ajaɯ]

Step 1: elision

/ajagsɯ/	MAX(g)/_C	*Cs	*VgV	MAX
→ 1. ajagɯ			*	*
2. ajagsɯ		*W	L	L
3. ajasɯ	*W	L	L	*W

Step 2: deletion

ajagɯ	MAX(g)/_C	*Cs	*VgV	MAX
→ 1. ajaɯ				*
2. ajagɯ			*W	L

Self-destructive feeding provides another example of overapplication opacity, a rule ordering effect which can be captured with the use of contextual faithfulness constraints. The interaction is similar to counterbleeding except that the earlier rule feeds the later rule. This does not change the analysis with contextual faithfulness; the same principles can be used. The contextual faithfulness constraint provides faithfulness in a certain context which no longer exists after the first rule has applied.

5.4 Discussion: opacity and contextual faithfulness

In this section, I have demonstrated various analyses of opaque processes using contextual faithfulness constraints. The addition of these constraints to an HS system allows us to capture certain rule-ordering effects which cannot be otherwise captured in a standard HS or parallel OT system. While contextual faithfulness constraints can derive certain rule ordering effects, it is not the case that all rule ordering effects can be captured with contexts defined in faithfulness constraints in HS.

All of the opaque cases examined here are types of overapplication opacity. These are cases where it is evident that a process has applied from the output but it is not evident how or why that process applied. In the Bedouin Arabic case, the palatalized [k^j] is left in the output, but the trigger for palatalization, [i], is no longer present. In these overapplication cases, the crucial candidate that needs to be prevented from winning is the one which satisfies both markedness constraints at Step 1. This differs from cases of underapplication opacity (counterfeeding, chain shifts, etc.). In these cases, the crucial candidate which needs to be prevented from winning is the candidate which satisfies both relevant markedness constraints at Step 2. In the counterbleeding case, both relevant markedness constraints are satisfied on the surface. In the counterfeeding case, one of the relevant markedness constraints is crucially violated on the surface for some of the inputs. Because of the nature of the contextual faithfulness analysis, it cannot provide a similar solution to underapplication opacity. To illustrate this difference, I provide an example of counterfeeding opacity below.

(22) Chain shift on vowel height in Basque (Hualde and de Urbina, 2003)
/seme-e/ → [semi-e]
/alaba-a/ → [alabe-a] ↛ *[alabi-a]

(23) Rule derivation - Basque

	/alaba-a/ ‘daughter’	/seme-e/ ‘son’
raising [-low] → [+high]	-	semi-e
raising [+low] → [-low]	alabe-a	-
	[alabe-a]	[semi-e]

In Basque, underlying low vowels raise to mid and underlying mid vowels raise to high. On the surface, [alebe-e] seems to have underapplied the rule for raising to high because the environment for the rule is present but the raising has not occurred. Unlike the counterbleeding cases, the problematic interaction happens at Step 2 in an HS system. The constraints must be ranked such that mid vowels will raise to high in the case of the underlying mid vowel as in (24). At Step 2 of the derivation with an underlying low vowel as in (25), it will be impossible to not also raise further to high.

(24) /seme-e/ → [semi-e]

Step 1: raising to high

/seme-e/	*[+low]/_V	*[-low, -hi]/_V	IDENT(hi)	IDENT(low)
→1. semi-e			*	
2. seme-e		*W	L	

*[-low, -hi]/_V ≫ IDENT(hi)

Step 2: convergence

/semi-e/	*[+low]/_V	*[-low, -hi]/_V	IDENT(hi)	IDENT(low)
→ 1. semi-e				

(25) /alaba-a/ → [alabe-a]

Step 1: raising to mid

/alaba-a/	*[+low]/_V	*[-low, -hi]/_V	IDENT(hi)	IDENT(low)
→ 1. alabe-a		*		*
2. alaba-a	*W	L		L

*[+low]/_V ≫ *[-low, -hi]/_V, IDENT(low)

Step 2: ranking problems

alabe-a	*[+low]/_V	*[-low, -hi]/_V	IDENT(hi)	IDENT(low)
→ 1. alabe-a		*		
💣 2. alabi-a		W	*L	

IDENT(hi) ≫ *[-low, -hi]/_V

Here, the ranking paradox arises at Step 2 of the derivation in (25). To prevent raising to high, IDENT(hi) must outrank *[-low, -hi]/_V. But in the derivation in (24) the opposite ranking is required to cause raising to happen. Unlike the counterbleeding case, where the two relevant markedness constraints should be satisfied on the surface, in the counterfeeding case, the constraint demanding raising to [high] is only satisfied if the input is a mid vowel.

Cases like Basque and other types of underapplication opacity cannot be analyzed using contextual faithfulness constraints. Contextual faithfulness constraints allow definition of a specified context which ceases to exist after some rule applies.[3] In the case of counterbleeding, the context is the same for the application of both rules so we cannot formulate a contextual faithfulness constraint which will only apply where it is needed, at Step 2 and not Step 1. An alternative faithfulness based solution for underapplication opacity is explored in Hauser et al. (2015) and will not be examined in this paper.

Contextual faithfulness constraints allow for analysis of overapplication opacity but cannot analyze all observed rule ordering effects. The crucial difference is that in cases of overapplication opacity there is a context associated with the application of some process and both markedness constraints are obeyed on the surface. In underapplication opacity, the context associated with the application of an ordered rule will also be relevant to the transparent case and one of the markedness constraints is crucially disobeyed on the surface for some inputs.

6 Directionality analyses

Because contextual faithfulness constraints can specify a locally adjacent context, they can also be used to control directionality of local processes. As with positional faithfulness, this is done by demanding faithfulness to a certain position. This system departs from positional faithfulness in that it is not controlled by prosodic prominence and the contexts defined are in the input, not the output. Any position can be specified with a local context, including those of prosodic prominence. Contextual faithfulness constraints are needed when directionality cannot be attributed to other such mechanisms. The addition of context to faithfulness constraints provides a framework for stipulating directionality into a system.

Because of the locality restriction, contextual faithfulness constraints can only be used to analyze locally directional processes. Long-distance directionality cannot be controlled by a constraint which only demands more stringent faithfulness in one local context. In this section, I provide an example of local directionality with direction of assimilation and cluster simplification. I also provide a case of long-distance directionality of harmony and discuss why contextual faithfulness cannot be used for such cases.

6.1 Directional assimilation

An example of local directionality comes from local place assimilation, a common phonological process which occurs in both rightward (progressive) and leftward (regressive) directions. The contextual faithfulness system allows for analysis of these effects by demanding faithfulness to a specific member of the cluster. When this specific faithfulness outranks a markedness constraint demanding assimilation, the other segment will assimilate. The ability to specify such a context provides a framework to control directionality of assimilation with constraint ranking. I provide an example of local place assimilation from Sanskrit (Allen, 1962; Steriade, 2001) which is analyzed with contextual faithfulness.

[3] In principle, these constraints could also be used for cases where a context *begins* to exists only after some rule applies. This would create a situation where one process would create some context then the contextual faithfulness would prevent a second process from applying in the context created by the first. This would be able to account for a situation in which one process feeds another but the second process can't apply until after some other process has applied to change the context created by the application of the first. This type of interaction would be a type of feeding.

(26) Progressive apical assimilation in Sanskrit (Allen, 1962)
/jyoti**ʂ**-**s**u/ → [jyoti**ʂʂ**u]
/βa**ɳ**-**n**aːm/ → [βa**ɳɳ**aːm]
/piɳ**ɖ**-**d**ʰi/ → [piɳ**ɖ**-**ɖ**ʰi]

Steriade (2001) points out that the perceptual cues for retroflexion are easier to perceive postvocalically. Because nothing is deleted, output based contextual faithfulness can be used to analyze directional assimilation. Steriade (2001) provides an example of such an analysis using the P-map framework which provides faithfulness relevant to specific positions where perceptual cues are most salient. This type of assimilation can also be analyzed with a contextual faithfulness constraint which demands faithfulness to the postvocalic consonant in the cluster. With this constraint outranking markedness, the second consonant in the cluster will be forced to assimilate to satisfy the markedness constraint.

(27) Sanskrit assimilation analysis

Constraints

- IDENT/V_: Assign one * for any feature changes of a segment which follows a vowel.
- CLUSTERCOND: Markedness constraint punishing consonant clusters which have different values for [apical].
- IDENT: General case IDENT.

Step 1: assimilation

/βa**ɳ**-**n**aːm/	IDENT/V_	CLUSTERCOND	IDENT
→ 1. βaɳɳaːm			*
2. βaɳnaːm		*W	L
3. βannaːm	*W		*

In this example, the candidate which shows regressive assimilation, candidate 3, is ruled out by the contextual faithfulness constraint. The constraint demands progressive assimilation by providing specific faithfulness for the first candidate in the cluster.

Sanskrit also has processes of regressive assimilation with different features and segments. To analyze these in the same system, the contextual faithfulness constraints must relativized to only target certain features. IDENT/V_, which caused progressive assimilation in the example above, will need to be made more specific so it does not also prevent regressive assimilation of other features. IDENT(apical)/V_ will capture this alongside a similar constraint for the regressive assimilation IDENT(nasal)/_V.

6.2 Cluster simplification

A crucial difference of the contextual faithfulness system proposed here from other positional faithfulness analyses is that the context specified is an input context. Wilson (2001) provides an example of a contextual faithfulness analysis of cluster simplification in Diola. With the type of contextual faithfulness he uses, cluster simplification cannot be analyzed because the constraints refer to the output context. Simply changing the contextual faithfulness constraints to refer to the input context allows for a simple analysis of the same data.

The Diola data reflects a larger trend in cluster simplification; deletion which applies to intervocalic consonant clusters tends to apply to the first consonant in the cluster. This can be analyzed with specific faithfulness to the prevocalic consonant in the cluster.

(28) Diola cluster simplification (Sapir, 1965)
/letkujaw/ → [lekujaw] 'they won't go'
/kutɛbsinaŋas/ → [kutɛsinaŋas] 'they carried the food'
/ɛkɛtbo/ → [ɛkɛbo] 'death there'

In this example, the consonant cluster is simplified by deleting the first consonant instead of the second. The problem with the previous positional faithfulness analysis with faithfulness for onsets is that the context was specified in the output. When this is the case, the contextual constraint is unviolated regardless of which consonant gets deleted because they both become onsets in the output, so a constraint demanding faithfulness to the onset position will be satisfied in both cases.

(29) Diola with output defined positional faithfulness (Wilson, 2001)
IDENT-*Onset*: Onset consonants must be faithful to their input correspondents.

Step 1: deletion

	/letkujaw/	IDENT-*Onset*	*VCCV	MAX
→ 1.	le.ku.jaw			*
→ 2.	le.tu.jaw			*
3.	let.ku.jaw		*W	L

With output defined contextual faithfulness, the faithfulness constraint cannot distinguish between the candidate which deletes the first consonant from the candidate which deletes the second consonant. Because the defined context of *Onset* is in the output and syllabification occurs during the derivation, both consonants are onsets in the output and no violations are assigned.

(30) Diola with input defined contextual faithfulness
MAX/_V: Assign one * for deletion of a consonant segment which is prevocalic in the input.

Step 1: deletion

	/letkujaw/	MAX/_V	*VCCV	MAX
→ 1.	lekujaw			*
2.	letujaw	*W		*
3.	letkujaw		*W	L

This system allows for the analysis of asymmetries based on segmental location in the input with constraints. The contextual faithfulness constraint places specific faithfulness on the second consonant in the cluster without requiring multi-level or targeted constraints. This is an example of a process involving local directionality. The rightmost consonant in the cluster is always preserved, and this is captured by creating a contextual faithfulness constraint which puts specific faithfulness on the desired segment. In Wilson (2001), targeted multi-level constraints are required to account for this phenomenon. In the input defined contextual faithfulness approach, the generalization is captured with only a faithfulness constraint and no additional technology is added to the system.

6.3 Harmony

Unlike the case of cluster simplification, long-distance directionality cannot be analyzed with contextual faithfulness. Since contextual faithfulness demands faithfulness in a specific position, it might seem that they could be added to a harmony analysis as a mechanism of controlling directionality of harmony, but this is not the case. These constraints can only specify local input contexts, and they cannot control directionality outside of a local context. Because of the assumptions regarding autosegmental spreading in Serial Harmony (McCarthy, 2009), long-distance harmony is not easily analyzed. I provide an example showing that the addition of a contextual faithfulness constraint is not enough to control this type of directionality in a serial system.

Chumash sibilant harmony is an example of a harmony system with absolute long-distance directionality (Applegate, 1972). The rightmost sibilant determines the anteriority of all sibilants in the stem and the directionality is absolute without regards to morphological structure (examples given below). Even though the contextual faithfulness constraints can provide faithfulness for the rightmost sibilant in the stem, they cannot also demand total harmony to the rightmost sibilant.

(31) Chumash sibilant harmony
/s-api-tʃo-it/ → [ʃapitʃolit]
/s-api-tʃo-us/ → [sapitsolus]
/s-api-tʃo-us-waʃ/ → [ʃapitʃoluʃwaʃ]

Because the harmony is over the entire word, there are many places where the assimilation can begin, leading to multiple possible derivations for each input. If the assimilation begins with the rightmost sibilant spreading to the left, the derivation will converge on the correct output. If the derivation begins with the spreading of another sibilant, the derivation will converge in a local maximum which is not the fully harmonized output. The contextual faithfulness constraints cannot distinguish between these two possibilities because they can only specify a context local to the segment requiring specific faithfulness (in this case, the rightmost sibilant). There is no way to control for which segment begins spreading first via contextual faithfulness. With the addition of the contextual faithfulness constraint, it is still the case that some of the derivation paths converge on the correct output, and some end up in a local maximum. I briefly detail the problem of the local maximum here. Additional details of the analysis are given in the appendix.

The derivation path resulting in the local maximum is presented in (32). The segments are represented with their feature specifications for anterior as subscripts. Linkage to the same token of [anterior] is indicated by parentheses. The crucial problem results from the fact that in Serial Harmony, delinking from one autosegment and linking to another is not considered one change and must be done in two steps. This results in segments which have (temporarily, in intermediate steps of the derivation) more than one specification for the same feature.

(32) Local maximum from /s_+ $ʃ_-$ $s_+ʃ_-$/

Pass	Operation	Output
0	input	s_+ $ʃ_-$ $s_+ʃ_-$
1	link [+ant] to ʃ	s_+ $(ʃ_-\ s)_+ʃ_-$
2	link [+ant] to s	$(s_+\ ʃ_-\ s)_+ʃ_-$
3	link [-ant] to s	$(s_+\ ʃ_-\ (s)_+ʃ)_-$
4	delink extra [+ant] from	$(s\ ʃ_-\ (s)_+ʃ)_-$
5	delink [-ant] from ʃ	$(s\ s\ (s)_+ʃ)_-$
6	local maximum	$(s\ s\ (s)_+ʃ)_-$

At Step 6 in this derivation, in order to satisfy the constraint demanding one feature specification per segment, a violation of SHARE must be created. Because SHARE has to outrank ONESPEC in earlier steps to compel spreading at all, this results in a local maximum.

(33) Step 6 - local maximum

$(s\ s\ (s)_+ʃ)_-$	Ident(ant)/_#	Share(ant)	Sp-L(ant)	Sp-R(ant)	OneSpec	Id(ant)
→ (1) $(s\ s\ (s)_+ʃ)_-$ faithful					*	
(2) $(s\ s)_+\ (ʃʃ)_-$ [+] delinks		*				*

Candidate 2, which removes the extra specifications, creates a violation of the spreading constraint. The contextual faithfulness constraint cannot distinguish between these two candidates and cannot prefer the candidate which removes the superfluous specifications. Even though the constraint provides specific faithfulness to the segment which controls the anteriority of the entire word, this type of faithfulness does not solve the problem of the local maximum in this framework.

6.4 Interim discussion: directionality

Contextual faithfulness constraints can capture local directionality of processes that cannot be attributed to other mechanisms. The form of the constraint allows for the specification of a context which cannot otherwise be inferred (as in the case of prosodically prominent positions, for example). Because the context specified must be local to the segment on which faithfulness is demanded, these constraints can only capture local directionality. Long-distance directionality, as in the case of Chumash sibilant harmony, cannot be controlled by contextual faithfulness to the rightmost sibilant.

Local directionality effects appear in Harmonic Serialism, but unlike the rule ordering effects, these effects are not specifically reliant on the use of a serial framework. The ability to produce rule ordering effects with contextual faithfulness relies on the fact that a context can be made which changes at a particular stage of the derivation. In the cases of local directionality, there is no need to reference an intermediate stage of the derivation. The contextual faithfulness constraints used here for local directional processes can also be implemented in a parallel OT framework.

7 Discussion

I have demonstrated that contextual faithfulness constraints can be used to analyze overapplication opacity and local directionality. Other than locality, there is no (internal to the theory) restriction on what possible contexts specified in these constraints can be. This lack of restriction on possible contexts permits the analysis of more phenomena than a system with restricted contexts like positional faithfulness, where contexts are restricted to a few prosodically prominent positions. The contextual faithfulness system can analyze phenomena which cannot be attributed to prosodic prominence. For example, the rule ordering effects analyzed in this paper utilize contextual faithfulness which is not tied to a position of prosodic prominence. The ability to define a context specifically related to the relevant markedness constraint is what allows the analysis to capture the rule ordering effects.

Because the contexts are less restricted, these constraints cause a broadening of typological predictions. In positional faithfulness, there is crucially no constraint IDENT-*Coda* which is complementary to IDENT-*Onset*. The theory only allows special faithfulness to certain prominent positions. In the contextual faithfulness system, because the additional contexts cannot be as restricted, there is nothing internal to the theory that rules out constraints which are complementary to the ones I propose in this paper. For example, the typological generalization that VCCV clusters are usually simplified by deleting the second consonant can be captured with the constraint MAX/_V. Currently there is nothing in the theory of contextual faithfulness which rules out a constraint which would demand deletion of the first consonant instead of the second consonant, MAX/_C. In this system, the explanation for the frequency of second consonant deletion can no longer be that CON does not include appropriate constraints that demand the alternative. Instead, some additional explanation beyond the scope of contextual faithfulness theory is needed.

Another potentially pathological prediction of the fairly unrestricted contexts of contextual faithfulness is spurious faithfulness patterns in various segmental positions. It is the case that specific faithfulness is needed for processes not related to prosodic prominence but it is not the case that this faithfulness could be used by a phonological grammar in every possible location. If contextual faithfulness constraints can be made which refer to any local context, specific faithfulness for seemingly random positions could be predicted.

For example, in the Bedouin Arabic case, I use the constraint MAX(i)/k_. This constraint prevents deletion of /i/ following unpalatalized voiceless consonants. This constraint in particular does not make pathological typological predictions. If the constraint were surface true in a language and not accompanied by active constraints for counterbleeding, the language would only allow deletion of /i/ after palatalized consonants. This would simply be a language with consonant vowel coalescence. In this case, there is a functional explanation for the existence of this constraint: the /i/ can be deleted, but only if its perceptual cues have been transferred to the preceding consonant via palatalization.

While MAX(i)/k_ is not necessarily a problem for typology, there is nothing internal to contextual faithfulness theory to rule out the existence of similar constraints like MAX(a)/C_. If such a constraint exists, there can be no functional explanation related to cue preservation. Adding a constraint like this to our typology would result in predicted languages with special faithfulness to /a/ following a consonant. These very specific constraints are necessary to analyze individual processes but adding them to the typology causes wide over-generation. This is problematic when considering CON to be a universal constraint set. In order to be typologically viable, contextual faithfulness constraints must either have more restrictions on possible contexts or exist in a system which constrains typology via some other mechanism.[4]

8 Conclusion

I have demonstrated that implementing contextual faithfulness constraints in Harmonic Serialism produces otherwise difficult to analyze rule ordering and directionality effects. The use of a serial framework combined with the ability to specify segmental contexts effectively allows the constraint set to demand a certain order of operations. Because Harmonic Serialism includes multiple passes through the same constraint set, contexts which are only available at an intermediate stage can be referred to via contextual faithfulness. This essentially renders the faithfulness constraint inactive when the context is either not yet or no longer available, allowing for the analysis of various rule ordering effects. Such effects have been problematic for constraint based grammars, which in standard form have no way to order application of processes. In addition, the relatively unrestricted ability to specify a segmental context for specific faithfulness provides a system to control otherwise unexplained patterns of directionality.

In order to analyze these phenomena, there must be few restrictions on the specified contexts in these constraints. This creates typological over-generation of possible faithfulness constraints and languages. Allowing any context to be specified provides the necessary freedom to capture specific contexts (like those pertaining to ordering of operations in counterbleeding), but also permits the existence of complementary constraints which do not reflect broader typological generalizations or any kind of phonetic grounding. Because of this, these constraints should be paired with another theory which constrains typology that is external to CON itself. Exactly how such a pairing would be implemented is a direction for future work.

[4] Having many possibilities for defining contexts in faithfulness constraints might not be a problem when combined with a theory of constraint induction, for example. Operations for creating faithfulness constraints would include specifying a local context, resulting in a more complex constraint. Since more complex constraints would be harder to learn that could offer an explanation for the relative rarity of these constraints typologically. This sort of typological explanation is beyond the scope of this paper but is a direction for future work.

References

Al-Mozainy, H. Q., 2007. Vowel Alternations in a Bedouin Hijazi Arabic Dialect: Abstractness and stress. Ph.D. thesis, University of Texas, Austin.

Allen, W. S., 1962. Sandhi, vol. 17. Mouton.

Applegate, R., 1972. Ineseño Chumash Grammar. Ph.D. thesis, University of California.

Baković, E., 2011. Opacity and Ordering. In: Goldsmith, John A and Riggle, Jason and Alan, CL (Ed.), The Handbook of Phonological Theory. John Wiley & Sons.

Beckman, J. N., 1997. Positional faithfulness, positional neutralisation and Shona vowel harmony. Phonology 14, 1–46.

Hualde, J. I., de Urbina, J. O., 2003. A Grammar of Basque. Walter de Gruyter.

Kenstowicz, M., 1994. Phonology in generative grammar. Blackwell Oxford.

Lombardi, L., 1999. Positional faithfulness and voicing assimilation in Optimality Theory. Natural Language & Linguistic Theory 17, 267–302.

McCarthy, J. J., 2007. Hidden generalizations: Phonological opacity in Optimality Theory. Equinox.

McCarthy, J. J., 2009. Harmony in Harmonic Serialism. Ms, University of Massachusetts Amherst. Available as ROA-1009 from the Rutgers Optimality Archive.

McCarthy, J. J., 2010. An introduction to Harmonic Serialism. Language and Linguistics Compass 4, 1001–1018.

Newman, S. S., 1944. Yokuts language of California. Viking Fund.

Sapir, J. D., 1965. A grammar of Diola-Fogny. Cambridge University Press.

Shaw, P. A., 1991. Consonant harmony systems: the special status of coronal harmony. Paradis and Prunet , 125–57.

Steriade, D., 2001. Directional asymmetries in place assimilation: A perceptual account. The Role of Speech Perception in Phonology , 219–250.

Wilson, C., 2001. Consonant cluster neutralisation and targeted constraints. Phonology 18, 147–197.

Appendix

A: Yawelmani counterbleeding

In the following tableaux, I show that the suffixes do not necessarily have to be underlyingly specified as unround for this analysis. I provide the same tableau as in the main text but with underlying /u/ in the suffixes instead of underlying /i/. Because I have defined the contextual faithfulness constraint preventing lowering from occurring before rounding harmony in terms of the context of non-identical specifications for rounding, it does not matter whether the suffix is underlyingly round or unround for the counterbleeding interactions. If harmony has applied, the preceding vowel and the suffix vowel will have identical specifications for [round] and the context will no longer be available so lowering is free to apply. If the input suffix vowel is /u/ instead of /i/ and already round, the specifications for round in /cu:m-hun/ will already be identical and lowering is free to apply at Step 1.

(34) /mi:k-hun/ → [mekhin]

This derivation becomes a double counterbleeding interaction when the suffix contains a round vowel instead of an unround vowel in the input. Rounding harmony must apply so that the suffix vowel and root vowel can match in rounding. After the two vowels match in rounding, lowering and closed syllable shortening can apply.

Rankings

- Id(long)/[hi] ≫ $*[\mu\mu\mu]_\sigma$ ≫ Id(long)
- *[long][hi] ≫ Id(hi)

Step 1: rounding harmony

/mi:k-hun/	Id(long)/[hi]	$*[\mu\mu\mu]_\sigma$	Id(long)	Id(hi)/[αrd][βrd]	*[long][hi]	Id(hi)	Round	Id(rd)
→1. mi:khin		*			*			*
2. mi:khun		*			*		*W	L
3. me:khun		*		*W	L	*W		L
4. mikhun	*W	L	*W				*W	L

Step 2: lowering

mi:k-hin	Id(long)/[hi]	$*[\mu\mu\mu]_\sigma$	Id(long)	Id(hi)/[αrd][βrd]	*[long][hi]	Id(hi)
→ 1. me:khin		*				*
2. mikhin	*W	L	*W			L
3. mi:khin		*			*W	L

Step 3: closed syllable shortening

me:khin	Id(long)/[hi]	$*[\mu\mu\mu]_\sigma$	Id(long)	*[long][hi]	Id(hi)
→ 1. mekhin			*		
2. me:khin		*W	L		

(35) /cu:mut/ → [co:mut]

When the input suffix contains a round vowel in /cu:mut/, the derivation becomes transparent. Lowering is the only process which applies.

Step 1: lowering

cu:mut	ID(hi)/[αrd][βrd]	*[long][hi]	ID(hi)
→ 1. co:mut			*
2. cu:mut		*W	L

(36) /cu:m-hun/ → [comhun]

This derivation (which results in the double counterbleeding interaction when the input suffix contains an /i/) is now only a counterbleeding interaction between lowering and closed syllable shortening.

Step 1: lowering

cu:mhun	ID(long)/[hi]	*[μμμ]$_\sigma$	ID(long)	*[long][hi]	ID(hi)
→ 1. co:mhun		*			*
2. cu:mhun		*		*W	L
3. cumhun	*W	L	*W		

Step 2: closed syllable shortening

co:mhun	*[μμμ]$_\sigma$	ID(long)
→ 1. comhun		*
2. co:mhun	*W	L

B: More details of harmony analysis

Full constraint definitions:
SHARE(F) (McCarthy, 2009)
Assign one violation mark for every pair of adjacent segments that are not linked to the same token of [F].

INITIAL(F) (McCarthy, 2009)
Let input F tier = $f_1 f_2 \ldots f_m$.
Let input segmental tier = $s_1 s_2 \ldots s_n$
Let output F tier = $f_1 f_2 \ldots f_0$.
Let output segmental tier = $s_1 s_2 \ldots s_p$.
Assign one violation mark for every $s_i \Re s_j$, where:
$f_k \Re f_l$,
f_k is associated with s_i,
and there is no s_x that precedes s_i and is also associated with f_k,
and f_l is associated with s_j,
and there is some s_y that precedes s_j and is also associated with f_l.

FINAL(F)
Let input F tier = $f_1 f_2 \ldots f_m$.

Let input segmental tier = $s_1 s_2 \ldots s_n$
Let output F tier = $f_1 f_2 \ldots f_0$.
Let output segmental tier = $s_1 s_2 \ldots s_p$.
Assign one violation mark for every $s_i \Re s_j$, where:
$f_k \Re f_l$,
f_k is associated with s_i,
and there is no s_x that follows s_i and is also associated with f_k,
and f_l is associated with s_j,
and there is some s_y that follows s_j and is also associated with f_l.

Analyzing Chumash with Serial Harmony:
There are three key assumptions of the Serial Harmony: privative features, harmony being compelled by Share(F) (violated by two adjacent segments not linked to the same autosegment), and a serial framework. The assumptions about GEN only allow for insertion or deletion of a single association line at once, or insertion/deletion between two pieces of existing structure. Because linking and delinking cannot happen at the same time, markedness constraints must be used to cause delinking.

The assumption that privative features be used is not crucial to the workings of the analysis. Serial Harmony can be used, and the constraints evaluated in the same way, with binary features as well. The original motivation for the use of privative features in Serial Harmony was that there are many arguments for using privative features in phonology generally, and many harmony processes can be analyzed as spreading of a single privative feature to other underspecified segments. However, the Chumash data have typically been analyzed as spreading both values of the feature [anterior] (Shaw, 1991). The anteriority of all sibilants in the word is determined by the anteriority of the rightmost sibilant. It is possible to alternatively analyze this as the actions of two independent privative features, [anterior] and [distributed], but this complicates the analysis. In the Serial Harmony framework, both the analyses of Chumash with these two privative features and the binary features work in the same way.

If we keep the original assumption from Serial Harmony that spreading feature are privative, we can assume /ʃ/ spreads [dist] and /s/ spreads [ant]. Therefore two of each constraint given below will be necessary, for example, the analysis needs to include both Share[ant] and Share[dist]. I give a possible derivation for /s-s-s-ʃ/ → [ʃʃʃʃ] in 8. This derivation starts with the rightmost sibilant spreading [dist] to the left. This is one of three possible derivations that will end up converging on the same correct output in this analysis. In the table, [s] indicates linkage to [ant], [θ] indicates linkage to both [ant] and [dist], [ʃ] indicates linkage to [dist] and [-] indicates a Share violation.

(37) Possible derivation for /s-s-s-ʃ/ → [ʃʃʃʃ]

Pass	Operation	Output
1	link [dist] to s	s-s-θʃ
2	link [dist] to s	s-θθʃ
3	link [dist] to s	θθθʃ
4	delink [ant] from s	θθʃʃ
5	delink [ant] from s	θʃʃʃ
6	delink [ant] from s	ʃʃʃʃ
7	convergence	ʃʃʃʃ

Because this derivation starts with spreading from the rightmost sibilant, the harmony completes. In order to compel spreading at all at Step 1, Share has to outrank both Ident constraints and the markedness constraint *[ant][dist]. Because of this ranking, all spreading must take place before any delinking can occur. This example spread the feature [dist] but a complementary example /ʃ-ʃ-ʃ-s/ → [s-s-s-s] would need to spread [ant]. In order to account for both of these data, each pair of constraints differing only in the feature [ant] vs. [dist] must be unranked with respect to each other. With this exception, all rankings needed for this analysis will be the same as in the binary feature analysis explored in the next section.

Using binary [anterior] works in the same way and results in the same problems with local maxima. A possible derivation is provided in (38). This derivation begins with spreading anteriority left from the rightmost sibilant, and converges on the correct output.

(38) Possible derivation for /s_+ $ʃ_-$ $s_+ʃ_-$/ → [$(ʃʃʃʃ)_-$]

Pass	Operation	Output
1	link [-ant] to s	s_+ $ʃ_-$ $(s_+ʃ)_-$
2	link [-ant] to ʃ	s_+ $(ʃ_-$ $s_+ʃ)_-$
3	link [-ant] to s	$(s_+$ $ʃ_-$ $s_+ʃ)_-$
4	delink [+ant] from s	$(ʃʃ_-$ $s_+ʃ)_-$
5	delink [-ant] from ʃ	$(ʃʃs_+ʃ)_-$
6	delink [+ant] from s	$(ʃʃʃʃ)_-$
7	convergence	$(ʃʃʃʃ)_-$

In this derivation, a contextual faithfulness constraint can ensure that the rightmost sibilant remains faithful and does not harmonize with the rest of the word. Because spreading starts from the rightmost sibilant, contextual faithfulness is enough to control direction of spreading in this particular derivation. When spreading starts from another location on the sibilant tier, the contextual faithfulness constraint demanding rightmost sibilant faithfulness is still unviolated, but the derivation will not converge on the correct output. This derivation was shown in the main text.

I provide the tableau for Step 1 of a derivation with input /s_+ $ʃ_-$ $s_+ʃ_-$/ in (40). At the first pass, candidates 1 and 4 tie as equally optimal on this constraint set. Because both of the candidates spread left and neither violates the contextual faithfulness constraint, either candidate can win. In theory, this would not necessarily be a problem if both derivation paths converged on the correct output, but this is not the case. If candidate 1 is chosen (spreads from rightmost sibilant) the derivation will continue to full harmony as in (38). If candidate 4 is chosen (spreads from another sibilant), the derivation will end in a local maximum as in (32).

(39) Constraints

- IDENT(F)/_#: Assign one * for insertion, deletion, or re-linking of [F] from the word final segment on the sibilant tier.
- SHARE(F): Assign one * for every pair of adjacent segments not linked to the same token of F.
- FINAL(F): Faithfulness constraint that punishes leftward spreading of [F].
- INITIAL(F): Faithfulness constraint that punishes rightward spreading of [F].
- IDENT(F): Assign one * for insertion, deletion, or re-linking of [F].

- *[ant][dist]: Assign one * for every segment that is specified both [ant] and [dist].

(40) Beginning of a possible derivation for /s-api-tʃo-us-waʃ/ → [ʃapitʃoluʃwaʃ]

Step 1: candidates 1 and 4 tie

	/s₊ʃ₋s₊ʃ₋/	Id(ant)/_#	Share(ant)	Initial(ant)	Final(ant)	OneSpec	Id(ant)
→ 1.	s_+ $ʃ_-$ $(s_+ʃ)_-$ [-] spreads L		**		*	*	*
2.	s_+ $ʃ_-$ s_+ $ʃ_-$ faithful		***W		L	L	L
3.	s_+ $ʃ_-$ $(s\ ʃ_-)_+$ [+] spreads R	*W	**	*W	L	*	*
→ 4.	$s_+(ʃ_-s)_+ʃ_-$ [+] spreads L		**		*	*	*

Here, the contextual faithfulness constraint does rule out candidate 3, which spreads to the rightmost sibilant. However, this candidate can also be ruled out by the faithfulness constraint Initial(ant). While it might seem that the contextual faithfulness constraint, by demanding specific faithfulness to the rightmost sibilant can control directionality of harmony from that sibilant, this is not the case. The problematic winner is candidate 4, which along with the desired winner, candidate 1, does not violate the contextual faithfulness constraint. Simply demanding faithfulness for the harmony trigger will not be enough to control long-distance directionality across the entire word.