

## RESEARCH

# Analyzing opacity with contextual faithfulness constraints

Ivy Hauser<sup>1</sup> and Coral Hughto<sup>2</sup><sup>1</sup> University of Texas at Arlington, Arlington, TX, US<sup>2</sup> University of Massachusetts Amherst, Amherst, MA, USCorresponding author: Ivy Hauser ([ivy.hauser@uta.edu](mailto:ivy.hauser@uta.edu))

Phonological opacity is well-studied and there are numerous proposals in the literature which analyze opacity in Optimality-Theoretic grammars. However, many analyses include significant elaborations to the basic architecture of Optimality Theory (Prince & Smolensky 1993/2004) or its serial alternative, Harmonic Serialism (McCarthy 2000). In this paper, we propose a method of analyzing opacity which avoids additional significant enhancements to the basic theory by using faithfulness constraints with input-defined contexts. These constraints bear many similarities to standard positional faithfulness constraints (Beckman 1997; 1998; Lombardi 1999), but the context is input-defined (as in Jesney 2011). Adding context to faithfulness constraints has previously been discussed as a potential solution to counterfeeding opacity, but dismissed on account of potentially creating an overly rich faithfulness theory (McCarthy 2007a). We argue that the analytical potential of these constraints outweighs the potential problems associated with overgeneration. We show that contextual faithfulness constraints can be employed to analyze multiple types of underapplication opacity in parallel OT and multiple types of under- and over-application opacity in Harmonic Serialism. We discuss the impact of including these constraints in a universal CON and suggest the potential of language-specific constraint induction for mitigating over-generation effects.

**Keywords:** opacity; Optimality Theory; faithfulness; Harmonic Serialism

## 1 Introduction

Many cases of opacity are classically difficult to analyze in standard parallel Optimality Theory (OT). There is a wealth of literature offering potential analyses of various opaque phenomena, (e.g. McCarthy 1999; Kiparsky 2000; Bermúdez-Otero 2003; Ito & Mester 2003; Kiparsky 2003; McCarthy 2007a; Jarosz 2014) many of which include significant elaborations to the basic architecture of Optimality Theory (OT; Prince and Smolensky 1993/2004) or Harmonic Serialism (HS; Prince & Smolensky 1993/2004; McCarthy 2000). In this paper, we propose a method of analyzing opacity which uses input-defined contextual faithfulness constraints. These constraints bear many similarities to standard positional faithfulness constraints (Beckman 1997; 1998; Lombardi 1999) with two crucial differences: the context refers to the input rather than the output (as in Jesney 2011), and possible contexts are not restricted to prominent positions. Adding context to faithfulness constraints is not new, but has never been proposed as a general solution for analyzing opacity. As faithfulness constraints are part of the basic architecture of OT, this proposal avoids adding major formal enhancements to the theory.

We examine the use of contextual faithfulness constraints in both parallel OT and HS. In parallel OT, input-defined contextual faithfulness constraints can be used to analyze multiple types of underapplication opacity. However, when implemented in HS, contextual

faithfulness constraints can be used to analyze both underapplication and overapplication opacity, with the addition of a distinction between faithfulness to the input of the current step of the derivation and faithfulness to the underlying representation. In addition, implementing faithfulness constraints with defined contexts in HS instead of parallel OT avoids previously documented pathologies associated with positional faithfulness constraints (Jesney 2011).

### 1.1 Types of opacity

Phonological opacity can be divided into two types: underapplication opacity, and overapplication opacity. Underapplication opacity refers to a generalization that is not surface-true, meaning that there are some surface forms in the language which meet the structural description of the process, but it has not applied. Overapplication opacity refers to a situation where the motivation for process application is not surface-apparent, meaning that there are some surface forms to which the process has applied, even though its structural description has seemingly not been met (Kiparsky 1973; paraphrased in Baković 2007).

The most frequently discussed cases of opacity are produced by counterfeeding (underapplication) and counterbleeding (overapplication) rule orders. These rule orders are often assumed to produce a natural class of opaque phenomena. However, Baković (2007) argues that these categorizations do not sufficiently characterize all of the observed opaque patterns, and proposes a revised typology of opaque interactions. We provide a brief sketch of this typology, focusing on the types of opacity we analyze in this paper.

#### 1.1.1 Underapplication

The most frequently discussed forms of underapplication opacity are produced by counterfeeding rule orders, defined in 1. Counterfeeding rule orders produce two types of underapplication opacity: counterfeeding on focus and counterfeeding on environment.

(1) Definition of counterfeeding

For two ordered rules  $\mathbb{A}$  and  $\mathbb{B}$ , where  $\mathbb{A}$  precedes  $\mathbb{B}$  in order of application,  $\mathbb{B}$  counterfeeds  $\mathbb{A}$  iff the output of  $\mathbb{B}$  meets the context of application for  $\mathbb{A}$ , but  $\mathbb{A}$  does not apply due to order of application (Kiparsky 2000).

In counterfeeding on focus, the two rules involved both apply to the same segment (e.g. chain shifts). This type of interaction is relatively easy to analyze in parallel OT, and there have been many proposals in the literature which invoke faithfulness constraints to block a second process from applying to a particular segment (e.g. Kirchner 1996; Gnanadesikan 1997; Moreton & Smolensky 2002; Jesney 2005).

In counterfeeding on environment, the two rules involved do not apply to the same segment. Therefore, creating a faithfulness constraint to prevent the second process from applying is more complicated. Sympathy Theory (McCarthy 1999), targeted constraints (Wilson 2001), local constraint conjunction (Moreton & Smolensky 2002), and OT with Candidate Chains (McCarthy 2007a) have been proposed as possible solutions which can account for both types of counterfeeding. Our proposed analysis accounts for both types of counterfeeding in parallel OT and HS. We further discuss how our analysis relates to these and other previous approaches in §4.2.

Our proposal is intended to provide analyses for counterfeeding on focus and counterfeeding on environment. Types of underapplication opacity that we do not aim to account for include: class/level restrictions, optionality, exceptionality (see Baković 2011a for

discussion of these phenomena as underapplication), and fed counterfeeding (see §2.4 for discussion of why our proposal does not account for fed counterfeeding).

### 1.1.2 Overapplication

The most commonly discussed type of overapplication opacity is produced by counterbleeding rule orders, where a rule appears to have applied even though its structural description is not met.

#### (2) Definition of counterbleeding

For two ordered rules  $\mathbb{A}$  and  $\mathbb{B}$ , where  $\mathbb{A}$  precedes  $\mathbb{B}$  in order of application,  $\mathbb{B}$  counterbleeds  $\mathbb{A}$  iff  $\mathbb{B}$  eliminates potential inputs to  $\mathbb{A}$  (Kiparsky 2000).

Counterbleeding is difficult to analyze in OT because two processes must apply when applying only one would be sufficient to satisfy the relevant markedness constraints. To illustrate, suppose that applying process  $A$  satisfies some markedness constraint  $M_A$ , and applying process  $B$  satisfies some markedness constraint  $M_B$ . If applying process  $B$  removes the structure that triggers process  $A$ , then it also removes the violation of  $M_A$ . Thus, applying process  $B$  alone is sufficient to satisfy both  $M_A$  and  $M_B$ , and applying process  $A$  in addition would only incur another, gratuitous violation of a faithfulness constraint (this explanation adapted from Jarosz 2014).

In OT, local constraint conjunction (Smolensky 1995; Ito & Mester 2003) has been used to analyze overapplication effects (e.g. Łubowicz 2002). Other approaches make use of output-output faithfulness (Burzio 1994; Benua 1997). We discuss further details of these approaches and how our analysis compares with them in §4.2.

There is usually no distinction made in the literature between counterbleeding on environment and counterbleeding on focus, and the two types may be logically equivalent (Baković 2011a). To our knowledge, only one example of counterbleeding on focus has been reported (Kiparsky 1968; Baković 2011a). Baković (2007) proposes categories (and OT analyses) for additional overapplication interactions including self-destructive feeding, gratuitous feeding, and cross-derivational feeding, which are not problematic for OT grammars. Our analysis is only intended to account for cases of overapplication which involve a “gratuitous violation of a faithfulness constraint” (Baković 2007) and we consider counterbleeding on environment as an example case.

## 1.2 Context in faithfulness constraints

Adding context to faithfulness constraints, in the form of positional faithfulness constraints, has been explored in detail by Beckman (1997) and Lombardi (1999), among others. Positional faithfulness constraints are intended to capture various phonological asymmetries by indexing faithfulness to particular prominent 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 may also trigger phonological processes. Positional faithfulness constraints typically specify output contexts in order to refer to prosodic positions which would not be specified in the input, such as onset position. Beckman (1997) provides a list of prominent positions which have some perceptual advantage requiring special faithfulness: root-initial syllables, stressed syllables, syllable onsets, roots, and long vowels.

Although the constraints that we propose in this paper are similar to positional faithfulness constraints, they have a few crucial differences, and thus to differentiate them

we have termed them *contextual faithfulness* constraints. Following Jesney (2011), these constraints specify an input context instead of an output context. This provides crucial distinctions not available with output contexts.<sup>1</sup> In addition, unlike traditional positional faithfulness constraints, contextual faithfulness constraints are not restricted to positions of prominence.

Faithfulness constraints with a specified context or focus have been discussed as a method of analyzing counterfeeding opacity in parallel OT (McCarthy 2007a). McCarthy shows how using a constraint which incurs violations only for deletion of certain segments is sufficient to analyze counterfeeding in parallel OT.<sup>2</sup> He does not, however, advocate this as a general solution to counterfeeding opacity, because enumerating all possible constraints of this type in CON would produce a faithfulness theory which is richer than necessary. McCarthy argues that this is a fatal flaw for the approach, and that using faithfulness constraints specified for certain segments should not be pursued as a general solution to counterfeeding opacity.

We argue in this paper that the contextual faithfulness approach McCarthy considers should not be immediately dismissed over concerns of an overly rich CON. Further investigation has shown that specifying context in faithfulness constraints can potentially act as a general solution to underapplication in parallel OT, and to multiple types of opacity in HS. While this proposal does require an enriched faithfulness theory, there are many ways that typological prediction can be constrained which do not limit the constraint set itself. For example, language-specific constraints can be learned via an induction algorithm and therefore never enter the factorial typology. We discuss the potential for constraint induction in §4, and show the factorial typologies which would result if the constraints we propose were to be included in a universal CON.

## 2 Parallel analysis

In this section, we define the template for constructing contextual faithfulness constraints (§2.1) and demonstrate how they can be used to analyze multiple types of underapplication opacity in parallel OT (§2.2), using example patterns to illustrate counterfeeding on focus and counterfeeding on environment. In §2.3, we show how contextual faithfulness constraints cannot be used to analyze overapplication opacity in parallel OT, leading into our analyses of these patterns in HS (§3).

### 2.1 Constraint definitions

Contextual faithfulness constraints have two crucial elements: the feature for faithfulness (F), and the context for faithfulness (G). We will use the terms *focus* to refer to the segment/class where this faithfulness constraint applies, and *context* to refer to the context for faithfulness. The context is always input-defined, and can refer to a property of the local environment ( $/\_[\alpha G]$ ) or a property of the segment itself ( $/\overline{[\alpha G]}$ ).<sup>3</sup> In §4.1.2, we propose that these constraints could be constructed by combining properties of relevant pre-existing markedness constraints, and thus would inherit from them restrictions on scope and domain.

<sup>1</sup> In order to analyze opacity with contextual faithfulness constraints, the contexts must be defined as input contexts. However, there are some candidates for which the input context and the output context would be identical. These examples do not point to the use of output contexts, because other cases of opacity show that input contexts are crucial for analyzing opacity with contextual faithfulness.

<sup>2</sup> For an example of this type of analysis and further discussion of implications for CON and typology see McCarthy (2007a: 25–27). We pursue a contextual faithfulness analysis of the same data he considers in §2.2.

<sup>3</sup> The overbar notation is used as an analogy to the standard environment notation  $/\_[\alpha G]$ . The overbar indicates that the relevant environment is a property of the segment itself.

- (3) IDENT[F]/ $_{[\alpha G]}$ : Let  $A$  be a segment in some context  $_{[\alpha G]}$  in the input. Assign one violation if the output correspondent of  $A$  does not have the same specification for [F] as  $A$ .  
i.e. Do not change the value of F for segments that are in the context of  $_{[\alpha G]}$  in the input.
- IDENT[F]/ $\overline{[\alpha G]}$ : Let  $A$  be a segment specified for some feature [G] in the input. Assign one violation if the output correspondent of  $A$  does not have the same specification for [F] as  $A$ .  
i.e. Do not change the value of F for segments that are  $[\alpha G]$  in the input.

## 2.2 Underapplication analyses in parallel OT

### 2.2.1 Counterfeeding on focus

An example of counterfeeding on focus is found in Hijazi Bedouin Arabic (HB Arabic; Al-Mozainy 1981).<sup>4</sup> There are two relevant processes: vowel raising and syncope. Underlying short /a/ raises to [i] in non-final open syllables, and high vowels delete in non-final open syllables. This interaction can be analyzed with a sequence of rules. The rules and their definitions are given in 4.

- (4) *HB Arabic* (Al-Mozainy 1981; McCarthy 2007a)  
Rules for counterfeeding on focus interaction
- i. Raising: [a] → [+high] /  $_{CV}$
  - ii. Syncope: [+high] →  $\emptyset$  /  $_{CV}$

This interaction can be analyzed in a rule-based framework by ordering the syncope rule before the raising rule, as shown in the rule derivation in 5. High vowels which were raised from low vowels are not subject to deletion because that rule applied earlier in the derivation. This can be seen with the output [difaʕ], which contains the environment for syncope, but the rule has not applied.

- (5) HB Arabic counterfeeding derivations
- |         |             |             |
|---------|-------------|-------------|
| UR      | /dafaʕ/     | /ʃaribat/   |
| syncope | -           | ʃarbat      |
| raising | difaʕ       | -           |
| SR      | [difaʕ]     | [ʃarbat]    |
|         | ‘he pushed’ | ‘she drank’ |

A standard OT analysis using the constraints given in 6 cannot capture the counterfeeding pattern. High vowel syncope is motivated by a markedness constraint against high vowels in open syllables (\*iCV) ranked above MAX, as shown in 7. However, attempting to account for vowel raising in the same way, by ranking the relevant markedness constraint (\*aCV) above IDENT[low], predicts /dafaʕ/ → \*[dfaʕ], not the attested [difaʕ], as illustrated in the tableau in 8. This is because the intended output [difaʕ] still violates the constraint motivating syncope (\*iCV), which must be ranked above IDENT[low]. The two tableaux in 7 and 8 present a ranking contradiction: the crucial ranking for [ʃarbat] to win is \*iCV ≫ MAX, but the crucial ranking for [difaʕ] to win would be MAX ≫ \*iCV.

<sup>4</sup> This is the same example analyzed by McCarthy (2007a), in which he provides but dismisses a similar contextual faithfulness analysis. See §1.2 for discussion of why we pursue that style of analysis here.

- (6) HB Arabic counterfeeding constraints for parallel OT (McCarthy 2007a)
- \*iCV: Assign one violation for every high vowel in a nonfinal open syllable.
  - \*aCV: Assign one violation for every low vowel in a nonfinal open syllable.
  - IDENT[low]: Assign one violation for every output segment whose corresponding input segment does not have identical specification for the feature [low].
  - MAX: Assign one violation for every input segment which does not have a correspondent in the output.

- (7) HB Arabic counterfeeding: high vowel deletion (transparent) in standard parallel OT

	/ʃaribat/	*aCV	IDENT[low]	*iCV	MAX
→ 1.	ʃarbat				*
2.	ʃaribat			*W	L
3.	ʃarabat	*W	*W		L

- (8) HB Arabic counterfeeding: vowel raising (opaque) in standard parallel OT<sup>5</sup>

	/dafaʕ/	*aCV	IDENT[low]	*iCV	MAX
→ 1.	difaʕ		*	*	
2.	dafaʕ	*W	L	L	
☛ 3.	dfaʕ			L	*W

Adding a contextual faithfulness constraint of the type we define in §2.1 can resolve the ranking paradox, allowing for the analysis of this counterfeeding pattern in parallel OT. The necessary constraint prevents underlying low vowels from raising to high (defined in 9).

- (9) MAX/[+low]: Let A be a segment specified [+low] in the input. Assign one violation if A does not have an output correspondent.  
i.e. Do not delete segments that are [+low] in the input.

In the case of vowel raising, shown in 10, ranking \*aCV above IDENT[low] motivates raising an underlying low vowel to high, while ranking the contextual faithfulness constraint above \*iCV prevents underlying low vowels from deleting.

- (10) HB Arabic counterfeeding: vowel raising (opaque) with contextual faithfulness

	/dafaʕ/	MAX/[+low]	*aCV	IDENT[low]	*iCV	MAX
→ 1.	difaʕ			*	*	
2.	dafaʕ		*W	L	L	
3.	dfaʕ	*W		L	L	*W

- (11) HB Arabic counterfeeding: syncope (transparent) with contextual faithfulness

	/ʃaribat/	MAX/[+low]	*aCV	IDENT[low]	*iCV	MAX
→ 1.	ʃarbat					*
2.	ʃaribat				*W	L

<sup>5</sup> We use the arrow symbol in tableaux to indicate the intended winner. When the intended winner and the winner under the current ranking are not the same, we use the bomb symbol to indicate the incorrect winner under the current ranking.

The high-ranked contextual faithfulness constraint prevents the problematic candidate (3.) from winning by specifying faithfulness to a particular property in the input. In this case, the underlying low vowel receives a violation for deletion. This constraint only applies to underlying [+low] segments (because the context is input defined), therefore the analysis of syncope in 11 remains the same: \*iCV must be ranked above MAX. The underlying high vowel in the transparent case 11 is unaffected by the same constraint, because the domain of faithfulness is restricted to underlying low vowels.

### 2.2.2 Counterfeeding on environment

Counterfeeding on environment is similar to counterfeeding on focus, except the two interacting rules overlap in context instead of focus. An example of counterfeeding on environment comes from Lomongo (Hulstaert 1961). Intervocalic voiced obstruents delete, and hiatus violations are repaired with glide formation. However, hiatuses derived from intervocalic voiced obstruent deletion do not undergo glide formation. The rules and their definitions are given in 12. We use the feature [vocalic] to distinguish vowels and glides.

- (12) *Lomongo* (Hulstaert 1961, as summarized in Baković 2011a: 45)  
 Rules for counterfeeding on environment interaction
1. Glide Formation: [+vocalic, -consonantal] → [-vocalic] / \_ V
  2. Intervocalic Deletion: [+voice, -sonorant] → ∅ / V \_ V

This pattern can be analyzed in rule-based frameworks by ordering the glide formation rule before the intervocalic deletion rule. As illustrated by input /o-bina/ in 13, even though intervocalic deletion creates an environment in which glide formation could apply, the output surfaces as [oina] because the glide formation rule is ordered earlier in the derivation.

### (13) Lomongo rule derivations

	Transparent	Counterfeeding
UR	/o-isa/	/o-bina/
Glide Formation	wisa	–
Intervocalic Deletion	–	oina
SR	[wisa]	[oina]
	‘hide’	‘you.SG’

As with counterfeeding on focus, contextual faithfulness constraints can be used to analyze this interaction. The relevant constraints are given in 14 and include the contextual faithfulness constraint ID[voc]/\_VCEOBS, which prevents changes in the feature [vocalic] for segments which occur before voiced obstruents ([+voi, -son] segments, abbreviated by VCEOBS) in the input. Essentially, this constraint prevents glide formation for candidates which contain the appropriate context for obstruent deletion in the UR.

- (14) Lomongo counterfeeding constraints: parallel OT
- \*[+voi, -son]/V\_V: Assign one violation for a [+voi, -son] segment between two vowels.  
 Abbreviation: \*VCEOBS/V\_V
  - \*HIATUS: Assign one violation for two adjacent vowels.
  - MAX: Assign one violation for any input segment without an output correspondent.
  - IDENT[vocalic]: Assign one violation for an output segment whose input correspondent does not have the same value for [vocalic].

- ID[voc]/\_[+voi,-son]: Let *A* be a segment in the context preceding a [+voi,-son] segment in the input. Assign one violation if the output correspondent of *A* does not have the same specification for [voc] as *A*.  
i.e. Do not change the value of [ $\alpha$  vocalic] for segments that occur before [+voi,-son] in the input.  
Abbreviation: ID[voc]/\_VCEOBS

(15) Lomongo transparent interaction: /o-isa/ → w-isa

/o-isa/	ID[voc]/_VCEOBS	*VCEOBS/V_V	MAX	*HIATUS	ID[voc]
→ 1. wisa					*
2. oisa				*W	L

(16) Lomongo counterfeeding interaction: /o-bina/ → o-ina

/o-bina/	ID[voc]/_VCEOBS	*VCEOBS/V_V	MAX	*HIATUS	ID[voc]
→ 1. oina			*	*	
2. obina		*W	L	L	
3. wina	*W		*	L	*W

The tableau in 15 shows transparent glide formation for input /o-isa/. Glide deletion is motivated by ranking \*HIATUS above ID[voc], yielding output [wisa]. The tableau in 16 shows the opaque counterfeeding interaction for input /o-bina/. Intervocalic obstruent deletion is motivated by ranking the markedness constraint demanding deletion (\*VCEOBS/V\_V) above MAX, the relevant faithfulness constraint, and \*HIATUS. This rules out the faithful candidate [obina], because it does not delete the intervocalic obstruent.

The candidate which is problematic for standard parallel OT is candidate (3.), [wina]. This candidate undergoes both intervocalic deletion and glide formation, thereby satisfying both relevant markedness constraints (\*VCEOBS/V\_V and \*HIATUS). However, when ranked above these markedness constraints, the contextual faithfulness constraint rules out this candidate. This constraint essentially prevents gliding for segments which become prevocalic due to the deletion of a voiced obstruent. This is done by including the specified input context—the context specified in the constraint is the same context in which the intervocalic deletion would occur.

This analysis captures the counterfeeding interaction and does not interfere with the transparent interaction, shown in 15. The high ranking contextual faithfulness constraint is vacuously satisfied when the context and/or focus is not present in the input.

### 2.3 Overapplication is not analyzable in parallel OT

While contextual faithfulness constraints provide a general solution to analyzing under-application opacity in parallel OT, this analysis cannot be extended to overapplication opacity. To show this, we consider an example counterbleeding case. In HB Arabic, there is a palatalization rule, where velar consonants palatalize preceding [i], and a deletion rule, where the high front vowel [i] deletes in open syllables. Sample derivations are shown in 17.



- (17) HB Arabic counterbleeding example derivations  
Data summarized in McCarthy (2007a: 11–25) from Al-Mozainy (1981).

	Transparent	Transparent	Counterbleeding
UR	/ʃaribat/	/ħa:kim/	/ħa:kim-in/
Palatalization	–	ħa:kjim	ħa:kjimin
Deletion	ʃarbat	–	ħa:kjmin
SR	[ʃarbat]	[ħa:kjim]	[ħa:kjmin]
	‘she drank’	‘ruling.SG’	‘ruling.PL’

In the transparent cases, the palatalization and deletion processes apply independently of each other. The counterbleeding case is illustrated by the input /ħa:kim-in/. The palatalization rule applies first to [k], before the following [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 ordering palatalization before vowel deletion, parallel OT cannot capture this, with or without contextual faithfulness constraints. The tableau in 19 illustrates how a parallel OT analysis using the constraint set in 18 fails.

- (18) Constraint definitions: HB Arabic counterbleeding in parallel OT (McCarthy 2007a)

- \*ki: Assign one violation for each sequence of an unpalatalized voiceless consonant before [i].
- \*iCV: Assign one violation for high vowels in nonfinal open syllables.
- MAX: See 6.
- IDENT[back]: Assign one violation for every pair of input-output correspondents which do not have the same feature specification for [ $\pm$  back].

In the tableau in 19, the intended output candidate (1.) [ħa:k<sup>j</sup>min] satisfies both markedness constraints: satisfying \*ki by violating ID[bk], and satisfying \*iCV by violating MAX. However, this candidate is harmonically bounded by candidate (3.), which satisfies both markedness constraints through vowel deletion alone, violating only MAX.

- (19) HB Arabic counterbleeding in standard parallel OT

	/ħa:kim-in/	*ki	IDENT[bk]	*iCV	MAX
→ 1.	ħa:k <sup>j</sup> min		*		*
2.	ħa:kimin	*W	L	*W	L
3.	ħa:kmin		L		*W

Adding a contextual faithfulness constraint does not favor the intended winner, as shown in 21. For HB Arabic, the necessary constraint must prevent vowel deletion in the context of a preceding unpalatalized voiceless velar consonant in an open syllable: MAX(i)/k<sub>CV</sub>. This constraint is defined formally in 20.

- (20) MAX(i)/k<sub>CV</sub>: Let *A* be a high, front vowel in the context k<sub>CV</sub> in the input. *A* must have a correspondent in the output.  
i.e. Do not delete [i] when in the context k<sub>CV</sub> in the input.

Because the context for faithfulness is input-defined, and all candidates in the tableau share the same input, the contextual faithfulness constraint assigns violations to both the

intended winner (1.) and the candidate which only deletes (3., the candidate which is the problematic winner without the contextual faithfulness constraint). The intended winner remains harmonically bounded by candidate (3.), and in fact the addition of MAX (i)/k\_CV eliminates both candidates (1.) and (3.), making the faithful candidate (2.) the incorrect winner under this ranking. While contextual faithfulness provides a way of capturing counterfeeding interactions in parallel OT, the same approach cannot be used to capture counterbleeding interactions in parallel OT.

(21) HB Arabic counterbleeding in parallel OT with contextual faithfulness

	/ħa:kim-in/	MAX(i)/k_CV	*ki	IDENT[bk]	*iCV	MAX
→ 1.	ħa:k <sup>h</sup> min	*		*		*
● 2.	ħa:kimin		*W	L	*W	L
3.	ħa:kmin	*	L	L		*

#### 2.4 Interim discussion: Parallel OT analyses

In this section, we have shown that adding contextual faithfulness constraints to parallel OT analyses provides a general solution for underapplication opacity, but does not provide a solution for overapplication opacity. In underapplication opacity, the necessary solution must prevent an operation from applying in certain circumstances, even though its conditions are met. Contextual faithfulness constraints provide this solution by penalizing the application of an operation only when a specific condition holds in the input, and not otherwise. For overapplication opacity, the necessary solution must instead favor a candidate which violates more faithfulness constraints over a candidate which violates fewer faithfulness constraints to satisfy the same markedness constraints. Because of this formal difference between the two types of opacity, contextual faithfulness constraints cannot analyze overapplication in parallel OT. Adding a new contextual faithfulness constraint will not favor a candidate which already has more faithfulness violations than the winning candidate. However, as we show in §3, using these constraints in Harmonic Serialism will allow for analysis of both types of opacity.

We do not intend to account for underapplication resulting from fed counterfeeding rule orders in this paper. Fed counterfeeding (e.g. Kavitskaya & Staroverov 2010) is a specific case of counterfeeding where, for two ordered rules  $\mathbb{A}$  and  $\mathbb{B}$  where  $\mathbb{A}$  precedes  $\mathbb{B}$  in order of application,  $\mathbb{A}$  feeds  $\mathbb{B}$  and  $\mathbb{B}$  counterfeeds  $\mathbb{A}$ . The analytical challenge of fed counterfeeding in OT differs from other types of underapplication. The additional faithfulness we propose allows for the analyses in this section because the problematic candidates are the ones which undergo an additional change. That problematic change is prevented by the highly ranked contextual faithfulness constraint. In cases of fed counterfeeding, the problematic candidate is the faithful candidate. For this reason, contextual faithfulness constraints cannot be used to favor the intended winner. Kavitskaya & Staroverov (2010) have used Optimality Theory with Candidate Chains (McCarthy 2007a) to analyze fed counterfeeding. We compare our approach to OT-CC in §4.2.

### 3 Harmonic Serialism analysis

In this section, we show how contextual faithfulness constraints can account for both underapplication opacity and overapplication opacity when used in the serial constraint-based framework, Harmonic Serialism. This is accomplished with the addition of a dis-

inction between faithfulness to the input of the current step of the derivation, and faithfulness to the underlying representation.

In HS, GEN is restricted to generating candidates which differ from their input by only one change. The candidate selected as optimal in one step of the derivation is then used as the input for the next step of the derivation, and this continues until there is no further change that can be made to improve harmony (i.e., when the faithful candidate is selected as optimal). Faithfulness constraints in HS are typically evaluated relative to the input to the current step of the derivation. Assuming that the grammar always has access to the lexicon, HS additionally allows for the existence of faithfulness constraints that are evaluated relative to the underlying representation. We label these two types of faithfulness  $\text{FAITH}_{IO}$ , for faithfulness between the input and output of the current stage of the derivation, and  $\text{FAITH}_{UO}$ , for faithfulness between the underlying representation and the output of the current stage of the derivation. In this section, we show that both types of faithfulness are necessary to account for opacity in HS:  $\text{FAITH}_{IO}$  can be used to account for overapplication opacity, and  $\text{FAITH}_{UO}$  can be used to account for underapplication opacity.

### 3.1 Overapplication in HS: counterbleeding

In the example of overapplication opacity from HB Arabic (see §2.3), for input /ħa:kim-in/, [i] should delete from the open syllable, but only after palatalization has applied to [k]. In the parallel OT analysis, candidate (3.) [ħa:kmin] in the tableau in 21 is the incorrect winner where deletion applies without palatalization, because deletion on its own satisfies both markedness constraints \*iCV and \*ki. The contextual faithfulness constraint proposed in 20,  $\text{MAX}_{IO}(i)/k\_CV$  demands preservation of a high front vowel [i] in a specific input context: following an unpalatalized [k] (the context given in \*ki) and in an open syllable (the context given in \*iCV). This constraint does not suffice to analyze this pattern in parallel OT, because it does not provide a way to prefer the desired output [ħa:k<sup>j</sup>min] over the problematic candidate [ħa:kmin], since both delete in the specified context. When used in HS, this constraint does derive the rule ordering effect of palatalization preceding deletion, as illustrated in the derivation in 22, using the same constraints and ranking as in 21.

At step 1 in 22, the desired final output [ħa:k<sup>j</sup>min] is not available in HS, because it involves two changes from the input at this step. The grammar instead must choose between applying palatalization first (candidate 1.), applying deletion first (candidate 2.), or remaining faithful (candidate 3.). The contextual faithfulness constraint rules out the deletion candidate (2.), while the faithful candidate (3.) violates both markedness constraints \*iCV and \*ki. This leaves the palatalization candidate (1.) as the winner at step 1. At step 2, the input context specified in  $\text{MAX}_{IO}(i)/k\_CV$  no longer exists. Therefore, the constraint is not violated by the deletion of [i] in the winning candidate [ħa:k<sup>j</sup>min]. Instead, only the general MAX constraint is violated, which is ranked below \*iCV. The derivation converges on the intended final output [ħa:k<sup>j</sup>min] in step 3.

#### (22) Counterbleeding derivation path

/ħa:kim-in/ → ħa:k<sup>j</sup>imin → ħa:k<sup>j</sup>min → [ħa:k<sup>j</sup>min]

Rankings (Constraints defined in 18)

$\text{MAX}_{IO}(i)/k\_CV \gg *iCV, \text{IDENT}[\text{back}]$

\*ki  $\gg \text{IDENT}[\text{back}]$

\*iCV  $\gg \text{MAX}$

Step 1: Palatalization occurs

	/ħa:k <sup>j</sup> imin/	MAX <sub>IO</sub> (i)/k_CV	*ki	IDENT <sub>IO</sub> [bk]	*iCV	MAX <sub>IO</sub>
→ 1.	ħa:k <sup>j</sup> imin			*	*	
2.	ħa:kmin	*W		L	L	*W
3.	ħa:kimin		*W	L	*	

Step 2: Deletion occurs

	ħa:k <sup>j</sup> imin	MAX <sub>IO</sub> (i)/k_CV	*ki	IDENT <sub>IO</sub> [bk]	*iCV	MAX <sub>IO</sub>
→ 1.	ħa:k <sup>j</sup> min					*
2.	ħa:k <sup>j</sup> imin				*W	L

Step 3: Convergence<sup>6</sup>

	ħa:k <sup>j</sup> min	MAX <sub>IO</sub> (i)/k_CV	*ki	IDENT <sub>IO</sub> [bk]	*iCV	MAX <sub>IO</sub>
→ 1.	ħa:k <sup>j</sup> min					

This interaction is only possible when contextual faithfulness constraints are used in the HS framework. In parallel OT, it is not possible to refer to an intermediate context or step of the derivation because all changes are made at once. Providing a context for faithfulness in HS, however, effectively allows reference to an intermediate state of the derivation without requiring storage of intermediate forms. By specifying a context which ceases to exist after another process has applied, the constraint is essentially rendered inactive after a certain step of the derivation. Using this constraint in HS resolves the difficulty presented by the presence of the final intended winner, which undergoes two changes, in the parallel OT analysis.

The transparent processes in HB Arabic are given in 23. The relevant context is non-existent in the transparent deletion and palatalization cases, so the contextual faithfulness constraint does not affect these processes.

(23) Transparent deletion in HB Arabic

Step 1: /ʃaribat/ → [ʃarbat]

	/ʃaribat/	MAX <sub>IO</sub> (i)/k_CV	*ki	IDENT <sub>IO</sub> [bk]	*iCV	MAX <sub>IO</sub>
→ 1.	ʃarbat					*
2.	ʃaribat				*W	L

Transparent palatalization in HB Arabic

Step 1: /ħa:kim/ → [ħa:k<sup>j</sup>im]

	/ħa:kim/	MAX <sub>IO</sub> (i)/k_CV	*ki	IDENT <sub>IO</sub> [bk]	*iCV	MAX <sub>IO</sub>
→ 1.	ħa:k <sup>j</sup> im		*			
2.	ħa:kim		*W	L		

<sup>6</sup> We show the convergence step here to illustrate an example of convergence. In future derivations, when the convergence step is not crucial to the explanation, we will eliminate this step to conserve space. We also refrain from showing candidates which are harmonically bounded. For example, a candidate which removes palatalization at step 3 and competes with the winner of step 3 is harmonically bounded by the winning faithful candidate and therefore not shown.

### 3.1.1 Why overapplication is analyzeable in HS with contextual faithfulness

We have demonstrated an analysis of overapplication opacity using contextual faithfulness constraints. The addition of these constraints to an HS system allows us to capture certain rule-ordering effects which cannot be captured with contextual faithfulness constraints in parallel OT. In this section, we discuss why this analysis works and what features of HS make the overapplication analysis possible.

For many rule ordering effects, HS provides a way of ordering processes by constraint ranking. HS uses the principle of *gradualness*, which requires that the output of each stage of the derivation be more harmonic relative to the previous output (McCarthy 2010). Because of this, higher ranked constraints will be satisfied first. McCarthy (2008) uses this feature of HS to account for metrically conditioned syncope, using constraint ranking to provide an intrinsic ordering where stress assignment applies before syncope. When constraints assigning stress are ranked higher than the constraint demanding syncope, stress assignment occurs earlier in the derivation and will condition syncope. Similar approaches are used by Elfner (2016) for the analysis of metrically conditioned epenthesis, and Pruitt (2010) for metrically conditioned shortening.

These rule ordering effects can be understood as a type of feeding interaction, where one process conditions the application of another. The works cited above use HS's gradualness to analyze feeding interactions which are otherwise problematic with parallel evaluation. The constraint ranking in HS enforces a particular order of application for the phonological processes. The ranking of contextual faithfulness constraints also enforces a particular order of application for processes, though the appropriate definition of the relevant context is also crucial.

Overapplication opacity (and underapplication, though the focus in this section is on the overapplication analysis) cannot be analyzed in HS simply by ranking constraints according to a desired process ordering. Taking the HB Arabic case as an example, palatalization applies before vowel deletion. However, ranking the markedness constraint demanding palatalization above the markedness constraint demanding deletion does not suffice for the analysis. The tableau in 24 shows an attempted analysis in standard HS. At step 1, the desired winner is candidate (1.), the candidate which palatalizes [k] preceding [i]. At step 2, vowel deletion should apply. candidate (2.) at step 1 instead deletes [i] immediately, which simultaneously satisfies both relevant markedness constraints. In order to rule out this problematic candidate and force palatalization to 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 25. This is the same problem which occurs in parallel OT (see §2.3).

#### (24) HB Arabic counterbleeding in standard HS

Step 1: /ħa:kim-in/ → [ħa:k<sup>j</sup>imin]

	/ħa:kim-in/	*ki	IDENT[back]	*iCV	MAX
→ 1.	ħa:k <sup>j</sup> imin		*	*	
● 2.	ħa:kmin		L	L	*W
3.	ħa:kimin	*W	L	*	

#### (25) Transparent vowel deletion in standard HS

/ʃaribat/ → [ʃarbat]

	/ʃaribat/	*ki	IDENT[back]	*iCV	MAX
→ 1.	ʃarbat				*
2.	ʃaribat			*W	L

Rule ordering effects like counterbleeding opacity present a problem for HS and parallel OT because there is no intrinsic way to order satisfaction of markedness constraints when simultaneous satisfaction of both relevant markedness constraints is an option.<sup>7</sup> Ranking the markedness constraint which should be satisfied first over the one that should be satisfied later does not suffice to analyze overapplication opacity because there is one change which will satisfy both relevant markedness constraints simultaneously. This is the case in both parallel and serial evaluation systems.

Contextual faithfulness constraints effectively order application of processes by preventing the one change that would satisfy both relevant markedness constraints simultaneously. They do this by providing extra faithfulness in a particular context, essentially blocking a process from applying until another process has applied. In HS, the output at one stage of the derivation becomes the input at the next stage of the derivation, which provides the crucial mechanism allowing for the analysis of overapplication.

In the HB Arabic case,  $MAX_{io}(i)/k_{CV}$  prevents vowel deletion in the context  $[k_{CV}]$ . Ranking this constraint above the relevant markedness constraints prevents deletion (the process which would satisfy both markedness constraints simultaneously) from happening at step 1. With that constraint preventing deletion, palatalization can instead apply to satisfy \*ki at step 1. The output of step 1 then becomes the input to step 2. This input now has a palatalized  $[k^j]$  preceding the vowel, so the context specified by  $MAX_{io}(i)/k_{CV}$  no longer exists. When the context in a contextual faithfulness constraint is not present in the input, the constraint is vacuously satisfied. At step 2, vowel deletion can occur to satisfy \*iCV as \*iCV is ranked above regular MAX.

This analysis is possible because HS uses an intermediate form as the input to next stage of the derivation. This allows contextual constraints to be rendered inactive<sup>8</sup> after a particular process applies and the specified context no longer exists. In the case of overapplication, we prevent the problematic change from happening at step 1, but allow that same process to happen at step 2 by defining a context which is changed by application of palatalization.

Contextual faithfulness constraints allow for analysis of overapplication opacity in HS but cannot analyze all observed rule ordering effects. In order to analyze underapplication effects, we must add a distinction between faithfulness to the original underlying representation and faithfulness to the input of the current stage of the derivation. This is explored in the following section.

### 3.2 Underapplication analysis in HS: Counterfeeding

In the HS literature, the standard way to evaluate faithfulness is between the output and the input of the current stage of the derivation. Here, we propose constraints which evaluate faithfulness between the output and the underlying representation. Faithfulness to the underlying representation has been used before in HS (McCarthy 2007b), but not as a method of analyzing opacity.<sup>9</sup> As we show in §3.1, *input-output* contextual faithfulness constraints ( $FAITH_{io}$ ) can be used to analyze overapplication opacity in HS, but *underlying-output* contextual faithfulness constraints ( $FAITH_{uo}$ ) are needed to analyze underapplica-

<sup>7</sup> While there is no built-in mechanism for ordering satisfaction of markedness constraints, extensions to basic HS have done this (McCarthy 2007a; Jarosz 2014). We provide a comparison with these analyses in §4.2.

<sup>8</sup> Contextual faithfulness constraints could also become active at some point in an HS derivation if the specified context is created by application of another process. This is not involved in the analysis of opacity so we refrain from exploring the consequences of that feature of contextual faithfulness in this paper.

<sup>9</sup> Referencing the UR is a part of OT with Candidate Chains (McCarthy 2007a) which is used to analyze opacity. However, referencing the UR is only a small part of that analysis which also references intermediate forms. Our analysis references the UR without needing to also store a continued reference to intermediate forms. McCarthy (2007a) also explores the general challenges that opacity poses for standard HS, specifically the use of  $FAITH_{io}$  constraints. For more detail on OT-CC and further comparison with our analysis, see §4.2.

tion opacity. Having a system with both types of faithfulness is only possible in a serial evaluation system like HS where the IO/UO distinction can be drawn.

As in parallel OT, a standard HS analysis of the HB Arabic counterfeeding case (see §2.2 for explanation of the data and subsequent parallel OT analysis using contextual faithfulness) fails due to a ranking contradiction, shown in examples 26–27.<sup>10</sup> We use the term standard as this analysis uses the more common IO faithfulness constraints and there is no addition of any contextual faithfulness constraints. The ranking  $\text{MAX}, *aCV \gg \text{ID}[\text{low}]$ ,  $*iCV$  could motivate raising the underlying mid vowel to high in step 1 of example 26. However, this ranking cannot be used because  $*iCV$  must dominate  $\text{MAX}$  in the transparent case shown in 27.

(26) HB Arabic counterfeeding in standard HS

Step 1: /dafaʕ/ → [difaʕ]

	/dafaʕ/	*aCV	IDENT[low]	*iCV	MAX
→ 1.	difaʕ		*	*	
2.	dafaʕ	*W	L	L	
3.	dfaʕ		L	L	*W

(27) Transparent vowel deletion in standard HS

Ranking needed:  $*iCV \gg \text{MAX}$

/ʃaribat/ → [ʃarbat]

	/ʃaribat/	*aCV	IDENT[low]	*iCV	MAX
→ 1.	ʃarbat				*
2.	ʃaribat			*W	L

The use of contextual  $\text{FAITH}_{\text{UO}}$  constraints resolves this problem by demanding faithfulness between the initial input (underlying representation) and the output of the current stage of the derivation. Formal definitions of these constraints are given in 28. These constraints are (vacuously) satisfied if the specified context, segment, or feature does not exist in the UR.

(28)  $\text{IDENT}_{\text{UO}}(\text{F})/[\alpha\overline{\text{G}}]$

Let  $A$  be a segment specified for some feature  $[\text{G}]$  in the UR. Assign one violation if the output correspondent of  $A$  does not have the same specification for  $[\text{F}]$  as  $A$ .

i.e. Do not change the value of  $[\text{F}]$  for segments that are  $[\alpha\text{G}]$  in the UR.  
(counterfeeding on focus/chain shifts)

$\text{IDENT}_{\text{UO}}(\text{F})/_{[\alpha\text{G}]}$

Let  $A$  be a segment in some context  $_{[\alpha\text{G}]}$  in the UR. Assign one violation if the output correspondent of  $A$  does not have the same specification for  $[\text{F}]$  as  $A$ .

i.e. Do not change the value of  $[\text{F}]$  for segments that are in the context  $_{[\alpha\text{G}]}$  in the UR.

(counterfeeding on environment)

<sup>10</sup> Here, we assume that one feature change qualifies as *one change* between input and output. It is sometimes claimed in the HS literature that deleting an entire segment cannot be considered one change. Instead, deletion occurs in two steps: reduction/removal of place, then deletion of the remaining segmental skeleton (McCarthy 2008; 2018). The question of whether deletion is one change or two will not be addressed by this paper, as our analysis works under either model of GEN.

While this does require a representation of both the UR and the current input to evaluate the faithfulness constraints, there are multiple reasons why this system is more economical relative to other analyses of opacity in HS: (1) it does not require keeping track of all intermediate forms, as in OT-CC (McCarthy 2007a); (2) if we assume the UR to be the lexical representation, speakers already have a representation of this form stored in the lexicon. Therefore, requiring a representation of the UR does not require representation of anything beyond what is already present in the lexicon.

The contextual faithfulness analyses used in parallel OT for underapplication can be adapted for HS by making the relevant contextual faithfulness constraint a  $FAITH_{UO}$  constraint instead of a  $FAITH_{IO}$  constraint. For HB Arabic counterfeeding, the constraint which prevents derived high vowels from deleting becomes  $MAX_{UO}/[+\overline{low}]$ , defined in 29. This constraint demands that segments in the output with [+low] input correspondents be realized and not deleted. For input /dafaʕ/, the low vowel becomes high at step 1 and the problematic candidate (3.) which deletes /a/ is ruled out by the highly ranked  $FAITH_{UO}$  constraint. The derivation will then converge on step 2.

- (29)  $MAX_{UO}/[+\overline{low}]$ : Let *A* be a segment specified [+low] in the UR. Assign one violation if *A* does not have an output correspondent.  
i.e. Do not delete segments that are [+low] in the UR.

- (30) HB Arabic counterfeeding (Al-Mozainy 1981)  
Derivation path: /dafaʕ/ → difaʕ ↗ \*dfaʕ  
Rankings  
\*aCV >> \*iCV, IDENT[low]  
 $MAX_{UO}/[+\overline{low}]$  >> \*iCV, IDENT[low]

HB Arabic counterfeeding in HS with the contextual  $FAITH_{UO}$  constraint

Step 1: /dafaʕ/ → [difaʕ]

	/dafaʕ/	$MAX_{UO}/[+\overline{low}]$	*aCV	IDENT[low]	*iCV	MAX
→ 1.	difaʕ			*	*	
2.	dafaʕ		*W	L	L	
3.	dfaʕ	*W		L	L	*W

Step 2: Convergence difaʕ ↗ \*dfaʕ

UR:	/dafaʕ/	$MAX_{UO}/[+\overline{low}]$	*aCV	IDENT[low]	*iCV	MAX
→ 1.	difaʕ				*	
2.	dfaʕ	*W			L	*W

The tableau in 31 shows the transparent case in HB Arabic, where [i] deletes in non-final open syllables. This is permitted by our constraint set because the vowel in the underlying representation is a high vowel, so the  $FAITH_{UO}$  constraint is not violated. The candidate with deletion is the optimal candidate, because the markedness constraint against high vowels in open syllables outranks the general faithfulness constraint.



(31) /*ʃaribat*/ → [*ʃarbat*]

Ranking

\*iCV &gt;&gt; MAX

Step 1: /*ʃaribat*/ → *ʃarbat*

	/ʃaribat/	MAX <sub>UO</sub> /[+low]	*aCV	IDENT[low]	*iCV	MAX
→ 1.	<i>ʃarbat</i>					*
2.	<i>ʃaribat</i>				*W	L

The parallel OT analyses we provided in §2 for counterfeeding on environment can also be analyzed with this approach. The same contextual faithfulness constraint we proposed for the parallel OT analysis is required in HS, except that it must be a FAITH<sub>UO</sub> constraint rather than a FAITH<sub>IO</sub> constraint.

The HS analysis of Lomongo uses the same data (12) and constraint set (14) as in §2.2.2 with one crucial difference: the contextual faithfulness constraint is a FAITH<sub>UO</sub> constraint, given in 32. This constraint now works in the HS analysis as it does in the parallel OT analysis—it prevents glide formation for vowel which originally preceded obstruents in the UR. The extra faithfulness allows for underapplication of gliding in the surface form.

(32) ID[voc]<sub>UO</sub>/\_VCEOBS: Let *A* be a segment in the context preceding a [+voi,-son] segment in the UR. Assign one violation if the output correspondent of *A* does not have the same specification for [voc] as *A*.

In step 1 of the transparent derivation (shown in 33), input /*o-isa*/ becomes [*wisa*] because the markedness constraint penalizing hiatus outranks the faithfulness constraint for [vocalic]. The FAITH<sub>UO</sub> constraint is not active in this derivation, and does not interfere with the transparent interaction. This is because the constraint can only assign violations when the designated context is present in the UR. The derivation will then converge on step 2.

In the opaque interaction, /*b*/ deletes at step 1 because the markedness constraint penalizing intervocalic sonorants outranks MAX (shown in 34). The contextual UO constraint does not assign any violations at this step because no segments change value of the feature [vocalic]. [*oisa*] then becomes the input to step 2 (in 35). At step 2, there is now a possibility for glide formation to occur. This is prevented by the high ranking contextual UO constraint, because the segment whose value of [vocalic] has changed preceded a voiced obstruent in the UR.

(33) Lomongo transparent interaction in HS

Step 1: /*o-isa*/ → [*w-isa*]

UR:	/o-isa/ o-isa	ID[voc] <sub>UO</sub> /_VCEOBS	*VCEOBS/V_V	MAX	*HIATUS	ID[voc]
→ 1.	<i>wisa</i>					*
2.	<i>oisa</i>				*W	L

- (34) Lomongo counterfeeding interaction in HS  
 Step 1: /o-bina/ → o-ina

UR: /o-bina/ o-bina	ID[voc] <sub>UO/_VCEOBS</sub>	*VCEOBS/V_V	MAX	*HIATUS	ID[voc]
→ 1. oina			*	*	
2. obina		*W	L	L	

- (35) Lomongo counterfeeding interaction in HS  
 Step 2: /o-bina/ → o-ina

UR: /o-bina/ oina	ID[voc] <sub>UO/_VCEOBS</sub>	*VCEOBS/V_V	MAX	*HIATUS	ID[voc]
→ 1. oina				*	
2. wina	*W			L	*W

### 3.3 Interim discussion: Underapplication in HS

Other analyses of counterfeeding and underapplication can be adapted to HS similarly with the use of contextual FAITH<sub>UO</sub> constraints. The crucial property which allows for the analysis of underapplication opacity is the reference to the original input, or UR. Faithfulness to a particular environment in the UR (when ranked highly) will prevent certain processes from applying even when the relevant markedness constraints remain active, resulting in underapplication on the surface.

Because these constraints reference elements in the underlying representation, it might seem that *richness of the base* (ROTB; Prince & Smolensky 1993/2004) would pose problems for the analysis. ROTB is an axiom generally held in OT work and states that the set of possible inputs to the grammar is universal (Smolensky 1996). In other words, possible inputs/URs are not restricted on a per language basis and any input should produce a phonologically valid output. This is still true for our analysis even though our proposed faithfulness constraints reference specific features in the UR. Just as in standard OT, contrast emerges through crucial properties of inputs. Opaque phenomena are themselves defined with reference to the underlying representation. If the UR does not contain the specified element in the FAITH<sub>UO</sub> constraint, the interaction will be transparent, and we would not want to predict opacity.

For example, in the HB Arabic counterfeeding case, our analysis will only produce the opaque interaction with a low vowel in the input because the FAITH<sub>UO</sub> constraint references the feature [+low] in the UR. The opaque input /dafaʕ/ becomes [difaʕ] but does not continue to delete as in \*[dfaʕ] even though input /ʃaribat/ does delete to [ʃarbat]. If instead the input were to contain a non-low vowel, the constraint MAX<sub>UO</sub>/[+low] would not prevent deletion, and the interaction would be transparent /difaʕ/ → [dfaʕ]. We argue that this is not problematic because inputs without the crucial features in the UR would still emerge as phonologically valid in the language, but the output would be considered a different word, as with any case of contrast in OT. By using these constraints in a system with ROTB, we do predict that opaque interactions hinge on specific properties of URs. However, we do not consider this to be problematic due to the nature of opacity and the way phonological contrast emerges in OT.

## 4 Discussion

In this paper, we have examined the analytical potential of contextual faithfulness constraints, demonstrating their utility in providing a general solution for underapplication opacity in parallel Optimality Theory, and for both underapplication and overapplication opacity in Harmonic Serialism with the addition of a distinction between faithfulness to the input of the current step of the derivation ( $\text{FAITH}_{IO}$ ) and faithfulness to the underlying representation ( $\text{FAITH}_{UO}$ ). Counterbleeding and counterfeeding interactions can be analyzed in HS using the generalized ranking in 36. This type of ranking has previously been used in positional faithfulness analyses (Beckman 1997; Lombardi 1999).

- (36) A generalized ranking for analyzing opacity in HS  
 CONTEXTUALFAITH  $\gg$  MARKEDNESS  $\gg$  GENERALFAITH

In cases of opacity, there are two (or more) relevant processes. Each of these processes is transparent for some input, so some markedness constraint must outrank some faithfulness constraint. In order for these transparent interactions to happen, we need a MARKEDNESS  $\gg$  GENERALFAITH ranking. Our proposal is the addition of the higher ranked contextual faithfulness constraints which account for opacity while maintaining the MARKEDNESS  $\gg$  GENERALFAITH ranking needed for the transparent interactions.

In this section, we examine typological consequences of this analysis (§4.1) and compare our analysis to other Optimality Theoretic approaches to opacity (§4.2).

### 4.1 Contextual faithfulness and the nature of Con

One of the main objections to using contextually defined faithfulness constraints to analyze opacity comes from McCarthy (2007a), who argues that such constraints would create an overly rich faithfulness theory. We argue that fears of an overly rich CON should not be cause to immediately discount particular constraint types, and we demonstrate two potential ways of approaching this problem.

The default option is allowing the full set of possible contextual faithfulness constraints to exist in the universal CON. While this would create more faithfulness constraints than necessary, an overly rich faithfulness theory does not necessarily cause pathological typological predictions. We show the hypothetical consequences of adding these constraints to a factorial typology in §4.1.1.

The second option, which we see as a promising avenue for future work in HS, is that these constraints are induced, and therefore not included in the universal CON. Because they are so specific to each interaction, contextual faithfulness constraints could be induced on a language-specific basis in response to opaque data. In section §4.1.2 we present a sketch of a potential induction algorithm, point out challenges, and suggest areas for future work.

#### 4.1.1 Factorial typology with contextual faithfulness

In this section, we show what would happen if these constraints were included in the factorial typology of the universal CON. We calculated two example factorial typologies within HS using OT-Help (Staub et al. 2010), one using the constraint set from the HB Arabic counterfeeding analysis (see §3.2) which includes a  $\text{FAITH}_{UO}$  constraint, and one using the constraint set from the HB Arabic counterbleeding analysis (see §3.1) which includes a  $\text{FAITH}_{IO}$  constraint.

We show the two typology calculations separately so it is clear which predicted languages arise from the use of the  $\text{FAITH}_{UO}$  constraint versus the  $\text{FAITH}_{IO}$  constraint. The

FAITH<sub>UO</sub> constraint only adds the desired underapplication pattern. The FAITH<sub>IO</sub> constraint adds the desired overapplication pattern, as well as an additional underapplication pattern which bears formal similarities to *do something except when* blocking (Baković 2011a).

Table 1 shows the factorial typology calculated using the data and constraints from the HB Arabic counterfeeding example of underapplication opacity (see 6 and 29). There are five languages predicted in this typology; for each, we give the optimal outputs for inputs /dafaʕ/ and /ʃaribat/ (which demonstrate whether the raising and deletion processes are applied), the ranking which yields that pattern, and a descriptive label. Languages 1–4 are all present in the typology without the addition of the contextual faithfulness constraint. The only language added to the typology with our proposed constraint is Language 5, the counterfeeding pattern.

Although McCarthy (2007a) was concerned about the use of contextually defined faithfulness constraints for counterfeeding resulting in an overly rich faithfulness theory, the inclusion of FAITH<sub>UO</sub> contextual faithfulness constraints for opacity does not necessarily cause unwanted typological predictions. In this example, the only pattern added to the typology is the counterfeeding pattern.

The table in 2 shows the factorial typology calculated using the data and constraints from the HB Arabic example of overapplication opacity (see 18 and 20). There are eight language types produced in this typology; for each, we give the optimal outputs for inputs /ʃaribat/, /ħa:kim/, and /ħa:kimin/ (which demonstrate whether the palatalization and vowel deletion processes are applied), the ranking which yields that pattern, and a descriptive label. Languages 1–6 are all present in the typology without the addition of the contextual faithfulness constraint.

The languages added to the typology with our proposed constraint are Languages 7–8. Language 7 is counterbleeding, the attested pattern in HB Arabic. Language 8 is an instance of underapplication in which vowel deletion repairs violations of \*ki and \*iCV, unless in the context of k<sub>CV</sub>, where deletion is blocked.

The underapplication pattern in Language 8 could formally be considered *do something except when* (DSEW) blocking because deletion applies except in a particular context (Prince & Smolensky 1993/2004; Baković 2011b). In Language 8, deletion applies to repair \*iCV and \*ki except in the context of k<sub>CV</sub>. This meets the formal criterion for DSEW blocking. However, this language differs from actual (observed) cases of DSEW blocking, which are typically motivated by a general phonotactic constraint in the language (Prince & Smolensky 1993/2004; Baković 2011b). This indicates that actual DSEW blocking is

**Table 1:** HS typology with HB Arabic using the contextual UO constraint.

	/ʃaribat/	/dafaʕ/	Ranking	Description
1	[ʃaribat]	[dafaʕ]	MAX, ID[low], MAX <sub>UO</sub> /[+low] » *aCV, *iCV	faithful
2	[ʃaribat]	[difaʕ]	*aCV, MAX, MAX <sub>UO</sub> /[+low] » *iCV, ID[low]	[a] raises to [i]
3	[ʃarbat]	[dafaʕ]	*iCV, ID[low], MAX <sub>UO</sub> /[+low] » *aCV, MAX	[i] deletes
4	[ʃarbat]	[dfaʕ]	*aCV » *iCV, ID[low] » MAX, MAX <sub>UO</sub> /[+low]	raising feeds deletion
5	[ʃarbat]	[difaʕ]	<b>*aCV, MAX<sub>UO</sub>/[+low]</b> » <b>*iCV, ID[low]</b> » <b>MAX</b>	<b>counterfeeding</b>

**Table 2:** HS typology with HB Arabic using the contextual IO constraint.

	/ʃaribat/	/ħa:kim/	/ħa:kimin/	Ranking	Description
1	[ʃaribat]	[ħa:kim]	[ħa:kimin]	MAX, ID[BK], MAX <sub>io</sub> (i)/k_CV » *ki, *iCV	faithful
2	[ʃarbat]	[ħa:k <sup>ɨ</sup> im]	[ħa:kmin]	*ki, *iCV » MAX, MAX <sub>io</sub> (i)/k_CV » ID[BK]	deletion bleeds palatalization
3	[ʃaribat]	[ħa:k <sup>ɨ</sup> im]	[ħa:k <sup>ɨ</sup> imin]	*ki, MAX, MAX <sub>io</sub> (i)/k_CV » *iCV, ID[BK]	palatalization
4	[ʃarbat]	[ħa:kim]	[ħa:kmin]	*iCV, ID[BK] » MAX, MAX <sub>io</sub> (i)/k_CV » *ki	delete to repair *iCV
5	[ʃaribat]	[ħa:km]	[ħa:kmin]	*ki, ID[BK] » MAX, MAX <sub>io</sub> (i)/k_CV » *iCV	delete to repair *ki <sup>11</sup>
6	[ʃarbat]	[ħa:km]	[ħa:kmin]	*ki, *iCV, ID[BK] » MAX, MAX <sub>io</sub> (i)/k_CV	deletion
7	[ʃarbat]	[ħa:k <sup>ɨ</sup> im]	[ħa:k <sup>ɨ</sup> min]	<b>*ki, MAX<sub>io</sub>(i)/k_CV</b> <b>» *iCV, ID[bk]</b> <b>» MAX</b>	<b>counterbleeding</b>
8	[ʃarbat]	[ħa:km]	[ħa:kimin]	<b>ID[bk], MAX<sub>io</sub>(i)/k_CV</b> <b>» *ki, *iCV</b> <b>» MAX</b>	<b>blocking</b>

likely motivated by a highly ranked markedness constraint, not a highly ranked contextual faithfulness constraint.

For example, Baković (2011b: 12) cites Kisseberth (1970) as the earliest argument for DSEW blocking, using data from Yawelmani Yokuts (Newman 1944; Kuroda 1967; Kisseberth 1969): short vowels delete between consonants except when that deletion would result in a tautosyllabic consonant cluster. In this case, the blocking is phonotactically motivated (avoidance of a particular marked structure in the language). The blocking in our Language 8 is unlike this observed case of DSEW blocking in that the Language 8 blocking appears to be phonotactically unmotivated. Deleting [i] in the context k\_CV does not result in a structure which is phonotactically illegal in the language. Although the pattern in Language 8 is not similar in motivation, it is formally similar to observed cases of DSEW blocking in that deletion applies except in a particular context.

#### 4.1.2 Learning and constraint induction

The constraints we propose are highly specific and include a lot of information: a specified focus, a specified context, and reference to either the most recent input or the UR. Another approach could be that these constraints are induced by learners on a language

<sup>11</sup> While this pattern may seem to be a case of the too-many-solutions problem, it is attested. Using data from Kirundi, Kochetov (2016) provides evidence of deletion as a repair to consonant-glide sequences, which is another context that might trigger palatalization. Thus, deletion to repair \*ki may be attested, but it does seem to be rare.

specific basis when presented with opaque data. In this case, no instances of the phonologically unmotivated blocking pattern (examined in the previous section) would arise as the contextual faithfulness constraints would not enter the factorial typology.

Constraint induction has been used in other domains for language specific constraints, such as morphologically indexed constraints, where lexically indexed constraints are induced to resolve ranking inconsistencies (Pater 2010). In this section, we discuss some relevant background on learning in HS, and show how the contextual faithfulness constraints needed for a given case of opacity can be derived by composing preexisting constraints. This section is not intended to be a detailed proposal for a learning algorithm, but rather a discussion of key ideas that could support the development of such an algorithm in future work.

The key components of a learning algorithm which can induce contextual faithfulness constraints would be (1) the capacity to re-rank constraints as needed, (2) the ability to detect the need for inducing a new constraint, and (3) the ability to compose the new contextual faithfulness constraint. One major challenge to actually implementing such a learning algorithm is that there has been relatively little work on learning in HS.

One of the most commonly assumed algorithms for learning a constraint ranking in parallel OT is Recursive Constraint Demotion (RCD; Tesar & Smolensky 1998). Extending RCD to HS is not straightforward, however, because HS allows for multi-step derivations from the UR to the surface form. Learning constraint rankings motivated only at an intermediate step of a derivation presents a hidden structure problem for which there is no unified solution (for various approaches see Staubs & Pater 2012; Tessier 2013; Tessier & Jesney 2014; Jarosz 2016).

Despite these difficulties, one major benefit of using RCD is that it provides a mechanism for detecting ranking inconsistencies, which is a crucial piece needed in detecting when it is necessary to induce a new constraint. Jarosz (2014) demonstrates that cases of potential feeding/bleeding display characteristic properties in HS, leading to characteristic types of ranking inconsistencies for opaque outputs. These properties could be used by a learning algorithm to diagnose opacity and inform the construction of a new contextual faithfulness constraint. RCD will rank constraints successfully for cases of potential bleeding/feeding when they result in transparent interactions. RCD will fail for these cases when they result in opaque interactions. The crucial properties of potential bleeding and potential feeding are formally different, so this information can be used to inform the learner about whether to induce a  $\text{FAITH}_{\text{UO}}$  constraint or a  $\text{FAITH}_{\text{IO}}$  constraint when RCD fails. We illustrate these characteristic properties using two of the examples analyzed here.

In the HB Arabic counterbleeding example, the two relevant processes are palatalization and deletion. Jarosz (2014: 3) uses the same HB Arabic example to illustrate the characteristic ranking inconsistency posed by counterbleeding in HS, which we summarize here. The ranking requirement for deletion can be modeled as  $M_A \gg F_A$  (\*iCV  $\gg$  MAX). The ranking requirement for palatalization can be modeled as  $M_B \gg F_B$  (\*ki  $\gg$  IDENT[back]). When implemented in HS, application of process A (deletion) simultaneously causes satisfaction of the markedness constraint involved in process B (palatalization),  $M_B$ . Thus, “the essential characteristic of a potential bleeding interaction is that satisfaction of  $M_A$  (\*iCV) results in the satisfaction of both  $M_A$  (\*iCV) and  $M_B$  (\*ki)” (Jarosz 2014: 3).

This essential characteristic can be used to identify cases of counterbleeding and trigger induction of a contextual faithfulness constraint. We illustrate this ranking problem in 24, reproduced here as 37. The desired winner is candidate (1.), the candidate which palatalizes at the first step of the derivation (indicated with the arrow). The incorrect winner based on the current constraint ranking is candidate (2.), the candidate which deletes at

the first step (indicated with the bomb symbol). The faithful candidate is candidate (3.). The crucial characteristic of potential bleeding is that satisfaction of  $M_A$  (done by candidate 2.) results in simultaneous satisfaction of both  $M_A$  and  $M_B$ . The ranking inconsistency is caused by the fact that the desired candidate only satisfies  $M_B$ .

(37) HB Arabic counterbleeding in standard HS

Step 1: /ħa:kim-in/ → ħa:k<sup>i</sup>jmin

	/ħa:kim-in/	*ki	IDENT[back]	*iCV	MAX
→ 1.	ħa:k <sup>i</sup> imin		*	*	
☛ 2.	ħa:kmin		L	L	*W
3.	ħa:kimin	*W	L	*	

These properties of the candidates would be present when RCD fails. We assume here (along with previous work, summarized above) that the learner would have access to inputs, outputs, candidates and violation profiles. Given the violation profiles of the three candidates we provide in 37, the learner can identify which constraints are  $M_A$  and  $M_B$ —these are the two markedness constraints violated by the faithful candidate (3.) but satisfied by the problematic candidate (2.). Of these two constraints,  $M_A$  is the constraint which is satisfied by the problematic candidate but violated by the intended winner, candidate (1.). When presented with this particular characteristic ranking inconsistency, the learner would be triggered to induce an IO contextual faithfulness constraint.

The ranking problem of counterfeeding is different. We again summarize Jarosz (2014: 3) on characteristic ranking inconsistencies in HS, applying this analysis to our example of counterfeeding in Lomongo. There are two relevant processes: A (intervocalic deletion) and B (glide formation). Unlike the counterbleeding case, standard HS can model the first step of the derivation where process A (deletion) applies with the ranking  $M_A \gg F_A$  (\*VCDOBS/V\_V  $\gg$  MAX).<sup>12</sup> The next step of the derivation presents the ranking inconsistency: there is no way to prevent process B from applying (which would require the ranking  $F_B \gg M_B$ ) as the ranking  $M_B \gg F_B$  is independently needed to account for transparent instances of process B. The characteristic problem of counterfeeding is that application of process A creates violations of  $M_B$ , which are subsequently not resolved even though process B is attested elsewhere in the language.

This characteristic inconsistency can be used to identify cases of counterfeeding and trigger induction of a UO contextual faithfulness constraint. The ranking problem presented by the Lomongo counterfeeding example is shown in 38. Once the ranking inconsistency has been detected, the learner can use properties of the candidates and their violation profiles to identify this as a counterfeeding-related inconsistency. In step 2 of the derivation, where RCD would fail, we show the three candidates the learner would need to correctly identify the inconsistency and involved constraints (whose properties are necessary for induction).

<sup>12</sup> Standard HS can model step 1 except in cases where deletion underapplies and HS's GEN models full deletion as one change, as in 30. In these cases, there will already be a ranking inconsistency at step 1. This does not necessarily pose a problem for the steps outlined here—the step 1 inconsistency would trigger induction of a redundant IO faithfulness constraint in these cases. If deletion instead requires multiple steps, full deletion would not be available at step 1, and induction can proceed as in the Lomongo example shown here. This showcases one of the general challenges of learning in HS—finer details of constraint induction will be determined by the capabilities of GEN (and there is no unified solution on what counts as one change in HS).

(38) Lomongo counterfeeding interaction in standard HS

Step 1: /o-bina/ → oina

/obina/	*VCD OBS/V_V	MAX	*HIATUS	ID[voc]
→ 1. oina		*	*	
2. obina	*W	L	L	

Step 2: oina → [oina]

/oina/	*VCD OBS/V_V	MAX	*HIATUS	ID[voc]
→ 1. oina			*	
● 2. wina			L	*W
3. obina	*W		L	

Candidate (1.) is the intended winner, which is also the faithful candidate, indicated with the arrow. Candidate (2.) is the problematic candidate which applies process B and incorrectly wins under the current ranking (indicated with the bomb symbol). Candidate (3.) is the candidate which removes the application of process A (the original UR). The crucial property of counterfeeding is that resolving  $M_A$  creates violations of  $M_B$ , and those violations must be left unresolved. The fact that candidate (1.) is the intended winner but candidate (2.) wins under the current ranking shows that the inconsistency is caused by the fact that  $M_B$  must be left unresolved. When this inconsistency is present, and the learner has independent evidence of  $M_B \gg F_B$  (from the transparent case), the learner would be triggered to induce a UO contextual faithfulness constraint.

Once the inconsistencies have been identified, the induction of a new contextual faithfulness constraint can be accomplished by combining properties of the existing involved constraints. In cases of counterbleeding opacity, the IO contextual faithfulness constraint can be constructed by combining properties of  $F_A$  with properties of  $M_A$  and  $M_B$  (these constraints would have been identified during the inconsistency detection).  $F_A$  provides the type of faithfulness for the constructed contextual faithfulness constraint.  $M_A$  and  $M_B$  provide the focus and context of faithfulness. The contextual overlap between the two markedness constraints becomes the focus, and the surrounding material becomes the context.

As an example, we show how induction would proceed for the HB Arabic counterbleeding interaction (see §3.1 for the analysis). In this example, deletion of [i] in open syllables should follow palatalization of [k] before [i]. The detection of a ranking inconsistency associated with counterbleeding would trigger the learner to induce a contextual  $FAITH_{IO}$  constraint and allow the learner to identify the involved constraints ( $M_A$  and  $M_B$ ,  $F_A$  and  $F_B$ ). In this case, MAX is  $F_A$ , so the contextual faithfulness constraint will be a  $MAX_{IO}$  constraint.

The two markedness constraints which must ultimately be satisfied are  $*iCV(M_A)$ , which is satisfied by applying deletion, and  $*ki(M_B)$ , which is satisfied by applying palatalization. The markedness constraints have a contextual overlap with the segment [i], so the focus of the new contextual faithfulness constraint will be [i]:  $MAX_{IO}(i)$ . The remaining material from the markedness constraints (k\_ from  $*ki$  and \_CV from  $*iCV$ ) is combined to form the context, yielding the final constructed contextual faithfulness constraint:  $MAX(i)/k\_CV$ . This process is schematized in 39.



## (39) Building a contextual faithfulness constraint for counterbleeding

Step 1. Identify  $F_A$ :

- MAX

Step 2. Find the contextual overlap between  $M_A$  and  $M_B$ :<sup>13</sup>

$$\begin{array}{c} * \quad k \mid i \mid \\ * \quad \quad \mid i \mid C \quad V \end{array}$$

Step 3. The contextual overlap becomes the focus of the contextual faithfulness constraint:

- MAX(i)

The material surrounding the overlap becomes the context for the faithfulness constraint:

- MAX(i)/k\_CV

For cases of counterfeeding opacity, the learner would need to induce a FAITH<sub>UO</sub> constraint, also by combining properties of the existing faithfulness and markedness constraints involved. In this case,  $F_B$  provides the type of faithfulness constraint, and the focus of  $M_A$  provides the context of faithfulness. The learner would again be able to identify the relevant constraints during inconsistency detection by comparing the violation profiles of the intended winner, the problematic candidate, and the faithful candidate (as summarized above).

## (40) Building a contextual faithfulness constraint for counterfeeding

Step 1. Identify  $F_B$ :

- IDENT[vocalic]

Step 2. Make this faithfulness constraint demand faithfulness between the UR and the current step of the derivation instead of the input and output of the current step:

- IDENT[vocalic]<sub>UO</sub>

Step 3. The focus of  $M_A$  becomes the context for the faithfulness constraint:

- IDENT<sub>UO</sub>(vocalic)/\_VCEOBS

In the case of the Lomongo counterfeeding interaction (see 38),  $F_B$  is ID[vocalic], so the type of the constructed contextual faithfulness constraint will be ID[voc]<sub>UO</sub>.  $M_A$  is \*VCDOBS/V\_V, the markedness constraint satisfied by the application of intervocalic voiced obstruent deletion. The focus of this constraint is VCDOBS (= [+voi, -son], see 14 for constraint definition), which becomes the context of new contextual faithfulness constraint. This yields the final constructed contextual faithfulness constraint: IDENT<sub>UO</sub>(vocalic)/\_[VCEOBS]. This process is schematized in 40.

In this section, we have discussed some key pieces needed to implement a learning algorithm for inducing contextual faithfulness constraints in cases of opacity: (1) and RCD-like

<sup>13</sup> The constraints need not be formatted in any particular way to align the contexts and find the overlap, provided we assume segments are feature bundles. Under this assumption, we can align constraints that are formulated with features or segments by using the featural composition of segments for alignment. We show an example with segments here for ease of explanation, and also because these are the constraints which are commonly used for these cases in previous literature.

algorithm for ranking constraints which can detect ranking inconsistencies implemented in HS, (2) a definition of the ranking inconsistencies which are characteristic of counterbleeding and counterfeeding opacity, and (3) a means of constructing the needed contextual faithfulness constraints by composing elements of independently-needed constraints. The main goal of the current paper is to showcase the analytical potential of contextual faithfulness constraints for analyzing opacity by providing analyses of several opaque interactions, and we leave the computational implementation of a learning algorithm for future work.

## 4.2 Comparison with other analyses

The existing literature contains many analyses of opacity in OT. In this section, we discuss how our analysis compares with OT with Candidate Chains (McCarthy 2007a), Serial Markedness Reduction (Jarosz 2014), local constraint conjunction (Smolensky 1995; Ito & Mester 2003), and output-output faithfulness (Benua 1997).

### 4.2.1 OT with Candidate Chains

OT with Candidate Chains (OT-CC; McCarthy 2007a) uses HS to analyze counterfeeding and counterbleeding. OT-CC essentially combines the OT framework with a derivational framework by storing faithfulness violations of intermediate forms during candidate evaluation. Candidates therefore consist of a chain of all intermediate output forms in addition to the final optimal output. After these chains are generated, EVAL has access to both the terminal link (the final, most harmonic member) of the chain and the path of improvement, given by the ordering of the locally unfaithful, intermediate mappings.

Markedness constraints evaluate the final member of the chain and faithfulness constraints evaluate the relationship between the first and last forms in the chain. OT-CC faithfulness constraints are similar to our UO constraints in that they demand faithfulness between the first/underlying form and the last/output form in the HS derivation. The candidate chains are evaluated by a new type of constraint, a precedence (PREC) constraint, which evaluates the complete derivations by specifying a preferred order of faithfulness violations. This effectively controls the order in which markedness violations are repaired.

One of the major critiques of OT-CC is that the large amount of new capacity added to basic HS only re-creates the derivational account. Kiparsky (2015: 9) writes that the additions of OT-CC make a systems which is “like stipulative rule ordering...only with constraint ranking dictating the order of application.” However, this approach accounts for a wide range of opaque phenomena, including counterfeeding, counterbleeding, and fed counterfeeding (Kavitskaya & Staroverov 2010). Our proposal accounts for counterfeeding and counterbleeding, but cannot account for fed counterfeeding (see §2.4 for an explanation of the unique analytical challenge posed by fed counterfeeding). Another benefit of OT-CC is that it has the potential to constrain instances of the *too many repairs* problem (e.g. Lombardi 2001; Steriade 2001; Wilson 2001). Unattested repairs could be prevented from surfacing in OT-CC due to a refined definition of gradualness combined with the use of PREC constraints (McCarthy 2006).

We argue that our approach is more economical in two major ways: additional representations and CON. OT-CC adds the new class of PREC constraints to evaluate candidate chains and a metaconstraint to control rankings of PREC constraints. While our approach does introduce new constraints, they are within the class of faithfulness constraints and are composed of focuses and environments from existing markedness constraints. We also do not require a ranking metaconstraint to mitigate typological effects. Even if contextual faithfulness constraints are included in the universal CON, they only add opaque patterns and a case of blocking which is formally similar to observed cases of *do something except*

when blocking. As detailed in the previous section, the blocking pattern and an overly rich inventory of faithfulness constraints could be avoided if the constraints are induced on a language-specific basis, which we see as a promising avenue for future work.

Our approach is also more economical with regards to candidate representation. OT-CC stores intermediate representations during candidate evaluation to create a new type of representation, the candidate chain. The chain of intermediate forms must be stored and available throughout the derivation. Our analysis does not require an enriched theory of candidate representation through storage of any additional representations. We only require that the UR be referenced by  $\text{FAITH}_{\text{UR}}$  constraints. We assume that the UR would already be stored in the lexicon, so this form need not be additionally stored, only referenced. No additional stored forms or references are needed to evaluate  $\text{FAITH}_{\text{UR}}$  constraints.

#### 4.2.2 Serial Markedness Reduction

Serial Markedness Reduction (SMR; Jarosz 2014) also uses HS, and introduces a new family of constraints that work within the derivation itself, evaluating input-output mappings instead of entire chains of candidates (as in OT-CC). These constraints evaluate the order in which markedness constraints are satisfied. This is similar in concept to OT-CC's PREC constraints, which evaluate the order of faithfulness violations across candidates. However, in SMR, there is no requirement for storage of intermediate forms. Jarosz (2014) argues that this adds considerably less machinery to basic HS relative to OT-CC and also avoids the problems of global rule interactions admitted by OT-CC.

In SMR, the candidate representations are elaborated to include a list of markedness constraints satisfied at each step. This information is stored on the candidates themselves in the form of a list, called the *Mseq*. The candidates are evaluated against a standard CON with the addition of a new class of constraint, the serial markedness (SM) constraint. The SM constraint evaluates each candidate's *Mseq*. We provide the general form for SM constraints in (41), where M1 and M2 represent two markedness constraints.

- (41) General form for SM constraints (Jarosz 2014: 5)  
 SM (M1, M2): M2 must not precede or occur simultaneously with M1. (“One violation for each occurrence of M2 that precedes or occurs simultaneously with M1”)

SMR adds considerably less technology to basic HS (relative to OT-CC), while maintaining the ability to account for many opaque phenomena. As in basic HS, introducing SMR only requires one EVAL loop per step of the derivation. It does not require a “second evaluation with an expanded set of constraints, as in OT-CC” (referring to PREC constraints) (Jarosz 2014: 6). This local evaluation, combined with the fact that SM constraints are vacuously satisfied unless both relevant markedness constraints are present in a candidate's *Mseq*, mitigate the issues of global rule interaction admitted by OT-CC. OT-CC allows interactions which are not allowed by rule ordering, such as counterfeeding from the past (Wilson 2006). Jarosz argues that OT-CC “cannot predict an opaque interaction between processes that are ordered far apart in the derivation because process interaction is defined locally” (Jarosz 2014: 12). SMR is overall argued to be more economical and more realistic with respect to typological predictions relative to OT-CC.

Our approach bears some similarities to SMR, as contextual faithfulness constraints also work within the derivation and evaluate candidate mappings. Neither approach requires storage of a chain of intermediate steps. Both approaches also order the satisfaction of markedness constraints. SMR does this directly, by keeping track of what markedness

constraints have been satisfied and evaluating the order of satisfaction with a new class of constraint.

In our approach, markedness constraint satisfaction does not have to be ordered directly, but instead emerges from the use of faithfulness constraints. Once the context has changed, the constraint is no longer active. This provides a way of ordering satisfaction of markedness constraints without explicitly demanding it. Our approach does not require the introduction of a new class of constraint (the SM constraint). Instead we require new constraints introduced into the class of faithfulness constraints, which is already present in basic HS and OT.

Our approach differs from both OT-CC and SMR in that it does not require an enriched theory of candidate representation, as long as we assume underlying lexical representations are already stored in speakers' lexicons. Our constraints do require access to those underlying representations at each step of the derivation, but does not require access to information about any intermediate steps, or previous constraints satisfied. For these reasons, we argue that our approach is more economical.

#### 4.2.3 Local constraint conjunction

Local constraint conjunction has been proposed as a richer faithfulness system for OT (Smolensky 1995; Ito & Mester 2003). The local conjunction of two constraints is violated when both conjoined constraints are violated. These constraints have been explored as a method of analyzing counterfeeding opacity, but cannot account for other cases of opacity (McCarthy 1999; Padgett 2002). However, a benefit of this approach is that it is also able to account for many other non-opaque phenomena including coda conditions (Smolensky 1993) and derived environment effects (Łubowicz 2002; Ito & Mester 2003).

Critiques of this approach have often involved typology, with the claim being that introducing the ability to conjoin constraints overpredicts observed patterns. Because local conjunction relies on the segmental proximity of violations in a specified domain, enlarging the domain (to the entire word for example) predicts interactions with triggers over a nonlocal context. Such interactions are not associated with process interaction and are an undesirable prediction of the system. Łubowicz (2005) showcases this issue and proposes that the domain of conjunction must be restricted to avoid predicting unattested patterns where, for example, palatalization triggered by a vowel context in one part of the word can trigger spirantization in another part of the word.

Analyzing opacity with contextual faithfulness constraints does not run the same risk of overprediction because they only reference a single segment as their focus, not an entire domain. This is not to say there is no risk for overprediction with contextual faithfulness—see §4.1 for discussion of typological predictions. McCarthy (2007a) argues that local conjunction account of counterfeeding opacity therefore relies solely on the proximity of the two violations, in some specified and crucially overlapping domain, obscuring the actual interaction of two processes. We argue that  $\text{FAITH}_{\text{VO}}$  constraints provide a more explanatory account by combining the pre-existing mechanism for producing contrast in OT, the interaction of markedness and faithfulness constraints, with the generalization that counterfeeding opacity is characterized by a special case of faithfulness in a particular context.

There is a formal similarity between our contextual faithfulness constraints and local constraint conjunction with respect to the potential for constraint induction. While contextual faithfulness constraints are only single (non-conjoined) constraints, they could be induced through the combination of pre-existing markedness constraints. However, because they are evaluated as a single non-conjoined constraint, they avoid the typological overprediction of conjoined constraints discussed by McCarthy (2007a).

#### 4.2.4 Output-output faithfulness

Output-output faithfulness (OO-FAITH; Burzio 1994; Benua 1997) has also been used to analyze opacity. This approach bears many analytical similarities to our contextual faithfulness approach. In an OO-FAITH analysis of opacity, “opacity is conditioned by a faithfulness relation between the output and the output of another related form” (Goldrick 2001: 12). Both our approach and OO-FAITH provide extra faithfulness for a particular segment/feature in a particular context between the output and some other representation. For OO-FAITH, this other representation is a related output form, often a form where a crucial feature from the lexical form/UR surfaces faithfully. In our approach, we instead implement a direct faithfulness relationship between the lexical form and the output of the current derivation. Effectively both approaches demand additional faithfulness, but refer to different representations.

We argue that our approach provides a more economical way of obtaining this extra faithfulness, as we only require access to either the input of the current stage of the derivation or the underlying form (depending on whether the constraint is IO or UO). This does not require CON to have access to additional intermediate forms, output forms, or forms associated with other derivations. Our approach does not require storage of any additional forms that are not already stored in the lexicon. In addition, our approach provides a general solution to counterbleeding and counterfeeding opacity, while OO-FAITH is restricted to cases of opacity where there is a related morphological form where the UR surfaces faithfully. The OO-FAITH approach can only account for particular cases of counterbleeding and does not offer a general solution for opacity in OT.

## 5 Conclusion

In this paper, we have proposed a general solution for analyzing opacity in constraint-based grammars by using contextual faithfulness constraints, which demand faithfulness in a specified input context. We have shown how these constraints can account for several types of underapplication opacity in parallel OT and HS. When implemented in Harmonic Serialism, our analysis can also account for overapplication opacity, with the addition of a distinction between faithfulness to the input of the current step of the derivation (FAITH<sub>IO</sub>) and faithfulness to the underlying representation (FAITH<sub>UO</sub>). Contextual faithfulness constraints are conceptually similar to positional faithfulness constraints, but are not limited to prominent contexts, and therefore can be used to analyze a wide range of opaque phenomena. Unlike many previous analyses of opacity in OT (and variants thereof), this approach does not introduce additional representations or significant changes to the standard grammatical system, only additional faithfulness constraints. All elements of our proposed constraints (faithfulness to particular contexts, faithfulness to the UR) have previously been used in the literature, but have not been combined to offer a general analysis of opacity, as we have demonstrated here.

We have argued that the creation of an overly rich faithfulness theory should not be immediately detrimental to the analysis. A rich CON does not necessarily lead to pathological typological predictions—the only languages which are added to our example typology are the attested opaque patterns and a blocking pattern which is formally similar to attested patterns of *do something except when* blocking. We have also provided a sketch of how these constraints might be induced on a language-specific basis instead of being included in the universal CON. This procedure would use particular ranking inconsistencies to diagnose opacity, then draw on the forms of pre-existing constraints to construct new contextual faithfulness constraints. Future work includes computational implementation of such a model and further typological testing.

## Abbreviations

OT = Optimality Theory, HS = Harmonic Serialism, HB = Hijazi Bedouin, SG = singular, PL = plural, OT-CC = Optimality Theory with Candidate Chains, IO = input-output, UO = underlying-output, OO = output-output, ID = IDENT, UR = underlying representation, ROTB = richness of the babse, DSEW = do something except when, RCD = recursive constraint demotion, VCDOBS = voiced obstruents, SMR = Serial Markedness Reduction, SM = serial markedness.

## Acknowledgements

Many thanks to John McCarthy, Gaja Jarosz, Joe Pater, Andrew Lamont, audiences at AMP 2014, AMP 2016, the University of Massachusetts Amherst Sound Workshop, the University of North Carolina Linguistics Colloquium, and the University of Leipzig IGRA Colloquium for input on this work.

## Funding Information

This material is based upon work supported by the National Science Foundation Graduate Research Fellowship Program under Grant Number 1451512. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

## Competing Interests

The authors have no competing interests to declare.

## References

- Al-Mozainy, Hamza Qublan. 1981. *Vowel alternations in a Bedouin Hijazi Arabic dialect: Abstractness and stress*. Austin, TX: University of Texas at Austin dissertation.
- Baković, Eric. 2007. A revised typology of opaque generalisations. *Phonology* 24(2). 217–259. DOI: <https://doi.org/10.1017/S0952675707001194>
- Baković, Eric. 2011a. Opacity and ordering. In John Goldsmith, Jason Riggle & Alan C. L. Yu (eds.), *The handbook of phonological theory*, 40–67. Malden, MA: Wiley-Blackwell. DOI: <https://doi.org/10.1002/9781444343069.ch2>
- Baković, Eric. 2011b. Opacity deconstructed. Ms. University of California San Diego.
- Beckman, Jill. 1998. *Positional faithfulness*. Amherst, MA: University of Massachusetts Amherst dissertation.
- Beckman, Jill N. 1997. Positional faithfulness, positional neutralisation and Shona vowel harmony. *Phonology* 14(1). 1–46. DOI: <https://doi.org/10.1017/S0952675797003308>
- Benua, Laura. 1997. *Transderivational identity: Phonological relations between words*. Amherst, MA: University of Massachusetts Amherst dissertation.
- Bermúdez-Otero, Ricardo. 2003. The acquisition of phonological opacity. In *Proceedings of the Stockholm workshop on variation within Optimality Theory*, 25–36.
- Burzio, Luigi. 1994. *Principles of English stress*. Cambridge: Cambridge University Press. DOI: <https://doi.org/10.1017/CBO9780511519741>
- Elfner, Emily. 2016. Stress-epenthesis interactions in Harmonic Serialism. In John McCarthy & Joe Pater (eds.), *Harmonic Grammar and Harmonic Serialism*, 261–300. Sheffield: Equinox Press.
- Gnanadesikan, Amalia. 1997. *Phonology with ternary scales*. Amherst, MA: University of Massachusetts Amherst dissertation.
- Goldrick, Matthew. 2001. Turbid output representations and the unity of opacity. In Masako Hirotani, Andries Coetzee, Nancy Hall & Jeong-Young Kim (eds.), *Proceedings of NELS 50*, 231–245. Amherst, MA: GLSA.

- Hulstaert, Gustaaf. 1961. *Grammaire du Lomongo*, vol. 1. Tervuren: Musée royal de l'Afrique centrale.
- Ito, Junko & Armin Mester. 2003. On the sources of opacity in OT: Coda processes in German. In Caroline Féry & Ruben van de Vijver (eds.), *The syllable in Optimality Theory*, 271–303. Cambridge: Cambridge University Press. DOI: <https://doi.org/10.1017/CBO9780511497926.012>
- Jarosz, Gaja. 2014. Serial markedness reduction. In *Proceedings of the 2013 annual meeting on phonology*. Washington, DC: Linguistic Society of America. DOI: <https://doi.org/10.3765/amp.v1i1.40>
- Jarosz, Gaja. 2016. Learning opaque and transparent interactions in Harmonic Serialism. In *Proceedings of the 2015 annual meeting on phonology*. Washington, DC: Linguistic Society of America. DOI: <https://doi.org/10.3765/amp.v3i0.3671>
- Jesney, Karen. 2005. *Chain shift in phonological acquisition*. Calgary, AB: University of Calgary MA thesis.
- Jesney, Karen. 2011. Positional faithfulness, non-locality, and the Harmonic Serialism solution. In Suzi Lima, Kevin Mullin & Brian Smith (eds.), *Proceedings of the 39th annual meeting of the North East Linguistic Society*. Amherst, MA: GLSA.
- Kavitskaya, Darya & Peter Staroverov. 2010. When an interaction is both opaque and transparent: the paradox of fed counterfeeding. *Phonology* 27(02). 255–288. DOI: <https://doi.org/10.1017/S0952675710000126>
- Kiparsky, Paul. 1968. Linguistic universals and linguistic change. In Emmon Bach & Robert Harms (eds.), *Universals in linguistic theory*, 170–202. New York, NY: Holt, Rinehart & Winston.
- Kiparsky, Paul. 1973. Abstractness, opacity and global rules. In Osamu Fujimura (ed.), *Three dimensions of linguistic theory*, 57–86. Tokyo: TEC.
- Kiparsky, Paul. 2000. Opacity and cyclicity. *The Linguistic Review* 17(2–4). 351–366. DOI: <https://doi.org/10.1515/tlir.2000.17.2-4.351>
- Kiparsky, Paul. 2003. Syllables and moras in Arabic. In Caroline Féry & Ruben van de Vijver (eds.), *The syllable in Optimality Theory*, 147–182. Cambridge: Cambridge University Press. DOI: <https://doi.org/10.1017/CBO9780511497926.007>
- Kiparsky, Paul. 2015. Stratal OT: A synopsis and FAQs. In Yuchau Hsiao & Lian-Hee Wee (eds.), *Capturing phonological shades within and across languages*, 1–45. Newcastle: Cambridge Scholars Publishing.
- Kirchner, Robert. 1996. Synchronic chain shifts in Optimality Theory. *Linguistic Inquiry* 27(2). 341–350.
- Kisseberth, Charles. 1969. *Theoretical implications of Yawelmani phonology*. Champaign, IL: University of Illinois at Urbana-Champaign dissertation.
- Kisseberth, Charles. 1970. On the functional unity of phonological rules. *Linguistic Inquiry* 1(3). 291–306.
- Kochetov, Alexei. 2016. Palatalization and glide strengthening as competing repair strategies: Evidence from Kirundi. *Glossa* 1(1). DOI: <https://doi.org/10.5334/gjgl.32>
- Kuroda, Shige Yuki. 1967. *Yawelmani phonology*. Cambridge, MA: MIT Press.
- Lombardi, Linda. 1999. Positional faithfulness and voicing assimilation in Optimality Theory. *Natural Language & Linguistic Theory* 17(2). 267–302. DOI: <https://doi.org/10.1023/A:1006182130229>
- Lombardi, Linda. 2001. Why place and voice are different: Constraint-specific alternations in Optimality Theory. In Linda Lombardi (ed.), *Segmental phonology in optimality theory*, 13–45. Cambridge: Cambridge University Press. DOI: <https://doi.org/10.1017/CBO9780511570582.002>

- Lubowicz, Anna. 2002. Derived environment effects in Optimality Theory. *Lingua* 112(4). 243–280. DOI: [https://doi.org/10.1016/S0024-3841\(01\)00043-2](https://doi.org/10.1016/S0024-3841(01)00043-2)
- Lubowicz, Anna. 2005. Locality of conjunction. In John Alderete, Chung hye Han & Alexei Kochetov (eds.), *Proceedings of the 24th west coast conference on formal linguistics*, 254–262. Cascadilla Press.
- McCarthy, John J. 1999. Sympathy and phonological opacity. *Phonology* 16(03). 331–399. DOI: <https://doi.org/10.1017/S0952675799003784>
- McCarthy, John J. 2000. Harmonic serialism and parallelism. In Masako Hirotsu, Andries Coetzee, Nancy Hall & Ji yung Kim (eds.), *Proceedings of the 30<sup>th</sup> meeting of the North East Linguistic Society*, 501–524. Amherst, MA: GLSA.
- McCarthy, John J. 2006. Candidates and derivations in Optimality Theory. Ms. University of Massachusetts Amherst.
- McCarthy, John J. 2007a. *Hidden generalizations: Phonological opacity in Optimality Theory*. Sheffield: Equinox.
- McCarthy, John J. 2007b. Restraint of analysis. In Sylvia Blaho, Patrik Bye & Martin Krämer (eds.), *Freedom of analysis?*, 203–231. Berlin: Mouton de Gruyter.
- McCarthy, John J. 2008. The serial interaction of stress and syncope. *Natural Language and Linguistic Theory* 26(3). 499–546. DOI: <https://doi.org/10.1007/s11049-008-9051-3>
- McCarthy, John J. 2010. An introduction to Harmonic Serialism. *Language and Linguistics Compass* 4(10). 1001–1018. DOI: <https://doi.org/10.1111/j.1749-818X.2010.00240.x>
- McCarthy, John J. 2018. How to delete. In Amel Khalfaoui & Matthew Tucker (eds.), *Perspectives on Arabic linguistics XXX: Papers from the annual symposia on Arabic linguistics*, 7–32. Amsterdam: John Benjamins. DOI: <https://doi.org/10.1075/sal.7.02mcc>
- Moreton, Elliot & Paul Smolensky. 2002. Typological consequences of local constraint conjunction. In Line Mikkelsen & Christopher Potts (eds.), *Proceedings of the 21st west coast conference on formal linguistics*, 306–319. Cascadilla Press.
- Newman, Stanley. 1944. *The Yokuts language of California*. New York, NY: Viking Fund.
- Padgett, Jaye. 2002. Constraint conjunction versus grounded constraint subhierarchies in Optimality Theory. Ms. University of California Santa Cruz.
- Pater, Joe. 2010. Morpheme-specific phonology: Constraint indexation and inconsistency resolution. In Steve Parker (ed.), *Phonological argumentation: Essays on evidence and motivation*, 123–154. Sheffield: Equinox.
- Prince, Alan & Paul Smolensky. 1993/2004. *Optimality theory: Constraint interaction in generative grammar*. Malden, MA: Wiley-Blackwell.
- Pruitt, Kathryn. 2010. Serialism and locality in constraint-based metrical parsing. *Phonology* 27(3). 481–526. DOI: <https://doi.org/10.1017/S0952675710000229>
- Smolensky, Paul. 1993. Harmony, markedness, and phonological activity. In *Rutgers optimality workshop* 1. 87–100.
- Smolensky, Paul. 1995. On the structure of the constraint component Con of UG. Ms, Johns Hopkins University.
- Smolensky, Paul. 1996. The initial state and ‘richness of the base’ in Optimality Theory. Ms, Johns Hopkins University.
- Staub, Robert & Joe Pater. 2012. Learning serial constraint-based grammars. In John McCarthy & Joe Pater (eds.), *Harmonic Grammar and Harmonic Serialism*, 369–388. Sheffield: Equinox.
- Staub, Robert, Michael Becker, Christopher Potts, Patrick Praat, John McCarthy & Joe Pater. 2010. OT-Help 2. Software Package. University of Massachusetts Amherst.
- Steriade, Donca. 2001. The phonology of perceptibility effects: The P-map and its consequences for constraint organization. Ms. University of California Los Angeles.



- Tesar, Bruce & Paul Smolensky. 1998. Learnability in Optimality Theory. *Linguistic Inquiry* 29(2). 229–268. DOI: <https://doi.org/10.1162/002438998553734>
- Tessier, Anne-Michelle. 2013. Error-driven learning in Harmonic Serialism. In Shayne Sloggett & Stefan Keine (eds.), *Proceedings of the 42nd meeting of the North East Linguistics Society*, 545–558. Amherst, MA: GLSA.
- Tessier, Anne-Michelle & Karen Jesney. 2014. Learning in Harmonic Serialism and the necessity of a richer base. *Phonology* 31(1). 155–178. DOI: <https://doi.org/10.1017/S0952675714000062>
- Wilson, Colin. 2001. Consonant cluster neutralisation and targeted constraints. *Phonology* 18(01). 147–197. DOI: <https://doi.org/10.1017/S0952675701004043>
- Wilson, Colin. 2006. Counterfeeding from the past. Ms. Johns Hopkins University.

**How to cite this article:** Hauser, Ivy and Coral Hughto. 2020. Analyzing opacity with contextual faithfulness constraints. *Glossa: a journal of general linguistics* 5(1): 82.1–33. DOI: <https://doi.org/10.5334/gjgl.966>

**Submitted:** 10 April 2019    **Accepted:** 14 April 2020    **Published:** 20 August 2020

**Copyright:** © 2020 The Author(s). This is an open-access article distributed under the terms of the Creative Commons Attribution 4.0 International License (CC-BY 4.0), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited. See <http://creativecommons.org/licenses/by/4.0/>.



*Glossa: a journal of general linguistics* is a peer-reviewed open access journal published by Ubiquity Press.

