# Serial Markedness Reduction

Gaja Jarosz
*Yale University*

## 1 Introduction

Recent work in Harmonic Serialism (HS) has led to new insights into several outstanding problems in phonology. The gradual, serial optimization inherent to HS offers novel perspectives on the too-many-repairs problem (McCarthy 2008a) and makes it possible to capture generalizations stated at intermediate levels of representation (McCarthy 2008b, Elfner to appear). This work has also highlighted problematic over-predictions of parallel OT resulting from an excessive power to do global optimization and shown that HS predicts a range of locality effects that better reflect typology (McCarthy 2009, Pruitt 2010). Existing work has shown that HS can model limited types of opaque interactions that are problematic for parallel OT (Elfner to appear). However, despite its serialist nature, HS does not provide a general solution to phonological opacity. For this reason McCarthy developed a significant elaboration of HS, OT with Candidate Chains (OT-CC; McCarthy 2007), which evaluates complete HS-like derivations in parallel. McCarthy proposed OT-CC as a general theory of opacity and demonstrated that it is capable of modeling both counterfeeding and counterbleeding opacity using PREC constraints, which evaluate the order in which operations apply in a derivation. A general approach to modeling these interactions in constraint-based theories has proven elusive so it is a major strength of OT-CC that it provides a mechanism capable of capturing these generalizations. At the same time, OT-CC requires significant machinery beyond that of HS or OT to model these interactions, and some have suggested this additional machinery is too powerful, admitting global rule interactions such counterfeeding from the past (Wilson 2006; cf. Wolf 2010).

This paper contributes to the growing body of work on HS and the ongoing debate about constraint-based approaches to opacity by proposing a mechanism that enables the modeling of both counterfeeding and counterbleeding interactions in HS proper. The new proposal, Serial Markedness Reduction (SMR), introduces a new family of constraints inspired by, but crucially different from, OT-CC's PREC constraints. The SMR constraints evaluate aspects of phonological derivations much like PREC constraints; however, unlike PREC constraints, SMR constraints work on the fly as derivations are gradually constructed. SMR is a minimal elaboration of HS, and it significantly streamlines the machinery necessary to model counterfeeding and counterbleeding opacity as compared to OT-CC. It also opens the door to a novel, local formulation of process interaction that avoids the need to construct and inspect complete derivations, as in OT-CC. Because process interaction is defined locally, SMR does not predict global rule interactions.

## 2 Background

This section reviews two familiar examples of counterfeeding and counterbleeding process interactions and discusses why HS cannot provide a general solution to such interactions (see McCarthy 2007 for extensive discussion, including discussion of the challenges these interactions pose for parallel OT).

**2.1** *Counterfeeding and Counterbleeding* A phonological process A potentially bleeds another process B if application of A eliminates potential inputs to B. In standard serialist terminology, if A is ordered before B, the resulting interaction is called bleeding, while if B precedes A, the result is counterbleeding. Counterbleeding is a kind of opaque interaction since it results in applications of B that are not surface-apparent, since subsequent application of A has destroyed the original motivation for B's application (Kiparsky 1973, McCarthy 1999). An example of counterbleeding opacity in Bedouin Arabic is

[*] I am grateful to Ryan Bennett and John McCarthy for helpful discussion.

Proceedings of *Phonology 2013*
Completed December 20, 2013

shown in (1). Deletion of high vowels in open, nonfinal syllables is responsible for mapping underlying /ʃarib-at/ 'she drank' to surface [ʃarbat], while palatalization of velars adjacent to front vowels results in the palatalization of the underlying /k/ in /ħaːkim/ → [ħaːk$^j$im] 'ruling (masc. sg.)'. The interaction, shown in the last column of (1), is one of counterbleeding since the underlying /k/ in /ħaːkim-iːn/ 'ruling (masc. pl)' palatalizes despite subsequent deletion of the medial [i] that conditions palatalization.

1) Bedouin Arabic **Counterbleeding** (McCarthy 2007, Al-Mozainy 1981)

| | a. Deletion | b. Palatalization | c. Interaction |
|---|---|---|---|
| Underlying | /ʃarib-at/ | /ħaːkim/ | /ħaːkim-iːn/ |
| Palatalization | — | ħaːk$^j$im | ħaːk$^j$imiːn |
| Deletion | ʃarbat | — | ħaːk$^j$miːn |
| Surface | [ʃarbat] | [ħaːk$^j$im] | [ħaːk$^j$miːn] |
| | *'she drank'* | *'ruling (masc. sg.)'* | *'ruling (masc. pl.)'* |

A potential feeding interaction occurs when phonological process A creates potential inputs to process B. Feeding results if A precedes B, while counterfeeding occurs when B precedes A. Counterfeeding is a kind of opaque interaction because the generalization expressed by B is not surface-true, in the sense that there are surface violations of B (Kiparsky 1973, McCarthy 1999). An example of counterfeeding in Bedouin Arabic is shown in (2). In Bedouin Arabic there is raising of /a/ to [i] in open, nonfinal syllables: /katab/ → [kitab] 'he wrote', and there is epenthesis into final clusters: /gidr/ → [gidir] 'pot'. The last column of (2) illustrates the counterfeeding interaction in which an /a/ that is placed into an open syllable as a consequence of epenthesis fails to raise: /gabr/ → [gabur] 'grave'.

2) Bedouin Arabic **Counterfeeding** (McCarthy 1999, 2007, Al-Mozainy 1981)

| | a. Raising | b. Epenthesis | c. Interaction |
|---|---|---|---|
| Underlying | /katab/ | /gidr/ | /gabr/ |
| Raising | kitab | | — |
| Epenthesis | — | gidir | gabur |
| Surface | [kitab] | [gidir] | [gabur] |
| | *'he wrote'* | *'pot'* | *'grave'* |

**2.2** *Harmonic Serialism and Opacity* Like parallel OT, Harmonic Serialism assumes that a single, fixed ranking of constraints defines the grammar of a language. However, HS differs from parallel OT in two fundamental ways. First, HS restricts GEN: the candidate set includes only those candidates that can be created by applying one basic operation to the input form (what exactly constitutes one operation is a topic of ongoing research). Second, HS applies EVAL recursively: the output of each optimization is fed back into EVAL iteratively until no further changes occur. In this way, HS determines the optimal outputs in each language via a serial derivation that applies operations in an order based on a fixed ranking of constraints.

Despite its serialist nature, HS does not offer a general solution to opacity. While the ranking of constraints can define a language's preference with regard to the order in which operations apply, it is precisely in those cases where the ordering of operations is crucial to the outcome that HS has no control over ordering, favoring transparent interactions as the default case. This is because the constraints involved in potential feeding and bleeding interactions affect each other's violations in ways that systematically favor feeding and bleeding over their opaque counterparts. The way these systematic preferences manifest themselves is different in counterbleeding than in counterfeeding opacity, and these are addressed in turn.

Consider again the counterbleeding interaction in (1) and the attempt to capture this interaction in HS in (3) (constraints from McCarthy 2007). Palatalization requires the ranking of a markedness constraint that penalizes plain velars when adjacent to high vowels, abbreviated *ki, above a constraint penalizing changes in consonant backness, IDENT[BACK]. Deletion of high vowels in open non-final syllables is compelled by *iCV, which must be ranked above MAX for deletion to be preferred. To produce counterbleeding, palatalization must apply first and be followed by deletion on the next iteration. However, given these rankings, which are necessary to account for the independent processes of palatalization and deletion, HS fails to select the desired candidate on the first iteration, as shown in (3). There is no way to favor palatalization (3b) over the transparent candidate that undergoes deletion (3c). Deletion (3c) satisfies both *_ki_ and *_i_CV while incurring only one low-ranked faithfulness violation (Max). In comparison to the deletion candidate, the palatalizing candidate (3b) has a gratuitous violation of high-ranked *_i_CV.

3) Counterbleeding in HS: Iteration 1

| /ħaːkimiːn/ | *_ki_ | Ident[back] | *_i_CV | Max |
|---|---|---|---|---|
| a. ħaː.ki.miːn | W* | L | * | |
| ☆ b. ħaː.kʲi.miːn | | * | * | |
| ☛ c. ħaːk.miːn | | L | L | W* |

This example illustrates the general problem counterbleeding opacity poses for HS. Schematically, a process in HS requires that some markedness constraint be ranked above some faithfulness constraint. If process A (deletion), modeled as $M_A \gg F_A$ (*_i_CV ≫ Max), potentially bleeds process B (palatalization), modeled as $M_B \gg F_B$ (*_ki_ ≫ Ident[back]), this means that application of A (deletion) destroys the environment for B (palatalization). In terms of constraint interaction, this means application of process A (deletion) causes the markedness constraint responsible for process B, $M_B$ (*_ki_), to become satisfied. In other words, the essential characteristic of a potential bleeding interaction is that satisfaction of $M_A$ (*_i_CV) results in the satisfaction of both $M_A$ (*_i_CV) and $M_B$ (*_ki_). Under these conditions, a process that satisfies only $M_B$ (palatalization) stands no chance. Put differently, HS provides no general mechanism to disfavor bleeding, only mechanisms to favor it.

Counterfeeding presents a different challenge to HS, which can be illustrated with reference to the example in (2). To account for the Bedouin Arabic process of epenthesis, a constraint penalizing final clusters *CC must be ranked above Dep, and to account for raising, a constraint penalizing low vowels in open non-final syllables, *_a_CV, must be ranked above Ident[low] (constraints from McCarthy 2007). The opaque interaction between epenthesis and raising requires that epenthesis applies but is not followed by raising. HS has no problem capturing the first step of the derivation, as shown in (4). As long as *CC is ranked above *_a_CV, the candidate with epenthesis (4b), which incurs violations of *_a_CV, will be optimal. At this point in the derivation, raising is not motivated and poses no problem. However, as shown in (5), once epenthesis has applied, there is no way to prevent raising from occurring, given that the ranking of *_a_CV above Ident[low] is necessary independently to account for raising. Unlike OT-CC, HS has no memory so it cannot distinguish an input like /gabur/ from an input like /katab/ even though only the former has been derived through the application of epenthesis.

4) Counterfeeding in HS: Iteration 1

| /gabr/ | *CC | Dep | *_a_CV | Ident[low] |
|---|---|---|---|---|
| a. gabr | W* | L | L | |
| ☞ b. gabur | | * | * | |
| c. gibr | W* | L | L | W* |

5) Counterfeeding in HS: Iteration 2

| /gabur/ | *CC | Dep | *_a_CV | Ident[low] |
|---|---|---|---|---|
| ☆ a. gabur | | | * | |
| ☛ b. gibur | | | L | W* |

This example illustrates the general problem posed by counterfeeding for HS. If process A (epenthesis), modeled as $M_A \gg F_A$, potentially feeds process B (raising), modeled as $M_B \gg F_B$, this means application of process A (epenthesis) creates violations of $M_B$ (*_a_CV), the constraint compelling process B (raising). Once the violations of $M_B$ are created, the ranking $M_B \gg F_B$ necessitates that process B applies on a subsequent iteration, and any attempt to prevent B from applying to these cases would eliminate process B from the language entirely. In general, HS has no record of the original input and therefore provides no way to systematically distinguish $M_B$ violations that are created by applying process A from $M_B$ violations present in the input or created by other processes. In sum, if the ranking is such that process A applies in cases of potential feeding ($M_A \gg M_B$), this means violations of $M_B$ will be created and then subsequently resolved, resulting in feeding rather than counterfeeding as the systematically favored outcome.

## 3 Serial Markedness Reduction

The previous section has shown that potential feeding and bleeding interactions in HS lead to interactions in the satisfaction of the markedness constraints responsible for the processes involved. If A potentially bleeds B, then A entirely erases the motivation for B, leading to a systematic preference for bleeding over counterbleeding. If A potentially feeds B and A applies, then A creates the motivation for B, and B cannot be stopped, leading to a systematic preference for feeding over counterfeeding. Lack of constraints favoring opaque interaction means transparent interactions are predicted in the general case.

To counter Harmonic Serialism's systematic preferences for transparent interaction, some mechanism penalizing transparent interaction is required. In order for the mechanism to provide a general solution to counterfeeding and counterbleeding opacity, it must reference the characteristic properties that distinguish these interactions from transparent ones. As the previous section has demonstrated, these characteristic properties have to do with how the satisfaction of one markedness constraint interacts with the satisfaction or violation of another. The novel contribution of Serial Markedness Reduction is to characterize opaque interactions in these general terms. SMR does this by examining how operations applied at each iteration affect the satisfaction of markedness constraints and how the satisfaction of markedness constraints on the current iteration relates to the markedness satisfaction that occurred on earlier iterations.

A unified treatment of counterfeeding and counterbleeding opacity is possible using the proposed family of Serial Markedness constraints, which evaluate the order in which markedness constraints are satisfied in a derivation. While these constraints are inspired by and superficially resemble OT-CC's PREC constraints, it turns out that SMR's focus on markedness satisfaction rather than faithfulness violation has significant consequences, not the least of which is the possibility of incorporating SM constraints directly into regular HS optimization. The rest of this section illustrates the SMR proposal, and the next section discusses some of the consequences of characterizing opacity in terms of markedness reduction.

**3.2** *Serial Markedness Reduction Basics* SMR endows SM constraints with the capacity to evaluate the order in which markedness constraints are satisfied in a derivation. To access this information about the derivation, the proposed extension of HS simply records which constraints are satisfied on each iteration in a list called *Mseq*. *Mseq* is part of the candidate, it gets updated each iteration based on which constraints are satisfied, and then it gets passed down to the next iteration as part of the input.

There are several ways that *Mseq* could be constructed. For concreteness, the formulation adopted here assumes that *Mseq* compares the markedness violations of each candidate to the violations incurred by the candidate that is identical to the input. To make sure this is well defined in cases where candidates have structure that is lacking in underlying representations (such as syllable structure), the first step of every derivation is assumed to be the selection of the optimal parse, as in OT-CC. This initial parsing iteration, designated as iteration 0 below, is demonstrated in (6), where its function is to assign syllable structure to the input based on the ranking. Since the constraints employed in the remainder of the paper are not sensitive to structure, this step would apply vacuously and will be omitted for simplicity.

6) *Mseq* Illustration: Iteration 0

| /asba/ | *COMP | AGREE[VC] | ID | *$s]_\sigma$ | *VOICE$_{COD}$ |
|---|---|---|---|---|---|
| ☞ a. as.ba | | * | | * | |
| b. a.sba | * | * | | | |

On the first regular HS iteration, *Mseq* begins empty < > in the input of Iteration 1, shown in (7). Each unfaithful candidate is compared to the fully faithful candidate, marked with '=', and its relative reduction in markedness is encoded it its *Mseq*. Crucially, *Mseq* only encodes those markedness constraints that have become satisfied as a consequence of applying the operation in question as compared to the faithful candidate, which had no operations applied to it. Thus, by voicing the coda [s], candidate (c) satisfies both AGREE and *s]$_\sigma$, which is encoded in its *Mseq*. Voicing the [s] did not reduce the number of violations of any other markedness constraints as compared to faithful candidate so other satisfied constraints, such as *COMP, are not encoded in candidate (c)'s *Mseq* (since candidate (a) already satisfied these constraints).

7) *Mseq* Illustration: Iteration 1

| /as.ba/ < > | *Comp | Agree[vc] | Id | $*s]_\sigma$ | *Voice$_{COD}$ |
|---|---|---|---|---|---|
| = a. as.ba < > | | * | | * | |
| b. a.sba < $*s]_\sigma$> | * | * | | (satis) | |
| ☞ c. az.ba < Agree+$*s]_\sigma$ > | | (satis) | * | (satis) | * |
| d. a.zba < Agree+$*s]_\sigma$ > | * | (satis) | * | (satis) | |

*Mseq* is part of the candidate and therefore gets passed down to the next iteration, shown in (8). At this point this derivation converges, but the candidate in (8b) illustrates how construction of *Mseq* would continue. Each candidate's newly satisfied markedness constraints are appended to the *Mseq* specified in the input. Thus, construction of *Mseq* references only information available on the current iteration but records a history of the markedness constraints that have been satisfied during the course of the derivation.

8) *Mseq* Illustration: Iteration 2

| /az.ba/ < Agree+$*s]_\sigma$> | *Comp | Agree[vc] | Id | $*s]_\sigma$ | *Voice$_{COD}$ |
|---|---|---|---|---|---|
| = ☞ a. az.ba < Agree+$*s]_\sigma$ > | | | | | * |
| b. as.ba < Agree+$*s]_\sigma$, *VC$_{COD}$ > | | * | * | * | (satis) |

With this background, it is now possible to provisionally define Serial Markedness constraints, which reference *Mseq* in order to evaluate the order in which markedness constraints have been satisfied[1]:

9) **Serial Markedness** (First Version)
SM($M_1$, $M_2$): $M_2$ must not precede or occur simultaneously with $M_1$. ("One violation for each occurrence of $M_2$ that precedes or occurs simultaneously with $M_1$")

SM($M_1$, $M_2$) favors a derivation in which $M_1$ is satisfied before $M_2$. Intuitively, if $M_1$ is responsible for the earlier process, and $M_2$ is responsible for the later process, then SM($M_1$, $M_2$) enforces that derivational order between the two processes. This makes it possible to penalize feeding for satisfying the markedness constraints in the wrong order and bleeding for simultaneously, rather than serially, satisfying the relevant markedness constraints. Crucially, SM constraints can be incorporated directly into regular HS optimization, as derivations are gradually constructed. The rest of the section illustrates how this works.

### 3.3 *Counterbleeding Illustration*

For the counterbleeding case, an SM constraint that favors palatalization before deletion is needed, namely SM(**ki*, **i*CV). Recall that counterbleeding opacity poses a challenge for HS on the first pass through the grammar because the bleeding candidate satisfies both markedness constraints. As shown in (10), the simultaneous satisfaction of both markedness constraints is recorded in the *Mseq* for the transparent candidate (c) and results in a violation of SM(**ki*, **i*CV).

10) Counterbleeding in SMR: iteration 1

| /ħaːkimiːn/ < > | **ki* | SM(**ki*, **i*CV) | **i*CV | Max | Id[bk] |
|---|---|---|---|---|---|
| a. ħaːkimiːn < > | W* | | * | | L |
| ☞ b. ħaːk$^j$imiːn <**ki*> | | | * | | * |
| c. ħaːkmiːn <**i*CV+**ki*> | | W* | L | W* | L |

As long as SM(**ki*, **i*CV) is ranked above **i*CV and Id[bk], palatalization will be favored to deletion, which does not incur a violation of SM(**ki*, **i*CV). The SM constraint penalizes the transparent bleeding candidate for satisfying both markedness constraints at the same time on the first pass, allowing the desired candidate to win. If SM(**ki*, **i*CV) were ranked lower, the bleeding candidate would win. In this case, however, the winner accrues one markedness constraint in its *Mseq* and continues on to the next pass. As shown in (11), deletion is now free to apply since satisfying **i*CV on this pass orders it after the satisfaction of **ki*, consistent with the requirements of SM(**ki*, **i*CV). Since the resulting candidate no longer violates

[1] More formally, SM constraints evaluate instances of markedness violations, indexed by locus. See Section 6.

any active markedness constraints, the derivation will converge on the next pass.

11) Counterbleeding in SMR: iteration 2

| /ħaːkʲimiːn/ <*ki> | *ki | SM(*ki, *iCV) | *iCV | MAX | ID[BK] |
|---|---|---|---|---|---|
| a. ħaːkʲimiːn <*ki> | | | W* | L | |
| ☞ b. ħaːkʲmiːn <*ki, *iCV> | | | | * | |

**3.3** *Counterfeeding Illustration* For the counterfeeding case, an SM constraint that favors raising before epenthesis is needed, namely SM(*aCV, *CC). On the first pass, shown in (12), everything proceeds as in HS, with epenthesis selected as optimal. The SM constraint is silent at this point because the *Mseq* for the epenthetic candidate (b) does not encode any ordering between *CC and *aCV.

12) Counterfeeding in SMR: iteration 1

| /gabr/ < > | *CC | SM(*aCV, *CC) | DEP | *aCV | ID[LOW] |
|---|---|---|---|---|---|
| a. gabr <> | W* | | L | L | |
| ☞ b. gabur <*CC> | | | * | * | |

The optimal candidate (12b) continues on to the next iteration, shown in (13). At this point the SM constraint penalizes the transparent feeding candidate (13b) for satisfying *aCV too late. With SM(*aCV, *CC) ranked above *aCV, the derivation converges on the counterfeeding candidate. With the opposite ranking, HS would favor the raising candidate, generating the transparent feeding interaction as before.

13) Counterfeeding in SMR: iteration 2

| /gabur/ <*CC> | *CC | SM(*aCV, *CC) | DEP | *aCV | ID[LOW] |
|---|---|---|---|---|---|
| ☞ a. gabur <*CC> | | | | * | |
| b. gibur <*CC, *aCV> | | W* | | L | W* |

## 4 Comparison of Basic SMR and OT-CC Properties

SMR characterizes bleeding as an interaction involving simultaneous satisfaction of two markedness constraints and feeding as an illicit ordering in the satisfaction of two markedness constraints. SMR provides a single constraint type that unifies these types of transparent interactions as those that fail to realize a preferred serial order in the satisfaction of the given markedness constraints[2]. As already discussed, one advantage of SMR is that, unlike PREC constraints in OT-CC, SM constraints can operate on the fly during the gradual construction of derivations. This means that SMR, like HS, requires only one EVAL loop during which all constraints are present, obviating the need for separate chain construction and a second evaluation with an expanded set of constraints, as in OT-CC. It also means that, like in HS but unlike in OT-CC, derivations are evaluated as they are constructed, and therefore SM constraints cannot look ahead to evaluate the complete derivation: they exert their influence locally during the course of regular, gradual HS optimization. This section demonstrates how SMR's characterization of opacity in terms of markedness satisfaction makes this possible and shows that the focus on markedness makes it possible to profitably restrict some of the power of the serial ordering constraints.

Both of these consequences follow from the fact that the ordering requirements made by SM constraints are weaker than those made by PREC constraints. To review, an SM constraint, SM($M_1$, $M_2$), penalizes two configurations: <$M_1$+$M_2$> and < $M_2$, $M_1$>. The former is needed to penalize bleeding, while the latter is needed to penalize feeding. Crucially, violations are incurred only once both markedness constraints appear in *Mseq*: SM constraints cannot *require* that processes occur; they can only penalize certain combinations of processes already present in a candidate. This differs from the definition of OT-

[2] Unification is desirable to the extent that counterfeeding and counterbleeding are empirically connected. If the goal to unify counterfeeding and counterbleeding is abandoned, it is possible to weaken the requirements of the constraint formulation. Penalizing bleeding requires only that $M_1$, $M_2$ occur sequentially rather than simultaneously, while penalizing feeding requires that only one of $M_1$ and $M_2$ ever occur.

CC's Prec constraint, which has two clauses. Prec(F$_1$, F$_2$) is violated if F$_2$ occurs before F$_1$, in a configuration such as <F$_2$, F$_1$>, and it is also violated when F$_2$ is not preceded by F$_1$, in a configuration like <F$_2$> (McCarthy 2007: 98). The crucial difference is that Prec constraints penalize lone occurrences of F$_2$, which means that Prec constraints can require that some processes occur before others can and that, at least in principle, Prec constraints can block individual processes from occurring on their own.

The previous section showed that the weaker formulation of SM constraints is sufficient when counterbleeding and counterfeeding opacity are characterized in terms of gradual satisfaction of markedness. In contrast, the additional power of Prec constraints to penalize lone occurrences of F$_2$ is needed when interactions are characterized in terms of the gradual accumulation of faithfulness violations. It is the characterization of counterbleeding in particular that requires this stronger ordering condition. The previous section showed that counterbleeding in SMR is characterized by simultaneous satisfaction of both markedness constraints. In contrast, when counterbleeding is viewed from the perspective of faithfulness violation, it can only be ruled out by this stronger ordering condition, as can be seen in (14). The derivation corresponding to the transparent interaction (14c) incurs exactly one faithfulness violation, Max, whereas the counterbleeding interaction involves an earlier violation of Id[bk]. The only way to rule out the transparent interaction is to penalize the lone violation of Max not preceded by Id[bk].

14) Counterbleeding in OT-CC requires stronger ordering conditions (McCarthy 2007)

| /ħaːkimiːn/ | **ki* | **i*CV | Max | Prec(Id[bk], Max) | Id[bk] |
|---|---|---|---|---|---|
| a. ħaːkimiːn < > | W* | W* | L | | L |
| b. ħaːkimiːn, ħaːk$^{j}$imiːn < Id[bk] > | | W* | L | | * |
| ☼ c. ħaːkimiːn, ħaːkmiːn < Max > | | | * | W* | L |
| ☞ d. ħaːkimiːn, ħaːk$^{j}$imiːn, ħaːk$^{j}$miːn < Id[bk], Max > | | | * | | * |

**4.1** *OT-CC's Ranking Metaconstraint* McCarthy shows that it is necessary to constrain the activity of Prec constraints to prevent overgeneration of unattested interactions. His solution is a ranking metaconstraint – F$_2$ ≫ Prec(F$_1$, F$_2$) – that prohibits Prec constraints from being too high-ranked (McCarthy 2007: 99). The tableau in (15) illustrates the undesirable consequence of ranking Prec too high in a language that has both palatalization and deletion (adapted from McCarthy 2007: 101). The Prec(Id[bk], Max) constraint will rule out the transparent interaction of palatalization and deletion, favoring instead a counterbleeding interaction where palatalization precedes deletion (not shown). Thus, in contexts where deletion potentially interacts with palatalization, both deletion and palatalization apply: /ħaːkimiːn/ → ħaːk$^{j}$imiːn → [ħaːk$^{j}$miːn]. However, in contexts where only deletion is conditioned, Prec(Id[bk], Max) will block deletion (15b) because deletion involves a lone occurrence of Max not preceded by a violation of Id[bk]: /ʃarib-at/ → *[ʃarbat]. This is precisely the configuration observed in the bleeding candidate, except in this case, there is no independent motivation for palatalization. This interaction is bizarre since deletion happens in this hypothetical language, but only when it counterbleeds palatalization. As McCarthy explains, the ranking metaconstraint requires that Prec(Id[bk], Max) be ranked below Max, which rules out the possibility of such a pathological interaction since Prec(Id[bk], Max) can no longer affect whether or not Max is violated from that low-ranked position.

15) Unwanted effect of high-ranked Prec in OT-CC

| /ʃarib-at/ | **ki* | Prec(Id[bk], Max) | **i*CV | Max | Id[bk] |
|---|---|---|---|---|---|
| ☛ a. ʃaribat <> | | L | W* | L | |
| b. ʃarbat <Max> | | * | | * | |

Although the metaconstraint accomplishes its task, the need for such a metaconstraint suggests that the Prec constraints are inherently too powerful. Moreover, as demonstrated above, the characterization of counterbleeding in terms of faithfulness violations requires these strong ordering conditions. It is only when the interaction is viewed from the perspective of markedness satisfaction, as it is in SMR, that a weaker ordering formulation is made available. The tableau in (16) shows that SMR does not predict this pathological interaction even when SM constraints are high-ranked. Since SM constraints are vacuously satisfied when only one process has applied, they cannot block individual processes. SM constraints only become relevant once both markedness constraints have been satisfied and a potential interaction is at stake. This means that SMR makes it possible to more appropriately constrain the power of the constraints that evaluate derivations in a way that obviates the need for a ranking metaconstraint.

16) SM constraints can be high-ranked:

| /ʃarib-at/ | **ki* | SM(**ki*, **i*CV) | **i*CV | Max | Id[bk] |
|---|---|---|---|---|---|
| a. ʃaribat <> | | | W* | L | |
| ☞ b. ʃarbat <**i*CV > | | | | * | |

**4.2** *Why Prec Constraints + HS Won't Work* For related reasons, SM constraints can be evaluated during regular HS optimization while Prec constraints cannot. Consider a version of Prec that evaluates derivations as they are constructed and the attempt to analyze counterbleeding opacity in (17) (see Staubs 2011 for similar discussion). To favor the palatalizing candidate (17b) and block deletion on the first pass, Prec(Id[bk], Max) must be ranked above **i*CV. As discussed above, Prec accomplishes this by penalizing the deletion candidate (17c) for having a lone violation of Max. The problem is that the independent existence of deletion requires that **i*CV be ranked above Max, and by transitivity, Prec(Id[bk], Max) must therefore be ranked above Max, which violates the ranking metaconstraint[3]. As before, violating the ranking metaconstraint predicts the pathological interaction with deletion applying only when it counterbleeds an earlier instance of palatalization. The input /ʃarib-at/ will fail to undergo deletion, just as in (15) above. In fact, no ranking of these constraints achieves deletion in /ʃarib-at/ and palatalization followed by deletion in /ħaːkimiːn/. Thus, a version of HS with Prec constraints is a non-starter because it predicts only this pathological interaction and does not allow for the possibility of counterbleeding. Characterization of process interactions in terms of markedness satisfaction is crucial.

17) Counterbleeding in Prec+HS: iteration 1

| /ħaːkimiːn/ | **ki* | Prec(Id[bk], Max) | **i*CV | Max | Id[bk] |
|---|---|---|---|---|---|
| a. ħaːkimiːn <> | W* | | * | | L |
| ☞ b. ħaːk$^j$imiːn <Id[bk]> | | | * | | * |
| c. ħaːkmiːn <Max> | | W* | L | W* | L |

## 5 Process (Non-)Interaction

In OT-CC it is crucial that Prec constraints refer only to orderings among instances of faithfulness violations that arise as a consequence of processes that potentially interact. Otherwise, OT-CC would predict that the application or non-application of one process could affect the application or non-application of another process even when there is no potential for feeding or bleeding between the two processes. McCarthy proposes chain convergence in OT-CC as a way to define process interaction and constrain the application of Prec constraints to interacting processes. Chain convergence collapses derivations that result in the same output derived via the same faithfulness violations accrued in different orders and makes such spurious orderings invisible to Prec constraints (McCarthy 2007: 97). SM constraints must likewise be constrained to reference only markedness constraints responsible for interacting processes. Since chain convergence crucially references complete derivations, it is not available in HS as a way to restrict the application of SM constraints, which must be active during the creation of the derivation. Fortunately, a simple characterization of process interaction that builds on the characterization of feeding and bleeding developed in Section 2.2 is possible. This section illustrates the issue posed by process non-interaction and introduces the solution, which avoids the need to construct and subsequently collapse entire derivations.

[3] This ranking paradox does not arise in OT-CC because in that context Prec constraints evaluate entire derivations, and, once completed, the derivation /ħaːkimiːn/ → ħaːk$^j$imiːn → [ħaːk$^j$miːn] does not violate **i*CV. Consequently, in OT-CC Prec(Id[bk], Max) needs only dominate Id[bk] to penalize the deletion-only candidate.

**5.1** *The Issue: Blocking Unrelated Applications of a Process* The issue arises when two processes in a counterfeeding relationship are conditioned in unrelated loci. This is illustrated with a hypothetical input /gabubr/ which should undergo both epenthesis and raising. In an ordered rule analysis, raising would apply first to the underlying /a/, and then epenthesis would apply to the result: /gabubr/ → gibubr → [gibubur]. No counterfeeding occurs because raising is conditioned in the input. Counterfeeding only occurs when the two processes interact at a single locus, precisely in those cases where epenthesis would have created the input to raising had it applied earlier. This is not the case for /gabubr/.

However, without further refinement, SM constraints predict that raising will be blocked for inputs like /gabubr/, as shown in (18) – (19). Recall that the potential feeding interaction requires that *CC be ranked above *_a_CV for epenthesis to occur (shown in 4). Given this ranking, epenthesis always applies first, even for inputs where both epenthesis and raising are motivated. This is because satisfying higher-ranked *CC by epenthesizing (18b) is preferred to satisfying lower-ranked *_a_CV by raising (18c).

18) Pathological Blocking in Preliminary Version of SMR: Iteration 1

| /gabubr/ < > | *CC | SM(*_a_CV, *CC) | DEP | *_a_CV | ID[LOW] |
|---|---|---|---|---|---|
| a. gabubr <> | W* | | L | * | |
| ☞ b. gabubur <*CC> | | | * | * | |
| c. gibubr <*_a_CV > | W* | | L | L | W* |

Once epenthesis occurs, the *Mseq* recording the satisfaction of *CC is passed down to the next iteration, shown in (19). At this point raising should apply, but SM(*_a_CV, *CC) blocks raising (recall that SM(*_a_CV, *CC) must be ranked above *_a_CV to favor counterfeeding for input /gabr/). The resulting interaction is unattested: the application of epenthesis at the end of the word prevents the normal application of raising at an entirely unrelated locus at the beginning of the word.

19) Pathological Blocking in Preliminary Version of SMR: Iteration 2

| /gabubur/ <*CC> | *CC | SM(*_a_CV, *CC) | DEP | *_a_CV | ID[LOW] |
|---|---|---|---|---|---|
| ☛ a. gabubur <*CC> | | | | * | |
| b. gibubur <*CC, *_a_CV> | | W* | | L | W* |

**5.2** *A Solution: Tracking Interacting Markedness Constraints* Characterizing process interaction in terms of the effects processes have on markedness violations makes it possible to define and reference process interaction with a minor extension to the theory already laid out. The main insight behind the proposed solution is that when two processes interact for some input form, the markedness constraints involved affect each other's violation or satisfaction. Specifically, feeding interactions entail that the satisfaction of one markedness constraint coincides with the violation of another, while bleeding interactions entail that the satisfaction of one constraint coincides with the satisfaction of another on the same iteration. If the processes do not interact for a particular input, satisfaction of one of the markedness constraints will have no effect on the other one. The proposed extension of *Mseq* keeps track of these interactions and restricts SM constraints so that they refer to constraints that interact in this way.

The key observation is that violations of interacting markedness constraints change on the same iteration. The crucial difference between interacting and non-interacting raising and epenthesis is that interaction means application of one process affects the violation profile of both markedness constraints. In the interacting case (20), epenthesis (20b) creates a violation of the markedness constraint responsible for raising: *_a_CV, as shown in the shaded cells. In other words, the epenthetic candidate has satisfied *CC and introduced a new violation of *_a_CV which was not present in the faithful candidate. In contrast, in the non-interacting case (21), the epenthetic candidate (21b) satisfies *CC with no consequence for *_a_CV, which was already violated in the faithful candidate, as highlighted in the shaded cells.

20) Interacting Raising and Epenthesis

| /gabr/ < > | *CC | SM(*_a_CV, *CC) | DEP | *_a_CV | ID[LOW] |
|---|---|---|---|---|---|
| a. gabr <> | W* | | L | L | |
| ☞ b. gabur <*CC> | | | * | * | |

21) Non-interacting Raising and Epenthesis

| /gabubr/ < > | *CC | SM(*_a_CV, *CC) | DEP | *_a_CV | ID[LOW] |
|---|---|---|---|---|---|
| a. gabubr <> | W* | | L | * | |
| ☞ b. gabubur <*CC> | | | * | * | |
| c. gibubr <*_a_CV > | W* | | L | L | W* |

Thus, potential feeding interactions mean that satisfaction of one markedness constraint coincides with the violation of another on the same iteration. One operation (in this case epenthesis) simultaneously resolves one problem and causes another. Potential bleeding interactions are the opposite, as shown earlier in tableau (10) for input /ħaːkimiːn/: in potential bleeding interactions, one operation simultaneously resolves two problems. Recall that the SMR solution for counterbleeding opacity in /ħaːkimiːn/ crucially relied on the simultaneous satisfaction of two constraints, *_ki_ and *_i_CV, appearing in *Mseq* on the same iteration. When the same two processes are motivated for some input in non-interacting loci, satisfaction of one will have no effect on the violation profile of the other. For example, deletion of the medial /i/ in hypothetical /ħaːrikim/ would have no effect on the violation of *_ki_ in the last syllable.

One way to harness this information is to have *Mseq* track both satisfaction and violation of markedness constraints each iteration. In the tableaux that follow, '+' is used to indicate markedness constraints that have been newly violated as compared to the faithful candidate, and '–' is used to indicate markedness constraints that have been newly satisfied. The tableaux from (20) and (21) are repeated below showing the revision to *Mseq*, highlighting the interaction between *_a_CV and *CC that results from the epenthesis operation in (22) but not (23). In general, this notation makes it possible to formally define interaction between $M_A$ and $M_B$ as the occurrence of $\pm M_A$ together with $\pm M_B$ on some iteration in *Mseq*.

22) Interacting Raising and Epenthesis: iteration 1

| /gabr/ < > | *CC | SM(*_a_CV, *CC) | DEP | *_a_CV | ID[LOW] |
|---|---|---|---|---|---|
| a. gabr <> | W* | | L | L | |
| ☞ b. gabur <+*_a_CV–*CC> | | | * | * | |

23) Non-interacting Raising and Epenthesis: iteration 1

| /gabubr/ < > | *CC | SM(*_a_CV, *CC) | DEP | *_a_CV | ID[LOW] |
|---|---|---|---|---|---|
| a. gabubr <> | W* | | L | * | |
| ☞ b. gabubur <–*CC> | | | * | * | |
| c. gibubr <–*_a_CV> | W* | | L | L | W* |

In order for the extended notation to affect the modeling of process interactions, the SM constraints must be elaborated to reference this additional information, as formulated in (24). The revised definition of SM($M_1$, $M_2$) requires that $M_1$ strictly precede an interacting $M_2$ if both are present in *Mseq*.

24) **Serial Markedness** (Revised)
SM($M_1$, $M_2$): $-M_2$ must not precede or occur simultaneously with $-M_1$ *if $M_1$ and $M_2$ interact.*

With the extended notation, the SM constraints are appropriately restricted to operate over just those markedness constraints that interact. On the second iteration of the interacting case (25), the SM constraint penalizes the raising candidate for satisfying *_a_CV since the *Mseq* for this candidate reveals that *_a_CV and *CC did in fact coincide on a previous iteration. In contrast, no violation occurs in the non-interacting case (26) since, in that case, *_a_CV and *CC do not co-occur on any iteration in the epenthetic candidate's *Mseq*.

25) Interacting Raising and Epenthesis: iteration 2

| gabur <+*_a_CV–*CC> | *CC | SM(*_a_CV, *CC) | DEP | *_a_CV | ID[LOW] |
|---|---|---|---|---|---|
| ☞ a. gabur <+*_a_CV–*CC> | | | | * | |
| b. gibur <+*_a_CV–*CC, –*_a_CV> | | W* | | L | W* |

26) Non-interacting Raising and Epenthesis: iteration 2

| gabubur <–*CC> | *CC | SM(*aCV, *CC) | DEP | *aCV | ID[LOW] |
|---|---|---|---|---|---|
| a. gabubur <–*CC> | | | | W* | L |
| ☞ b. gibubur <–*CC, –*aCV > | | | | | * |

Crucially, the revised formulation can still detect counterbleeding since the two markedness constraints interact on the first iteration, as illustrated in (27).

27) Counterbleeding in SMR: iteration 1

| /ħaːkimiːn/ < > | *ki | SM(*ki, *iCV) | *iCV | MAX | ID[BK] |
|---|---|---|---|---|---|
| a. ħaːkimiːn < > | W* | | * | | L |
| ☞ b. ħaːkʲimiːn <–*ki> | | | * | | * |
| c. ħaːkmiːn <–*iCV–*ki> | | W* | L | W* | L |

## 6 Further Details: Tracking Loci

The notation introduced so far is sufficient for analyzing many basic process interactions, but when more complex cases are considered, it is important to be fully explicit about the contents of *Mseq* and to provide a notation to make such concreteness possible. In general, *Mseq* must track instances of changes in markedness violations together with indices to loci that make it possible to distinguish among multiple changes in the violation of a single markedness constraint. Consider a hypothetical input /gababr/ with multiple potential raising contexts, some interacting and others not. The first [a] should raise since it does not interact with the epenthesis in the final cluster, while raising of the second [a] should be counterfed by epenthesis. For SM constraints to treat these instances differently, they must be differentiated by locus.

Loci can be distinguished using the notation McCarthy (2007) proposes for OT-CC, indicating after each violation the first index in the string where the marked configuration matches. This notation is shown in (28)-(29). Since *CC ranks above *aCV, epenthesis occurs on the first iteration. The fact that epenthesis creates a violation of *aCV in the second [a] is recorded as +*aCV@4 in the *Mseq* for candidate (28b). As shown in (29), the SM constraint can correctly block raising in the interacting context (29c) on all subsequent iterations, without affecting raising in the non-interacting case (29b).

28) Illustration of loci indices: iteration 1

| /gababr/ < > | *CC | SM(*aCV, *CC) | DEP | *aCV | ID[LOW] |
|---|---|---|---|---|---|
| a. gababr <> | W* | | L | L* | |
| ☞ b. gababur <–*CC@5+*aCV@4 > | | | * | ** | |
| c. gibabr <–*aCV@2> | W* | | L | L | W* |
| d. gabibr <> | W* | | L | L* | W* |

29) Illustration of loci indices: iteration 2

| /gababur/ <–*CC@5+*aCV@4 > | *CC | SM(*aCV, *CC) | DEP | *aCV | ID[LOW] |
|---|---|---|---|---|---|
| a. gababur <–*CC@5+*aCV@4> | | | | W** | L |
| ☞ b. gibabur <–*CC@5+*aCV@4, –*aCV@2> | | | | * | * |
| c. gabibur <–*CC@5+*aCV@4, –*aCV@4> | | W* | | * | * |

## 7 Discussion

SMR is a minimal extension of HS that takes advantage of this framework's inherent serial character to capture opaque interactions. The paper introduces a novel characterization of feeding and bleeding interaction in terms of changes in markedness violations. A novel constraint type, Serial Markedness, makes reference to this general characterization of process interaction and unifies the treatment of counterbleeding and counterfeeding opacity, penalizing transparent interactions for their failure to satisfy markedness constraints in the preferred derivational order.

Focusing on markedness satisfaction allows the ordering requirements made by SM constraints to be weaker than those of PREC constraints in OT-CC, which has significant consequences. As shown in Section 4.1, the weaker formulation reins in some of the excess power of PREC constraints, obviating the need for a ranking metaconstraint. In addition, the focus on markedness satisfaction makes available the utilization of SM constraints within standard HS optimization, as shown in Section 4.2. Although SMR endows HS with an ability to track aspects of the derivational history, this ability is restricted: SMR does not have access to the original input, to intermediate representations, nor does it have access to the completed derivation. In all these respects, SMR is true to the gradual, myopic spirit of HS. Finally, the focus on markedness satisfaction makes possible a novel and simple formulation of process interaction that avoids the need to construct entire derivations and subsequently collapse them. In sum, SMR significantly simplifies the machinery necessary to model counterfeeding and counterbleeding opacity as compared to OT-CC.

This paper has focused primarily on architectural differences between SMR and OT-CC, arguing that SMR is a more economical theory. However, SMR has broader theoretical and empirical implications for process interaction, which cannot be explored in depth here due to space limitations. One such difference between SMR and OT-CC is the formulation of process interaction, which has implications for global rule interactions such as counterfeeding from the past (CFFTP; Wilson 2006, Wolf 2010). CFFTP is an interaction that goes beyond the power of ordered rules but is predicted to be possible by OT-CC. In this interaction, a process that is nonadjacent in the derivation to a later process blocks the later process from applying. SMR cannot predict an opaque interaction between processes that are ordered far apart in the derivation because process interaction is defined locally: in SMR, processes that do not affect each other's markedness violations *on a single iteration* cannot interact opaquely. The existence of real cases of CFFTP is controversial (though see Wolf 2010 for possible examples), which provides some support for SMR's more restrictive predictions and motivates further investigations into the nature of global rule interactions and the alternative perspectives offered by SMR and OT-CC.

Another prediction of SMR is that phonological processes' placement in a derivation should be based on the markedness constraints that drive them: if multiple processes are driven by a single markedness constraint, SMR predicts these processes should pattern together in terms of their placement in the derivation. This partially derives from HS: the relative ranking of markedness constraints determines the order in which they will be satisfied (when they do not interact). SMR extends this general prediction to interacting processes: the SM constraints enforce a specified derivational order between all processes driven by the specified markedness constraints. In contrast, OT-CC predicts that processes violating the same faithfulness constraint should behave as though they occur at the same point in the derivation. Overall, new insights about locality and serialism offered by recent work in HS motivate a further look at these and other consequences of Serial Markedness Reduction for opacity and phonology more generally.

## References

Elfner, Emily. to appear. Stress-epenthesis interactions in Harmonic Serialism. In *Harmonic Grammar and Harmonic Serialism*, ed. John McCarthy and Joe Pater. London: Equinox Press.

Kiparsky, Paul. 1973. Abstractness, opacity, and global rules. In *Three Dimensions of Linguistic Theory*, ed. Osamu Fujimura, 57-86. Tokyo: TEC.

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

McCarthy, John J. 2007. *Hidden generalizations: phonological opacity in Optimality Theory*. London: Equinox Press.

McCarthy, John J. 2008a. The gradual path to cluster simplification. *Phonology* 25(2). 271-319.

McCarthy, John J. 2008b. The serial interaction of stress and syncope. *Natural Language & Linguistic Theory* 26. 499-546.

McCarthy, John J. 2009. Harmony in harmonic serialism. Ms, University of Massachusetts, Amherst.

Pruitt, Kathryn. 2010. Serialism and locality in constraint-based metrical parsing. *Phonology* 27(03). 481-526.

Staubs, Robert. 2011. Chains and Targets: Opacity in Harmonic Serialism. Ms, University of Massachusetts, Amherst.

Wilson, Colin. 2006. Counterfeeding from the past. Ms, the Johns Hopkins University, Baltimore, MD.

Wolf, Matthew. 2010. On the existence of counterfeeding from the past. Talk presented at 84th Linguistic Society of America Meeting, Baltimore.