# Pathological Effects of Local Disjunction*

Charlie O'Hara
*University of Southern California*

## 1 Overview

When working in Optimality Theory (Prince & Smolensky, 1993/2004; McCarthy & Prince, 1995), a balance must be found when positing constraints between those that are not powerful enough to explain attested forms of opacity, and those that are too powerful and predict problematic and pathological languages. Constraint connectives, which create new constraints whose violations are defined relative to those of two simple constraints, can be a simple and intuitive way to extend the power of OT to capture opacity. Local Conjunction (Smolensky, 1993; Moreton & Smolensky, 2002; Itô & Mester, 2003) is the best known connective, but since its inception considerable restraint has been placed upon it to prevent it from predicting pathological effects (Itô & Mester, 1996; Łubowicz, 2005; Pater, 2009).

Yet, as explored in Wolf (2007), Local Conjunction is just one of 16 possible constraint connectives. Along with *and*, some have posited *or*, as we do in natural language and logic. I call this connective Local *Dis*junction.[1] Local Disjunction was first posited by Hewitt & Crowhurst (1996), and explored further in later work by Crowhurst & Hewitt (1997); Crowhurst (2011); Downing (2000, 1998). In this paper, I argue that Local Disjunction creates pathological effects, and even restriction like that which we have applied to Local Conjunction cannot save it.

First in section 2, I discuss the constraints placed on Local Conjunction, namely domain restrictions and markedness-faithfulness type restrictions, and define Local Disjunction, with an eye towards using similar restrictions to those used in Local Conjunction. Before showing the pathologies, I walk through the inner mechanics of Local Disjunction and explain its unique effects in section 3. Then I show that markedness-markedness disjunction creates the same sorts of pathologies as markedness-faithfulness conjunction in 4. Finally, in section 5, I explore how faithfulness-faithfulness disjunction, the last permutation of disjunction, creates strange patterns.

## 2 Constraint Connectives

**2.1** *Restricting Local Conjunction* Local Conjunction (Smolensky, 1993; Moreton & Smolensky, 2002; Itô & Mester, 2003) in its simplest form is defined as such:

(1) An input output pair /$x$/-[$y$] incurs violations of the constraint P&$_D$Q for each domain $D$ such that both constraints P and Q assign a violation mark to /$x$/-[$y$] within $D$.

P&Q is only violated if both P and Q are violated within the domain $D$. Thus, Local Conjunction punishes only *the worst of the worst*, the candidate that violates both constraints (Smolensky, 2006).

(2) *Local Conjunction*

| /$x$/ | P | Q | P&Q |
|---|---|---|---|
| a. best of the best | | | |
| b. violates $Q$ | | * | |
| c. violates $P$ | * | | |
| d. worst of the worst | * | * | * |

* This paper would not have been possible without helpful comments from Karen Jesney and Reed Blaylock, and audiences at USC PhonLunch, SoCaSiL 2013, and Phonology 2014, particularly Eric Baković, Ricardo Bermúdez-Otero, Dustin Bowers, René Kager and Bruce Hayes.

[1] Following Tesar (2014), I call this connective local disjunction rather than the now confusing boolean conjunction.


Proceedings of *AMP 2014*
Completed February 27, 2015

Unconstrained constraint connectives can create a legion of pathological effects. Specifically here, allowing too large of a domain for Local Conjunction can create strange long distance or counting effects, and allowing any combination of markedness and faithfulness constraints creates languages where contrasts are only preserved in marked positions. Thus, the definition in (1) has been greatly restricted to avoid these pathologies. Łubowicz (2005) offers the following definition for restricted Local Conjunction:

(3) *Restricted Local Conjunction: (taken from Łubowicz (2005))*
C=$C_1$&$C_2$ is violated iff

1. $\text{LOC}_{C_1} \cap \text{LOC}_{C_2} \neq \emptyset$
2. $C_1$ results in $C_2$ if $C_1$ is Faithfulness and $C_2$ is Markedness

Local Conjunction has been restricted both in the domain upon which it searches for violations, and in what types of constraints it can connect. By expanding the domain of Local Conjunction from the locus of violations of the constraints, we can create constraints with unlikely and unpredicted effects (Łubowicz, 2005; Kawahara, 2006). For example, by setting the domain to the *Stem* level and a self-conjunction between a constraint and itself, we can create constraints that ban a certain (larger than one) number of violations of that constraint within the stem. This can create languages that allow a marked structure in any position, as long as that structure appears nowhere else in the stem (see 5) by conjoining two markedness constraints. Here (and throughout this paper) I use the constraint *VTV, which drives intervocalic voicing. I define it as in (4)

(4) *VTV
Assign a violation mark for each voiceless segment in an intervocalic position.

(5) With high ranking $*\text{VTV}\&_{Stem}*\text{VTV}$:
$[\text{pateka}]_{Stem}$, $[\text{padeka}]_{Stem}$, $[\text{patega}]_{Stem}$, $*[\text{padega}]_{Stem}$

By combining two faithfulness constraints, a language can be formed where a faithfulness constraint can be violated once, but not twice within a stem (see 6), which can create a similar pattern. This can create a weird type of chain-shift like behavior where each input form can repair just one of its violations of *VTV, so even though an input with a *VTV violation would not surface faithfully, a different input would surface as it.

(6) With ID(VOICE)$\&_{Stem}$ID(VOICE)≫*VTV≫ID(VOICE)

| /patega/ | ID(VOICE)$\&_{Stem}$ID(VOICE) | *VTV | ID(VOICE) |
|---|---|---|---|
| ☞ a. pateka | | | * |
| b. patega | | *W | L |
| /pateka/ | ID(VOICE)$\&_{Stem}$ID(VOICE) | *VTV | ID(VOICE) |
| ☞ c. patega | | * | * |
| d. padega | *W | L | **W |
| e. pateka | | **W | L |

This effect is not dependent on the conjoined constraints being the same, and in fact seems even weirder if the conjoined constraints are less related. The constraint *VTV $\&_{Stem}$*MIDV predicts a language that allows mid-vowels ([tebadagak]) and intervocalic voiceless obstruents ([tabadakak]), but not both within the same stem, no matter the distance between them. (*[tebadakak]) Languages of this type are not attested.

Yet domain isn't the only thing that cause pathologies in Local Conjunction. Conjunctions of one markedness and one faithfulness constraint are able to create pathological effects called *markedness reversals* (Itô & Mester, 1996, 1998). A markedness reversal is where a marked structure is banned except in marked positions. However, typologically, contrasts are supposed to be best maintained in unmarked positions, leading to our theories of positional privilege (Beckman, 1998). In markedness reversals, contrast is best maintained in marked positions. These can be caused even if the conjunction domain is restricted to the degree shown in (3).

Itô & Mester (1996) give the example of the constraint IDENT(VOICE)&NOCODA. If this constraint ranks above *VOICEOBS≫IDENT[VOICE], we see that onset voice obstruents must be devoiced, but coda voiced obstruents cannot be devoiced because the locus of violation of IDENT[VOICE] (the coda voiced obstruent) is part of the locus of violation for NOCODA (the coda), thus violating the conjunction.

(7) *Markedness Reversal*

| /bad/ | ID(VOICE)&NOCODA | *VOICEOBS | ID(VOICE) |
|---|---|---|---|
| a. pat | *W | L | **W |
| ☞ b. pad | | * | * |
| c. bad | | **W | L |
| d. bat | *W | * | * |

Candidate (7-a) shows devoicing in the onset and the coda, but this candidate loses because the [t] violates ID(VOICE) and violates NOCODA, thus violating the conjunction. Candidate (7-c) shows no loss of voice contrasts in any position, but loses because it violates *VOICEOBS more than Candidate (7-b). We can't devoice coda obstruents in this language, because it would violate the conjunction, but we must devoice all other obstruents.

These markedness reversals can be created when a markedness (8) and a faithfulness (9) constraint are conjoined, but do not occur if two constraints of the same type are conjoined. To avoid predicting these phenomena, we need to restrict M&F conjunctions more extremely than we restrict M&M or F&F. Therefore, when evaluating Local Disjunction we may restrict M∨F to a greater degree than M∨M or F∨F as well.

(8) A *markedness constraint* is a constraint $M$ such that for any given output $y$, assigns the same violation marks to all input-output pairs /$x$/-[$y$].

(9) A *faithfulness constraint* is a constraint $F$ such that it assigns no violation marks to the input-output pair /$x$/-[$x$], for all $x$.

**2.2** *Local Disjunction* If we were to define Local Disjunction in the least restricted way we could as follows:

(10) An input output pair /$x$/-[$y$] incurs violations of the constraint P$\vee_D$Q for each domain $D$ such that at least one of the constraints, P or Q, assign a violation mark to /$x$/-[$y$] within $D$.

Crucially, while P&Q is violated only if both P and Q are violated (11), P∨Q (P or Q) is violated if one or both are violated (12). P∨Q punishes the worst of the worst candidates, and both of the bad candidates that violate just Q or just P. Local Disjunction instead favors only *the best of the best* candidates, the one that violates neither constraint.

(11) *Local Conjunction*

| /$x$/ | P | Q | P&Q |
|---|---|---|---|
| a. best of the best | | | |
| b. violates $Q$ | | * | |
| c. violates $P$ | * | | |
| d. worst of the worst | * | * | * |

(12) *Local Disjunction*

| /$x$/ | P | Q | P∨Q |
|---|---|---|---|
| a. best of the best | | | |
| b. violates $Q$ | | * | * |
| c. violates $P$ | * | | * |
| d. worst of the worst | * | * | * |

Large domain Local Disjunction can create similar issues to large domain Local Conjunction. In order to avoid domain related issues, I restrict the domain of Local Disjunction. It's less intuitive to constrain the domain for Local Disjunction than for Local Conjunction. In (3), Łubowicz (2005) defined Local Conjunction as violated when the locus of violation of one constraint intersected with that of the other constraint. The intersection aspect of this is necessary, since the locus of violation of the two conjoined constraints may not be the same. Some constraints are violated on a segmental level, some are violated on a syllable level, etc..

Now we need to also include those loci of violation of just one of the constraints without the other, since those violate Local Disjunction as well.

(13) *Domain-Restricted Local Disjunction*
Incur a violation mark of P∨Q for each locus of violation of P that intersects with a locus of violation of Q, each locus of violation of P that does not intersect with a locus of violation of Q, and each locus of violation of Q that does not intersect with a locus of violation of P.

If P&Q represents a set-theoretic intersection of constraints, only interested in the places where both constraints are violated, P∨Q represents a union of the constraints, looking where one or both are violated. In set-theory, it's well known that the number of items in the union of P and Q is equal to the number of items in P plus the number of items in Q that aren't in P.

(14) $|P \cup Q| = |P - Q| + |Q - P| + |P \cap Q|$

M&F Local Conjunction was also restricted more harshly than M&M or F&F conjunction. It will be important for our argument to consider each of these permutations for disjunction. M&F conjunction is the most troublesome of the conjunctions, so it is not surprising to see that M∨F is easily identifiable as theoretically problematic. Wolf (2007) shows that a disjunction between a markedness and a faithfulness constraint results in a constraint that is neither, breaking a standard assumption about OT and opening the theory up to infinite and circular chain-shifts (Moreton, 1999).

Using the definitions seen in (8) and (9) we can see that markedness constraints do not reference the input at all, and only care about the output; and faithfulness constraints assign no violations to an output that matches its input. Using the arbitrary example of *g∨IDENT(VOICE), we can see that markedness-or-faithfulness constraints (M∨F) are neither markedness nor faithfulness constraints themselves.

(15) *Local Disjunction is neither M nor F*

| /gi/ | *g∨ID(VOICE) | *g | ID(VOICE) |
|---|---|---|---|
| a. gi | * | * | |
| b. ki | * | | * |
| /ki/ | *g∨ID(VOICE) | *g | ID(VOICE) |
| c. ki | | | |
| d. gi | * | * | * |

If *g∨ID(VOICE) were a markedness constraint, /$x$/-[ki] would need to receive the same violations for any input $x$. /gi/-[ki] receives one violation because IDENT(VOICE) is violated by the g, but /ki/-[ki] receives none, since no segments change their voicing and [ki] has no [g]. Since some violations of M∨F are just violations of F, M∨F must make reference to the input.

If *g∨ID(VOICE) were a faithfulness constraint, /$x$/-[$x$] would need to incur 0 violations for all $x$. We can see that /gi/-[gi] violates this constraint once, since it violates *g. Since some inputs will violate M, their faithful mappings will also violate M∨F. Thus M∨F cannot be a faithfulness constraint either.

It's commonly assumed that we should restrict CON to containing only markedness or faithfulness constraints. This assumption protects us from predicting circular and infinite chainshifts, which are unattested and seem computationally unviable, (Moreton, 1999).[2] So we should not allow M∨F disjunctions into our CON. If we are to allow Local Disjunction, we will need to restrict it so that different MF-type disjunctions are fully banned. We have precedent to do this because we restricted different MF-type conjunction more harshly than other conjunctions. But how do M∨M and F∨F disjunctions fare?

## 3 How Local Disjunction works

Before looking at M∨M and F∨F disjunction in particular, we should understand how exactly Local Disjunction extends the set of possible grammars. Local Conjunction has been said to punish *the worst of the worst* (Smolensky, 2006), in that it can ban candidates that violate two constraints, but not those that violate a single of the conjoined constraints. In contrast, Local Disjunction *favors the best of the best*.

If no conflicting constraints are higher ranking, the disjunction chooses the candidate that violates neither of the conjoined constraints. This is the same effect seen if P dominates Q, and Q dominates all the constraints P∨Q dominated.

[2] However, in order to create these problematic effects, a constraint must favor a marked and unfaithful candidate over the marked and faithful candidate (Wolf, p.c.). M∨F does not do this, so fear of these chain shifts is not enough to rule out M∨F. For the time being, I assume we restrict CON to markedness and faithfulness constraints, since M∨F disjunction has not been attempted in the past. The most obvious effect of M∨F disjunction would be favoring the candidates that are faithful but are not marked, which I worry would create markedness reversal-like effects.

(16) *LD favors the best of the best*

| /x/ | P∨Q |
|---|---|
| ☞ a. best of the best | |
| b. violates $Q$ | * |
| c. violates $P$ | * |
| d. worst of the worst | * |

(17) *Same as high ranking P and Q*

| /x/ | P | Q |
|---|---|---|
| ☞ a. best of the best | | |
| b. violates $Q$ | | * |
| c. violates $P$ | * | |
| d. worst of the worst | * | * |

However, if some higher ranked constraint rules out the best of the best candidates, P∨Q acts differently than P directly dominating Q. In (18), we can see that C rules out the best of the best candidates, and thus all the other candidates tie on P∨Q. On the other hand, in (19), the candidate that violates Q but not P wins because P and Q must be treated as independent constraints.

(18) *LD cannot break a tie*

| /x/ | C | P∨Q |
|---|---|---|
| a. best of the best | *! | |
| ☞ b. violates $Q$ | | * |
| ☞ c. violates $P$ | | * |
| ☞ d. worst of the worst | | * |

(19) *Candidate b wins*

| /x/ | C | P | Q |
|---|---|---|---|
| a. best of the best | *! | | |
| ☞ b. violates $Q$ | | | * |
| c. violates $P$ | | *! | |
| d. worst of the worst | | *! | * |

Even with P∨Q, some ranking must exist further down in the constraint ranking that says P≫Q.[3] Often, this will result in being no different than the situation without the disjunction. But, when we have some constraint C' that intervenes between P∨Q and P that favors candidate c or d to b, we can create a pattern where C' is only active when either P or Q must be violated.

There are two types of systems that can be caused depending on how C' incurs violations; the *worst of the worst system*, or the *irrelevant choice system*. If C' (or C) prefers the worst of the worst candidate to the one that just violates $P$, the language prefers the best of the best candidates, but failing that defaults to the worst of the worst candidate. In the worst of the worst system shown in (20), the best of the best wins whenever C is not violated by the best of the best candidate. Yet, when the best of the best candidate is ruled out by C, P∨Q no longer chooses between the candidates, and allows C' to select the worst of the worst candidate. Neither "somewhat bad" candidate can surface. Some worst of the worst systems can be modeled other ways, but Local Disjunction allows you to create *any* worst of the worst system.

(20) *Worst of the worst system*

| /x/ | C | P∨Q | C' | P | Q |
|---|---|---|---|---|---|
| a. best of the best | *W | | | L | |
| b. violates $Q$ | | * | *W | L | * |
| c. violates $P$ | | * | *W | * | L |
| ☞ d. worst of the worst | | * | | * | * |

In irrelevant choice systems, C' just prefers the candidate that violates P to that which violates Q. Here, when C rules out the best of the best candidate, we move down to C' which selects the candidate that violates just P (21-c). This can appear just like a language where Q≫P, unless there are some contexts where C' does not assign violation marks to any candidates, in which case, (21-b) wins instead. Then it is apparent that P≫Q. The "irrelevant choice" is that the choice of whether to satisfy P or Q is not based on either of those constraints at all, but by C'. Compounding the complexity, like the worst of the worst system, if C does not ban the best of the best candidate, the best of the best candidate wins. Again, systems that look like this may be possible without using Local Disjunction, but Local Disjunction uniquely predicts an infinite variety of them.

3 This can be said without loss of generality, because if Q≫P, we can just put Q on the left side of the disjunction, since disjunction is a commutative operation.

(21) *Irrelevant choice system*

| /$x$/ | C | P∨Q | C' | P | Q |
|---|---|---|---|---|---|
| a. best of the best | *W | | | L | |
| b. violates $Q$ | | * | *W | L | *W |
| ☞ c. violates $P$ | | * | | * | |
| d. worst of the worst | | * | | * | *W |

The following is an example of a worst of the worst system. Imagine a language with the constraint ID(CONT)∨ID(VOICE). Usually this constraint protects sounds in the language from spirantizing or voicing.

(22) *Voiceless obstruents are protected usually*

| /pa/ | ID(CONT)∨ID(VOICE) |
|---|---|
| a. pa | |
| b. ba | *W |
| c. ɸa | *W |
| d. βa | *W |

If we let a higher markedness constraint *VTV, which marks intervocalic voiceless obstruents, the disjunction fails to decide the winner. Normally, the language would just choose to voice the /p/, since ID(CONT) dominates ID(VOICE).

(23) **VTV forces at least one of the* ID *constraints to be violated*

| /apa/ | *VTV | ID(CONT)∨ID(VOICE) | ID(CONT) | ID(VOICE) |
|---|---|---|---|---|
| a. apa | *W | L | | L |
| ☞ b. aba | | * | | * |
| c. aɸa | *W | * | *W | L |
| d. aβa | | * | *W | * |

But if we throw in an intervening constraint *VDV, which marks all intervocalic stops regardless of voicing, we can force this language to both spirantize and voice intervocalically.

(24) **VDV causes us to choose the worst of the worst candidates*

| /apa/ | *VTV | ID(CONT)∨ID(VOICE) | *VDV | ID(CONT) | ID(VOICE) |
|---|---|---|---|---|---|
| a. apa | *W | L | *W | L | L |
| b. aba | | * | *W | L | * |
| c. aɸa | *W | * | | * | L |
| ☞ d. aβa | | * | | * | * |

This example is a worst of the worst system like that in (20), with our general C=*VTV, C'=*VDV, P=ID(CONT) and Q=ID(VOICE). With a different C and C', we can imagine a language with an irrelevant choice system, where the disjunction allows us to pick the candidate that violates Q in certain contexts, like in (21). In the coming sections I will show how this type of constraint interaction can create pathological patterns with M∨M and F∨F.

## 4 M∨M creates markedness reversals

M∨M disjunction can create languages where unmarked and doubly marked things exist, but no intermediates. These are worst of the worst systems, like that in (20).[4] This can easily create pathologies because this allows us to produce only marked things in marked positions.

In (25), we see a language that allows voiced obstruents only in coda. With NOCODA∨*VOICEOBS ≫ ID(VOICE), voiced obstruents are usually repaired by devoicing, as with input /data/. Candidate (25-a) cannot win because it violates the disjunction, which ranks above ID(VOICE). However, with all other faithfulness

4 It is more difficult to create the irrelevant choice system like that in (21), but it should be possible. Since worst of the worst systems are often pathological with markedness constraints, I see no reason to explore irrelevant choice systems.

constraints high ranking (crucially DEP), if we posit an input like /pad/, the devoicing candidate (25-e) fails because devoicing still does not repair the violation of NOCODA. The best of the best candidate here is (25-f), but it cannot be reached because DEP outranks the disjunction. Thus, the worst of the worst candidates, that maintains the /d/ in coda, wins because it beats the devoicing candidate at ID(VOICE).

(25) *Markedness Reversal*

| /data/ | DEP | NOCODA∨*VOICEOBS | ID(VOICE) | NOCODA | *VOICEOBS |
|---|---|---|---|---|---|
| a. da.ta | | *W | L | | *W |
| ☞ b. ta.ta | | | * | | |
| /pad/ | DEP | NOCODA∨*VOICEOBS | ID(VOICE) | NOCODA | *VOICEOBS |
| ☞ c. pad | | * | | * | * |
| d. pa.di | *W | * | | L | * |
| e. pat | | * | *W | L | L |
| f. pa.ti | *W | L | *W | L | L |

However, as shown in (26), codas in general are not problematic, because the constraints that would lead to a coda consonant leaving the coda (DEP or MAX) are ranked higher than any constraint that incurs violations for coda position consonants.

(26) *Codas are fine*

| /tat/ | DEP | NOCODA∨*VOICEOBS | ID(VOICE) | NOCODA | *VOICEOBS |
|---|---|---|---|---|---|
| a. ta.ti | *W | L | | L | |
| ☞ b. tat | | * | | * | |

This language maintains full voicing contrasts in syllable codas, but loses this contrast in onsets. Note that this is exactly the same pattern as the markedness reversal language predicted by M&F disjunction in (7) by Itô & Mester (1996). We get a marked segment, in this case voiced obstruents, only in a marked position, codas. Faithfulness should be less respected in marked positions not in privileged positions.

This type of markedness reversal is possible with any two markedness constraints, as long as one marks a position and the other marks a structure.

(27) *Markedness reversals predicted by Local Disjunction*

| Constraint | Pathology predicted |
|---|---|
| *VTV∨*VELFRIC≫ID(VOICE) | The only voiceless obstruent allowed intervocalically is [x] |
| *VCDCODA∨PARSE-$\sigma$≫ID(VOICE) | Voiced obstruent codas are only allowed in unparsed syllables. |
| *NASFRIC∨*DORSAL≫ID(PLACE) | The only dorsal segments allowed are [x̃] and [ɣ̃] |

These pathologies show that we cannot allow M∨M in CON (at least without substantial restriction). Thus, so far we have shown that M∨F and M∨M cannot be allowed unbounded in CON. The case for disjunction already looks weak, and we will see in the coming section that F∨F disjunction may create some problems of its own.

## 5 F∨F Effects

When two faithfulness constraints are disjoined, the disjunction prefers the best of the faithful candidates; the one that violates neither constraint. Yet, failing that, the candidate that violates both, that changes twice, fares no worse on the disjunction than the candidates that violate just one of the constraints.

This can create two types of effects:

- **Phonologically Derived Environment Effects (PDEEs)**: If we already need to violate one of the disjoined constraints in order to avoid violating a high ranking markedness constraint, we can freely violate the other in order to avoid a structure marked by an intervening constraint, even though non-derived structures marked by that constraint would be protected by the disjunct.
- **Choice of Repair**: If avoiding violating our top constraint can be done by violating either of our disjoined constraints in *different* segments, the choice of which constraint we violate is usually chosen by the lower relative ranking of the constraints. However, a constraint that intervenes between them can

cause the grammar to choose the other repair to avoid some marked segment, which appears normally elsewhere in the language.

**5.1** *PDEEs* In serial frameworks, PDEEs are phenomenon that occur only to phonologically derived material. For example, /VbV/→[VbV], but /VpV/→VbV→[VβV]. Underived VbV is fine, but derived VbV is subject to lenition. Local Disjunction has the ability to predict almost any possible PDEE. If a PDEE is defined as two phenomena, one modeled by $M_1 \gg F_1$, and one with $M_2 \gg F_2$; the ranking $M_1 \gg F_1 \vee F_2 \gg M_2 \gg F_1, F_2$, creates a PDEE. In this case $M_2$ only repairs through $F_2$, if you got to $M_2$ through violating $F_1$.

(28) *VTV ($M_1$)≫ ID(VOICE)($F_1$) creates /VpV/→VbV
V[-CONT]V($M_2$) ≫ ID(CONT) ($F_2$) creates VbV→[VβV]
ID(CONT)∨ID(VOICE) ≫ *V[-CONT]V prevents /VbV/→[VβV]

(29) *Phonologically Derived Environment Effect*

| /aba/ | *VTV | ID(CONT) ∨ ID(VOICE) | *V[-CONT]V | *β |
|---|---|---|---|---|
| a. aβa | | *W | L | *W |
| b. apa | *W | *W | * | |
| ☞ c. aba | | | * | |
| /apa/ | *VTV | ID(CONT) ∨ ID(VOICE) | *V[-CONT]V | *β |
| ☞ d. aβa | | * | | * |
| e. apa | *W | L | | L |
| f. aba | | * | *W | L |

In (29), /aba/-[aba] is safe, because it does not violate *VTV, and since it is the best of the best candidates, the disjunction selects it. However, /apa/-[apa] does violate *VTV, so we must move past the disjunction. Then, *V[-cont]V (which marks intervocalic stops, regardless of voicing), rules out all but the worst of the worst candidates, (29-d), /aβa/.

PDEEs have been observed in several languages, but there are questions whether these can be readily learned (White, 2013). The ease at which they can be predicted with Local Disjunction seems at odds with their low typological distribution, but this could possibly be only a learning issue. Thus, the brunt of my argument against F∨F disjunction is based on the choice of repairs process.

**5.2** *Choice of repairs* F∨F disjunction can force a language's choice of repairs to depend on seemingly irrelevant material. If violating a high ranking markedness constraint forces us to violate at least one of the disjoined faithfulness constraints, but does offer us the choice, the choice can be made based on avoiding some marked structure that appears freely throughout the rest of the language.

For example, violations of the constraint *VTV can be avoided by voicing the obstruent, or deleting a vowel. Ranking all faithfulness constraints but ID(VOICE) and MAX above *VTV prevents any other repair method. If we use the disjunction ID(VOICE)∨MAX, and rank MAX over ID(VOICE), voicing is our default repair. However, if PARSE-$\sigma$ intervenes between the disjunction and ID(VOICE), the vowel deletes if it benefits footing (assuming high ranking foot-structure constraints that restrict feet to being two syllables).

(30) PARSE-$\sigma$ *is active if *VTV is violated by the best of the best*

| /bata/ | *VTV | ID(VOICE)∨ MAX | PARSE-$\sigma$ | MAX | ID(VOICE) |
|---|---|---|---|---|---|
| ☞ a. (ba.da) | | * | | | * |
| b. bad | | * | *W | *W | * |
| c. (ba.ta) | *W | L | | | L |
| d. bat | | * | *W | *W | L |
| /badata/ | *VTV | ID(VOICE)∨ MAX | PARSE-$\sigma$ | MAX | ID(VOICE) |
| ☞ e. (ba.dat) | | * | | * | |
| f. (ba.da).da | | * | *W | L | *W |
| g. (ba.da).ta | *W | L | *W | L | |
| h. (ba.dad) | | * | | * | *W |

In this tableau, even-voweled inputs like /bata/ cannot surface faithfully because [t] is intervocalic. Thus, the disjunction cannot favor the best of the best candidate (the fully faithful one) so it doesn't help. Then, because odd-syllabled output strings cannot be fully parsed, voicing is chosen as the repair (30-a) rather than vowel deletion (30-c or d). However, an odd-voweled input like /badata/ deletes the vowel. In order to avoid violating PARSE-$\sigma$ it becomes even syllabled (30-e). Once the *VTV environment is repaired, there is no reason to also perform the other repair (30-d or h).

Here *VTV drives syncope. If the best of the best candidate of an odd voweled input like /badada/, that violates neither ID(VOICE) and MAX, does not violate *VTV, syncope cannot occur. This form in (-a) violates neither *VTV nor the disjunction, protecting it from syncope. Underlying forms with an odd number of syllables and no intervocalic voiceless obstruents will surface with unparsed syllables.

(31) *Syncope is caused only when interacting with *VTV*

| /badada/ | *VTV | ID(VOICE)∨MAX | PARSE-$\sigma$ | MAX | ID(VOICE) |
|---|---|---|---|---|---|
| ☞ a. (ba.da).da | | | * | | |
| b. (ba.dad) | | *W | L | *W | |
| c. ba.da.da | | | ***W | | |

Note that while PARSE-$\sigma$ is not active for deletion when there are no input VTVs, it does still act to force us to foot the word as much as we can. This language does not look particularly strange on the surface. VTV clusters are never seen, and considering that, odd syllable words of all sorts appear on the surface. However, we could see some alternations that involve this intervocalic voicelessness driven syncope. Vowels in *VTV positions are susceptible to syncope, no matter which syllable in the word they fall (even if the vowel might be stressed when it does appear). These patterns are not attested.

(32) *Examples of *VTV conditioned syncope*

| | | | | | |
|---|---|---|---|---|---|
| /pata/ | → | [(pa.da)] | /pata-na/ | → | [(pat.na)] |
| /patada/ | → | [(pat.da)] | /patada-na/ | → | [(pa.da).(da.na)] |
| /ataba/ | → | [(ta.ba)] | /ataba-na/ | → | [(a.da).(ba.na)] |
| /ata/ | → | [(a.da)] | /ata-na/ | → | [(ta.na)] |

This sort of effect can happen anywhere where the high ranking markedness constraint's violations can be avoided by violating any two different faithfulness constraints, but it looks weirdest when the loci of violation of the faithfulness constraints are not the same. This is possible for any contextual markedness constraint, if you choose one faithfulness constraint that changes the marked structure, and one that changes the context. We do see choices of repairs in human language, but they tend to be based on position, not avoiding some other markedness constraint, that otherwise is not relevant.

## 6 Conclusion

In the end, we have observed that Local Disjunction creates a number of patterns that we do not expect to see. While we considered restricting out the M∨F disjunction that Wolf (2007) argued should be banned from CON, in order to find some fruit from the same type disjunctions, the fruit were rotten. M∨M disjunction can create markedness reversals similar to those created by M&F conjunction. F∨F disjunction allows a grammar to choose repairs based on relatively irrelevant markedness constraints. These effects create a large amount of difficulty, whereas the problems Local Disjunction solves are relatively small. If one of these categories did not have problematic effects, we could have restricted disjunction to applying to the others, but all three create problems in relatively simple constructions. Each of the pathological constructions we saw above is generalizable, and for most useful seeming disjunctions, a pathological effect can be created. These problems would occur no matter how we define the domain of Local Disjunction. Thus, we cannot admit Local Disjunction into our CON.

## References

Beckman, Jill N. (1998). *Positional Faithfulness*. Ph.D. thesis, University of Massachusetts Amherst, Amherst.

Crowhurst, Megan & Mark Hewitt (1997). Boolean operations and constraint interactions in optimality theory. Ms. University of North Carolina & Brandeis University [ROA-229].

Crowhurst, Megan J. (2011). *Constraint Conjunction*, Blackwell Publishing, vol. IV of *The Blackwell Companion to Phonology*, chap. 62.

Downing, Laura J. (1998). On the prosodic misalignment of onsetless syllables. *Natural Language & Linguistic Theory* 16, 1–52.

Downing, Laura J. (2000). Morphological and prosodic constraints on kinande verbal reduplication. *Phonology* 17, 1–38.

Hewitt, Mark S. & Megan J. Crowhurst (1996). Conjunctive constraints and templates in optimality theory. *NELS 26*, 101–116.

Itô, Junko & Armin Mester (1996). Rendaku 1: Constraint conjunction and the OCP. Paper presented at the Kobe Phonology Forum.

Itô, Junko & Armin Mester (1998). Markedness and word structure: OCP effects in Japanese. Ms. University of California, Santa Cruz [ROA-255].

Itô, Junko & Armin Mester (2003). *Japanese morphophonemics: markedness and word structure*. MIT Press, Cambridge, MA.

Kawahara, Shigeto (2006). A faithfulness ranking projected from a perceptibility scale: The case of [+voice] in japanese. *Language* 82, 536–574.

Łubowicz, Anna (2005). Locality of conjunction. Alderete, John, Chung-hye Han & Alexei Kochetov (eds.), *Proceedings of the Twenty-Fourth West Coast Conference on Formal Linguistics*, Cascadilla Proceedings Project, Somerville, MA, 254–262. [ROA-764].

McCarthy, John J. & Alan Prince (1995). Faithfulness and reduplicative identity. *University of Massachusetts Occasional Papers* 18, 249–384.

Moreton, Elliott (1999). Non-computable functions in optimality theory. Ms. University of Massachusetts, Amherst [ROA-667].

Moreton, Elliott & Paul Smolensky (2002). Typological consequences of local constraint conjunction. Mikkelsen, L. & C. Potts (eds.), *WCCFL 21 Proceedings*, Cascadilla Press, Cambridge, MA.

Pater, Joe (2009). Review of Smolensky and Legendre (2006). The Harmonic Mind. *Phonology* 26, 217–226.

Prince, Alan & Paul Smolensky (1993/2004). *Optimality Theory: Constraint Interaction in Generative Grammar*. Blackwell, Oxford.

Smolensky, Paul (1993). Harmony, markedness, and phonological activity. Rutgers Optimality Workshop I, New Brunswick, NJ. ROA-87.

Smolensky, Paul (2006). *Optimality in Phonology II: Harmonic Completeness, Local Constraint Conjunction and Feature-Domain Markedness*, MIT Press, vol. 2: Linguistic and Philosophical Implications, chap. 14, 586–720.

Tesar, Bruce (2014). *Output-Driven Phonology: Theory and Learning*. Cambridge University Press, Cambridge.

White, James (2013). *Bias in phonological learning: evidence from saltation*. Ph.D. thesis, UCLA.

Wolf, Matthew (2007). What constraint connectives should be permitted in ot? *University of Massachusetts Occasional Papers in Linguistics* 36, 151–179.

Wolf, Matthew (p.c.). Correspondence.