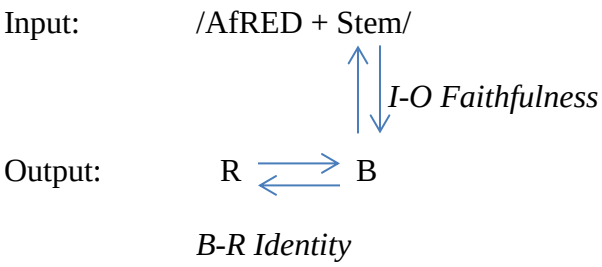
BOUNDARY-PROXIMITY Constraints in Order-Disrupting Reduplication

1. Introduction

I propose an analysis of thorny patterns of reduplication in the unrelated languages Saisiyat (Austronesian: Taiwan) and Pima (Uto-Aztecan: Arizona) using two related constraints, BOUNDARY-PROXIMITY/SYLLABLE and BOUNDARY-PROXIMITY/SEGMENT (abbreviated throughout as BPROX/SYLL and BPROX/SEG). In these patterns, the reduplicated portions are minimal and atemplatic; in addition , the reduplication is order-disrupting, i.e., the input ordering of segments is disrupted (not faithfully preserved) in the output. BPROX constraints provide an elegant account for both the size and the position of the reduplicated portions.

I couch my analysis in the theoretical framework of Optimality Theory (OT: Prince and Smolensky 1993, 2004), and I specifically use the Base-Reduplicant Correspondence (BR-Correspondence) account of reduplication (McCarthy and Prince 1995). I use their Basic Model with only IO- and BR-Correspondence, not the Full Model with IR-Correspondence as well. The Basic Model is simpler than and thus superior in terms of parsimony to the Full Model; IR-Correspondence moreover introduces empirical and theoretical liabilities (Idsardi and Raimy 1997, Struijke 2002, Inkelas and Zoll 2005). In the Basic Model, the Reduplicant can only get its content from the Base, not the input.

1) Basic Model (McCarthy and Prince 1995)



Typically accounts of minimal reduplication in the BR-Correspondence framework require economy constraints, such as the *STRUC family (Zoll 1994), ALL-SYLL-L/R (Mester and Padgett 1994) or MARKEDNESS (Gafos 1998), to prevent satisfaction of BR-MAX, which prefers maximal copying. Several accounts have been given of minimal reduplication using economy constraints, including Spaelti (1997), Gafos (1998), and Walker (2000). However, economy constraints come with both theoretical and empirical liabilities (Gouskova 2003), because instead of only penalizing marked structure, they penalize any structure at all. Gouskova argues that valid markedness constraints must evaluate marked structures only in contrast with unmarked structures. An economy constraint such as *SYLL considers all non-null structure to be marked; as Gouskova shows, this kind of constraint predicts unattested patterns of economy.

I instead propose an account of the Saisiyat and Pima reduplicative patterns using the concept of proximity between corresponding segments (Odden 1994, Suzuki 1998, Rose 2000, Zuraw 2002, Nelson 2003, Rose and Walker 2004, Lunden 2004, Kennedy 2005). Specifically, I build on Kennedy's (2005) constraint PROXIMITY, which demands that corresponding segments be as close as possible to one another by penalizing material that intervenes between the corresponding segments (Kennedy 2005). Kennedy uses PROXIMITY to account for Marantz's Generalization (Marantz 1982) that reduplication copies from the closest edge inward; i.e., prefixal reduplicants copy from left to right, whereas suffixal reduplicants copy from right to left. PROXIMITY also accounts for the minimal size of the reduplicant because copying more segments moves each pair of copies further away from each other. For instance, as (3) copies more material than (2), the copies in (3) are farther away from (2), so PROXIMITY favors the smaller reduplicant (2) over the larger one (3). More specifically, in (2), only one segment [a] intervenes between the

corresponding instances of [p], while in (3), there are three intervening segments [ati]. Thus, PROXIMITY assigns one violation mark to (2), and three violation marks to (3).

2) /RED-patik/ -> [**pa**-patik] Material between copies of [p]: 1 segment [a]

3) /RED-patik/ -> [**pati**-patik] Material between copies of [p]: 3 segments [ati]

I argue that Kennedy's version of PROXIMITY should be redefined in two ways, as a constraint I call BOUNDARY-PROXIMITY (BPROX). First, instead of evaluating every segment in the Reduplicant for its closeness to its Base correspondent, BPROX only evaluates edge elements of the Reduplicant and their Base correspondents. Second, instead of counting the segments between each corresponding pair, BPROX penalizes phonological boundaries that intervene between the correspondents. Penalizing boundaries between units rather than the units themselves avoids a pathology encountered by the segment-counting definition.

I argue for two specific phonological units which BPROX is sensitive to: the segment and the syllable. Both subconstraints, BPROX/SEG and BPROX/SYLL, demand that corresponding edge elements be close to one another, but they can differ as to how they evaluate specific forms. For instance, the corresponding copies of [p] in (4) are in adjacent syllables, with only one syllable boundary (.) and two segment boundaries (_) intervening, while the copies in (5) are two syllables apart, with two intervening syllable boundaries and four intervening segment boundaries. Thus, both BPROX/SYLL and BPROX/SEG favor the smaller reduplicant (4) over the larger one (5), just as does the segment-counting version (cf. (2-3) above).

4) /RED-patik/ -> [**p_a_**-.pa.tik] 2 segment boundaries, 1 syllable boundary

   a. Violations of BPROX/SEG: 2

   b. Violations of BPROX/SYLL: 1

5) /RED-patik/ -> [**p_a_.t_i_**-.pa.tik] 4 segment boundaries, 2 syllable boundaries

   a. Violations of BPROX/SEG: 4

   b. Violations of BPROX/SYLL: 2

However, BPROX/SYLL and BPROX/SEG can differ in how they evaluate different forms. For a candidate to be unmarked with respect to BPROX/SYLL, the corresponding elements must be in the same syllable, as in (6).

6) /RED-patik/ -> [**p**_-a_p.tik] 2 segment boundaries, 0 syllable boundary

   a. Violations of BPROX/SEG: 2

   b. Violations of BPROX/SYLL: 0

BPROX/SYLL thus favors (6) over (4) (no violations vs. one violation), while BPROX/SEG evaluates both (4) and (6) equally (two violations). However, in (6), the input sequence /patik/ is not faithfully preserved in the output, i.e., the order of [patik] is disrupted by the second occurrence of [p], so constraints governing segment order and position, e.g., LINEARITY and CONTIGUITY, are violated. In (4) [pa-patik], however, the sequence [patik] appears undisrupted, satisfying both LINEARITY and CONTIGUITY. BPROX/SYLL thus uniquely can cause segments to shift their position to satisfy it, leading to the appearance of either Reduplicant infixation (7), violating CONTIGUITY, or Base metathesis (8), violating LINEARITY.

7) $[pa]_B[\mathbf{p}]_R.[tik]_B$ Reduplicant [p] splits Base [patik]

8) $[\mathbf{p}]_R[aptik]_B$ Base [a] and [p] switch their input sequence /pa/

I analyze a case of such order-disruption in both Saisiyat and Pima using BPROX constraints. I argue that order-disrupting reduplication can be given a simple account using these constraints. Thus BPROX not only can motivate minimal and atemplatic reduplication, but also can force

material to rearrange in reduplication. These latter effects cannot be accounted for with economy constraints.

1.1 Data: Patterns of Reduplication

I illustrate my proposal with patterns of reduplication from two unrelated languages. I analyze two patterns of reduplication in Saisiyat: Ca- reduplication and Progressive reduplication (Zeitoun and Wu 2005; Zeitoun and Wu forthcoming; personal fieldwork). In Ca- reduplication, the first consonant of the root is copied along with the fixed segment [a], which together prefix to the root (9-12). Ca- reduplication indicates several meanings, including instrument nouns (9), future instrumental focus verbs (10), reciprocals (11), and plural adjectives and stative verbs (12).

9) [ħi.la] 'sunshine' -> [$ħ_1$a-.$ħ_1$i.la] 'sun'

10) [bø.tøʔ] 'tie' -> [b_1a-.b_1ø.tøʔ] 'will be used to tie'

11) [ʂək.laʔ] 'know' -> [$ʂ_1$a-.$ʂ_1$ək.laʔ] 'know each other'

12) [lo.man] 'naughty' -> [l_1a-.l_1o.man] 'naughty (pl.)'

Progressive reduplication occurs with the Agent Focus infix [om]/[əm]/[øm], and, unlike Ca- reduplication, only indicates Progressive aspect. In Progressive reduplication, only the first consonant of the root is copied, and the second copy of this consonant is infixed into the Agent Focus infix, separating the vowel of the infix from the consonant [m], and occupying the coda of the first syllable (13-15).

13) [k-o.m-i.taʔ] 'see-AF' -> [k_1-o-k_1-.m-i.taʔ] 'be seeing-AF'

14) [r-ə.m-ə.mə] 'dye-AF' -> [r_1-ə-r_1-.m-əmə] 'be dyeing-AF'

15) [ħ-ø.m-a.ŋiħ] 'cry-AF' -> [$ħ_1$-ø-$ħ_1$-.m-a.ŋiħ] 'be crying-AF'

However, if there is no room in the coda of the first syllable, since neither complex onsets nor codas are allowed in Saisiyat (17), the vowel of the infix is copied as well, so that the Reduplicant is a CV sequence (16).

16) [ʂ-om-.βət] ‘beat-AF’ -> [$ʂ_1o_2$-.$ʂ_1$-o_2m-.βət] ‘be beating-AF’

17) *[$ʂ_1$-o-$ʂ_1$-.m-βət], *[$ʂ_1$-o-$ʂ_1$-m-.βət]

When only the single consonant is copied (13-15), both the infix ([om], [øm], [əm]) and the root ([kitaʔ], [ħaŋiħ], [rəmə]) are split up into non-contiguous portions ([o...m], [ø...m], [ə...m]; [k...itaʔ], [ħ...aŋiħ], [r...əmə]). Any account of this pattern must provide a motivation for this non-contiguity; in other words, it must eliminate a competing candidate respecting CONTIGUITY, such as (18), where the infix [om] and root [kitaʔ] are not split.

18) *[k_1-om-.k_1i.taʔ]

I also analyze a similar pattern of reduplication in the unrelated language Pima, which also features minimal, order-disrupting reduplicants and violations of CONTIGUITY or LINEARITY. In Pima plural reduplication, typically only a single initial consonant is copied (Riggle 2004, 2006). The second copy of this consonant surfaces in the coda of the first syllable in the word (19-21).

19) [ma.vit] ‘lion (sg.)’ -> [mam.vit] ‘lion (pl.)’

20) [si.puk] ‘cardinal (sg.)’ -> [sis.puk] ‘cardinal (pl.)’

21) [kuʃ.va] ‘lower skull (sg.)’ -> [kukʃ.va] ‘lower skull (pl.)’

However, if the resulting coda would be ill-formed (25), the first vowel is also copied, so that the reduplicant is a CV sequence (22-24).

22) [ho.dai] ‘rock (sg.)’ -> [ho.ho.dai] ‘rock (pl.)’

23) [ɲu.maɡʃ] ‘liver (sg.)’ -> [ɲu.ɲu.maɡʃ] ‘liver (pl.)’

24) [biʃp] ‘horse collar (sg.)’ -> [bi.biʃp] ‘horse collar (pl.)’

25) *[hoh.dai], *[ɲuɲ.maɡʃ], *[bibʃp]

When only the single consonant is copied (19-21), the underlying position of elements (as shown by the singular forms) is changed: the second copy of the consonant ([m], [s], [k]) interrupts the underlying sequence of the root ([mavit], [sipuk], [kuʃva]). Any account of this pattern must motivate this interruption of the underlying sequence.

1.2 Generalizations

In these reduplicative patterns, the Base is not copied maximally into the Reduplicant. In most of these patterns, moreover, the underlying sequence of segments is not preserved faithfully in the reduplicated form. In the Saisiyat and Pima patterns, only the first consonant of the root is reduplicated; in this sense these reduplications are minimal. Usually failure to copy the Base maximally is due to templatic forces, such that the Reduplicant must satisfy certain prosodic conditions modeled by general templatic constraints (cf. Generalized Template Theory, e.g. McCarthy and Prince 1995, Urbanczyk 1996 and 2006). However, a single consonant is not a prosodic unit and templatic constraints cannot refer to it: in this sense these reduplications are atemplatic. Economy constraints that favor less material over more material are often used to account for minimality in reduplication without using templatic constraints (e.g., Spaelti 1997, Gafos 1998, and Walker 2000).

While economy constraints do prefer the least amount of material possible to be copied, they do not account for the failure to faithfully preserve input sequences of segments. In both the Saisiyat and Pima patterns, the internal correspondent of the copied consonant occupies the coda of the

first syllable whenever possible. Only when this would create a phonotactically illegal output is the first CV of root copied instead. Therefore, the basic, general pattern is the order-disrupting one: in other words, the order-disrupting pattern is the elsewhere case. I schematize the reduplicative patterns below.

Table 2. Generalized Schematization of Reduplicative Patterns in Saisiyat and Pima

Pattern	Schema	Examples
Saisiyat Ca-Reduplication	C_1... -> C_1a-.C_1...	ħ$_1$i.la -> ħ$_1$a-.ħ$_1$i.la
Saisiyat Progressive Reduplication	$C_1V_1.C_2$... -> $C_1V_1C_1.C_2$...	k$_1$-o.m-i.taʔ -> k$_1$-o-k$_1$-.m-i.taʔ
	$C_1V_1C_2.C_3$... -> $C_1V_1.C_1V_1C_2.C_3$... *$C_1V_1C_1C_2C_3$...	ʂ$_1$-o$_1$m-.βət -> ʂ$_1$o$_1$-.ʂ$_1$-o$_1$m-.βət *ʂ$_1$-o$_1$-ʂ$_1$-m-βət
Pima Plural Reduplication	$C_1V_1.C_2$... -> $C_1V_1C_1.C_2$...	m$_1$a.vit -> m$_1$am$_1$.vit
	$C_1V_1.C_2$... -> $C_1V_1.C_1V_1.C_2$... *$C_1V_1C_1.C_2$...	h$_1$o$_1$.dai -> h$_1$o$_1$.h$_1$o$_1$.dai *h$_1$oh$_1$.dai

1.3 Overview of Paper

In this paper I propose a mechanism, BOUNDARY-PROXIMITY (BPROX), which can account for these patterns in an elegant and straightforward manner. In Section 2, I build on Kennedy's (2005) constraint PROXIMITY, which penalizes segments that intervene between correspondents. I redefine this constraint as BPROX, which penalizes phonological boundaries that intervene between edges of the Reduplicant and their correspondents in the Base. This constraint encourages a minimum of corresponding elements, so that the edge elements in correspondence

have a minimum of intervening segmental or syllable boundaries, the number of which is often increased by copying more material from the Base. In Sections 3 and 4, I show how BPROX/SYLL accounts for the position of the internal correspondents in Pima and Saisiyat and forces discontiguity or metathesis, depending on the parsing of the reduplicative form. BPROX/SEG is also necessary to account for some further subpatterns in Pima reduplication. In Section 5, I compare the BPROX analysis to other possible analyses, including reduplicant infixation (Riggle 2006) and Base-Reduplicant Adjacency (Lunden 2004). I show that Riggle's (2006) analysis, while empirically adequate, is less parsimonious than the BPROX analysis. While Lunden's (2004) analysis is quite similar to the BPROX analysis, it cannot account for all of the data in Saisiyat and Pima reduplication.

2. Proposal: BOUNDARY-PROXIMITY

I propose that the patterns of reduplication in Saisiyat and Pima, and order-disrupting reduplication in general, can be accounted for with BOUNDARY-PROXIMITY constraints that demand that edge elements in the Reduplicant be as close as possible to their corresponding elements in the Base. More specifically, BPROX demands that no phonological boundaries of a certain type (e.g., syllable boundaries) intervene between these elements. This constraint is a reformulation of the constraint PROXIMITY used in Kennedy (2005) to account for anchoring effects; this reformulation is necessary to avoid a pathology, namely, the prediction of unattested patterns of reduplication.

2.1 PROXIMITY: Kennedy (2005)

Several OT accounts have been proposed regulating the distance between corresponding segments (Odden 1994, Suzuki 1998, Rose 2000, Zuraw 2002, Nelson 2003, Rose and Walker

2004, Lunden 2004). The constraint PROXIMITY is introduced in Kennedy (2005) to capture the effects of Marantz's Generalization (Marantz 1982), which states that reduplicants copy adjacent material, i.e., prefixes copy from the left edge (26), while suffixes copy from the right edge (27).

26) /RED-patik/ -> [pa-patik], *[ik-patik]

27) /patik-RED/ -> [patik-ik], *[patik-pa]

Marantz's Generalization is typically captured in OT by BR-ANCHOR constraints, which demand that the left or right edges of the Base and the Reduplicant be in correspondence (e.g., McCarthy and Prince 1995, Kager 1999). Kennedy shows that BR-ANCHOR constraints both overgenerate and undergenerate attested patterns of reduplication. For example, ranking BR-ANCHOR-RIGHT over BR-ANCHOR-LEFT when a Reduplicant is left-aligned (i.e., a prefix) causes opposite-edge anchoring: the right edges of the Reduplicant and the Base must be in correspondence, so the Reduplicant copies from the wrong edge of the Base (e.g., [ik-patik] above). Such opposite-edge anchored reduplication is not attested: in the cases where Marantz's Generalization is violated, the next closest available anchoring site is chosen. Opposite edge-anchoring, as in the hypothetical reduplication in (28), does not occur, even when a higher-ranked constraint prohibits same-edge copying.

28) /RED-patik/ -> [at-patik], *[ik-patik]

Kennedy shows that BR-ANCHOR constraints cannot account for these minimal violations of Marantz's Generalization, such as in Chuukese, where suffixing reduplication does not copy from the right edge (a vowel) but from the segment preceding it (a consonant), e.g., in (29), due to a general process of word-final vowel deletion (Kennedy 2005, from Goodenough and Sugita 1980).

29) /seniŋe-RED/ -> [seniŋe-niŋ]; *[seniŋe-ŋe], *[seniŋe-sen]

Since BR-ANCHOR-RIGHT, which is categorical, is violated by the winner [seniŋe-niŋ] (because the right edge of the Reduplicant [ŋ] is not in correspondence with the right edge of the Base [e]), it cannot beat the challenger [seniŋe-sen], which also violates BR-ANCHOR-RIGHT but satisfies BR-ANCHOR-LEFT, which must still be present in the grammar even if low-ranked. Thus categorial ANCHORING cannot account for the attested minimal violations of Marantz's generalization, but rather has a sour-grapes effect: if the adjacent edge itself cannot be copied (as in Chuukese), then the opposite edge must be copied, not the next closest site to adjacent edge, as in the tableau below (from Kennedy 2005; the constraint FREE-VOWEL penalizes word-final vowels).

Table 3. Underprediction: Mis-anchoring (Kennedy 2005)

/seniŋe-RED/	FREE-VOWEL	ANCHOR-RIGHT	ANCHOR-LEFT
seniŋe-ŋe	*!		*
seniŋe-niŋ		*	*!
:(seniŋe-sen		*	

To rectify these problems, Kennedy proposes a new constraint, PROXIMITY, that generates the attested patterns where Marantz's Generalization is minimally violated for specific reasons, but does not overgenerate unattested violations. Kennedy uses the following definition of the constraint PROXIMITY (30).

30) PROXIMITY: no material intervenes between segments in correspondence. (Kennedy 2005)

PROXIMITY is only relevant to correspondence between elements of the same output, which includes BR-Correspondence. PROXIMITY ideally demands that these correspondents be adjacent;

however, constraints on phonotactics and reduplicant size may cause it to be violated. Kennedy's version of PROXIMITY is crucially a gradient constraint, so that even if correspondents cannot be adjacent, they must be as close as possible in order to violate it minimally. (31-32) below show how PROXIMITY can account for Marantz's generalization: (31), which conforms to this generalization violates PROXIMITY less than (32), which does not.

31) p a - p a t i k 1 segment [a] intervenes between copies of [p]
1

32) i k - p a t i k 4 segments [kpat] intervene between copies of [i]
4

Kennedy's analysis evaluates violations of PROXIMITY in segments: the number of violation marks is however many segments intervene between each pair of correspondents. According to Kennedy, because PROXIMITY "measure distances between every pair of correspondents ... longer reduplicants therefore incur more violations." Thus if more material is copied into the Reduplicant, the distance between each segment in the Reduplicant and its Base correspondent increases (33).

33) p a t i - p a t i k 3 segments [ati] intervene between copies of [p]
3

In fact, the violations increase exponentially: as more segments are copied, each of these segments is farther away from its correspondent, so that the number of violations is the product of the number of copied segments and the number of intervening segments, as shown in (34-35).

34) [pa-patik] Violations of PROXIMITY: 2(copies)×1(intervening) = 2

35) [pati-patik] Violations of PROXIMITY: 4(copies)×3(intervening) = 12

Nothing seems to be gained from having exponentially increasing violations. In fact, Kennedy only ever counts the violations of PROXIMITY incurred by one pair of correspondents in his analysis, so there is no need to evaluate PROXIMITY for every pair of segments.

2.1.1 PROXIMITY Pathology

I now show that these features of PROXIMITY, counting violations gradiently by the segments themselves and counting up the violations of each pair of segments, predict unattested patterns of reduplication. Assuming BR-MAX is high-ranked enough to compel total copying of the Base, PROXIMITY would favor splitting the Base and the Reduplicant into pieces so that the correspondents are closer together. The fewer segments there are that separate the corresponding pairs, the less PROXIMITY will be violated. Splitting the Base and the Reduplicant into smaller portions, each of whose parts are in correspondence, will prevent the segments in the other portions from intervening between the corresponding portions, so that each portion has fewer violations of PROXIMITY. Ideally, PROXIMITY would encourage each of these portions to be as small as possible, i.e., a segment; in this limiting case, there would be no non-corresponding segments between the portions, and PROXIMITY would be maximally satisfied.

For example, PROXIMITY would prefer the completely split output form of /RED-mafe/ in (36) to the fully contiguous output in (37). In (36), no segments intervene between any of the copies in correspondence, because all the corresponding elements are adjacent to one another. Therefore (36) completely satisfies PROXIMITY. In (37), on the other hand, each pair of correspondents is separated by three segments, e.g., the copies of [m] are separated by [afe]. Because all four of the corresponding pairs are separated by three segments, and because PROXIMITY counts up the

individual violations of each pair of corresponding segments, then there are twelve (four times three) total violations of PROXIMITY.

36) $[m]_R$-$[m]_B$-$[a]_R$-$[a]_B$-$[f]_R$-$[f]_B$-$[e]_R$-$[e]_B$ 4(copies)×0(intervening) = 0 violations

37) [m a f e]$_R$ - [m a f e]$_B$ 4(copies)×3(intervening) = 12 violations

3 3 3 3

Therefore PROXIMITY favors the split candidate (36) over the contiguous candidate (37). If PROXIMITY is ranked above O-CONTIGUITY (McCarthy and Prince 1994, 1995), which penalizes such splitting, then the split candidate in (36) will beat the contiguous candidate in (37).

38) O-CONTIGUITY: The portion of S_2 (e.g., the Base corresponding to the input root) standing in correspondence forms a contiguous string (McCarthy and Prince 1995)

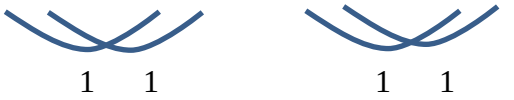
Table 4. PROXIMITY Pathology

/RED-mafe/	PROXIMITY	O-CONTIGUITY
$[mafe]_R$-$[mafe]_B$	12!	
:($[m]_R$-$[m]_B$-$[a]_R$-$[a]_B$-$[f]_R$-$[f]_B$-$[e]_R$-$[e]_B$		4

Such completely split reduplication is never attested, but it is predicted by PROXIMITY. The fully split candidate in (36), [mmaaffee], is almost universally disfavored on phonotactic grounds. However, split candidates with syllable-sized portions, such as (39), do obey phonotactic constraints, and still violate PROXIMITY much less than contiguous candidates such as (37). In (39), each segment is separated from its correspondent by only one other segment, e.g., the copies of [m] are separated by the intervening segment [a]. Since all four corresponding pairs are

each separated by one intervening segment, there are four (four times one) total violations of PROXIMITY in (39).

39) [m a]$_R$ - [m a]$_B$ - [f e]$_R$ - [f e]$_B$ 4(copies)×1(intervening) = 4 violations



Therefore the syllable-size splitting candidate (39) still beats the fully contiguous candidate (37) when PROXIMITY outranks O-CONTIGUITY.

Table 5. PROXIMITY Pathology (continued)

/RED-mafe/	PROXIMITY	O-CONTIGUITY
[mafe]$_R$-[mafe]$_B$	12!	
:([ma]$_R$-[ma]$_B$-[fe]$_R$-[fe]$_B$	4	2

This type of reduplication is attested, though rarely (Mandarin Adjective AABB reduplication does this, though probably for other reasons; see Walker and Feng 2004). However, PROXIMITY as defined in Kennedy (2005) predicts that Reduplicants and Base will be split up as much as phonotactically possible to get correspondents as close as possible to each other. Both to prevent unattested types of splitting and to predict the small range of attested splitting types, a new definition of PROXIMITY is needed that alters how it counts violations.

2.2 Formalizing BOUNDARY-PROXIMITY

To avoid this pathology, I propose a redefinition of PROXIMITY that I call BOUNDARY-PROXIMITY, which counts phonological boundaries rather than units that intervene between elements in correspondence. Moreover, BOUNDARY-PROXIMITY only evaluates the edges of Reduplicants, not every segment in the Reduplicant. Together, these two changes can avoid the pathology.

40) BOUNDARY-PROXIMITY (BPROX): Assign a violation mark for every phonological boundary that intervenes between an edge element in the Reduplicant and its correspondent in the Base.

To show how this constraint prevents the above pathology, I first use the version of BOUNDARY-PROXIMITY sensitive to segmental boundaries and the left edges of the Reduplicant, which I call BOUNDARY-PROXIMITY/LEFT,SEGMENT.

41) BOUNDARY-PROXIMITY/LEFT,SEGMENT (BPROX/L,SEG): Assign a violation mark for every segmental boundary that intervenes between a left edge element in the Reduplicant and its correspondent in the Base.

BPROX/L,SEG assigns violation marks to intervening segmental boundaries, not to intervening segments themselves. I define a segmental boundary as the meeting place between two segments, e.g., there is one segmental boundary between the [m] and the [a] in the sequence [ma]. In the fully split candidate, repeated below in (42), the corresponding pairs are each separated by one segmental boundary (represented by "_"), e.g., the copies of, e.g., [m] have one boundary between them. Since the Reduplicant is fully split in (42), each segment of the Reduplicant is a left edge. Crucially, I define the left edge not as the single leftmost segment in the Reduplicant, but as each left edge (i.e., element of a Reduplicant portion that has no Reduplicant material immediately to its left). Therefore, each Reduplicant segment in (42) is evaluated for violation of BProx/L,Seg; since there are four left-edge segments, and each segment has one segmental boundary intervening between it and its Base correspondent, there are four (four times one) violations of BProx/L,Seg in total.

42) [m]_[m][a]_[a][f]_[f][e]_[e] 4(edges)×1(boundary) = 4 violations

43) [m_a_f_e]_[mafe] 1(edge)×4(boundaries) = 4 violations

In the fully contiguous candidate, repeated above in (43), the corresponding pairs are each separated by four segmental boundaries, e.g., the copies of [m] have four boundaries ($m_{\_1}a_{\_2}f_{\_3}e_{\_4}m$) between them. However, there is only one left-edge segment, [m], in this Reduplicant, so its four intervening boundaries are the only ones that violate BPROX/L,SEG, which is therefore is violated four times. Since both (42) and (43) tie on BPROX/L,SEG, the low-ranking constraint O-CONTIGUITY can then eliminate the Reduplicant-splitting candidate.

Table 6. Proximity Pathology Prevented

/RED-mafe/	BPROX/L,SEG	O-CONTIGUITY
-> [mafe]-[mafe]	4	
[m]-[m]-[a]-[a]-[f]-[f]-[e]-[e]	4	4!

The two crucial changes in the definition of BPROX together avert the pathology. The first change is that rather than all corresponding elements needing to be close to each other, only left or right edges of the Reduplicant and their correspondents need to be. Because I define these edges as the termini of any contiguous portion of the Reduplicant, split Reduplicants have more edges than fully contiguous ones, and thus the potential for more violations of BPROX. Therefore, nothing is gained by splitting the Reduplicant: even though the correspondents are closer together, more of them are evaluated by BPROX.

As noted above, Kennedy's analysis only concentrates on how much one pair of corresponding elements violates PROXIMITY; having all the pairs' violations add up is never necessary to his account of Marantz's Generalization. I specify that only that edgemost elements of the Reduplicant are important, not those of the Base, so as not to run into the overgeneration

problems that Kennedy points out in the BR-ANCHOR account. For example, in the Chuukese example above (repeated from (29) as (44)), the right edge of the Base [e] does not have a correspondent in the Reduplicant, while the right edge of the Reduplicant [ŋ] does of course have a Base correspondent.

44) /seniŋe-RED/ -> [seniŋe-niŋ]; *[seniŋe-ŋe], *[seniŋe-sen]

In minimal violations of Marantz's Generalization, the outer Reduplicant edge [ŋ] is not in correspondence with the edge of the Base it affixes to [e], but rather to an element as close as possible to that edge ([ŋ], which is left-adjacent to [e]). When a higher-ranked constraint (here the prohibit on word-final vowels) prevents the Reduplicant edge from being in correspondence with the nearest Base edge (which would violate BPROX the least), then the Reduplicant edge should correspond to the closest Base element that the higher-ranked constraint allows, thus violating BPROX as little as possible. In this way, Marantz's Generalization can only ever be violated minimally, as even when edge-in association is not permitted, the association of elements is still as close as possible to the affixed edge, as in the Chuukese pattern (44).

Table 7. Minimal Violation of Marantz's Generalization

/seniŋe-RED/	FREE-VOWEL	BPROX/L,SEG
-> [sen_i_ŋ_e]$_B$_[niŋ]$_R$		4
[s_e_n_i_ŋ_e]$_B$_[sen]$_R$		6!
[seniŋ_e]$_B$_[ŋe]$_R$	*!	2

For now, I will not distinguish between left and right edges, as they usually accrue the same number of violations. However, in Section 4, I show a case where BPROX crucially must specify the right edge.

The second change is that I refine the "closeness" part of PROXIMITY to demand that the correspondents not be separated by a phonological boundary, for instance a syllable boundary, which would demand that the correspondents occupy the same syllable. I define the boundary as the meeting point of two phonological units, e.g., there is one syllable boundary between [ma] and [fe] in [ma.fe]. Thus in the fully split candidate [m-m-a-a-f-f-e-e], each Reduplicant segment and its Base correspondent are separated by a segmental boundary, violating BPROX/SEG.[1] Compare this to PROXIMITY, which does not assign violation marks to the fully split candidate, as no segment intervenes between the corresponding elements.

Importantly, the locus of violation of BPROX is the boundary itself, not the corresponding elements. BPROX does not have to count a potential infinite amount of intervening material between the elements in its evaluation, but rather assigns one violation mark to each boundary that meets the qualification of intervening between the two correspondents. Therefore BPROX is a categorical constraint; as McCarthy (2003) and Gouskova (2003) argue, only categorical constraints are valid in OT, as they do not require the capacity for the constraint to count infinitely in its definition; it can, however, count as many loci of violation meet the description in the definition. PROXIMITY as so defined is gradient, as it must count potentially infinite material between corresponding elements (though redefining it so the locus of violation is the intervening segment would make it categorical).

I now turn to the syllable-sized splitting candidate, repeated below in (46). I use a different subconstraint of BPROX, indexed to the syllable boundary and the right edge.

[1] Because of this definition of boundaries, BProx/Seg can only be fully satisfy by geminate, or long, segments, where the Reduplicant and its Base correspondent share the same segment. Thus this analysis makes the strong prediction that Reduplication will surface as gemination or length in some language. I explore this further in section 5.

45) BPROX/R,SYLL: Assign a violation mark for every syllable boundary that intervenes between a right edge element in the Reduplicant and its correspondent in the Base.

The splitting candidate in (46) has two right Reduplicant edges, [a] and [e], each of which has one syllable boundary (represented by ".") intervening between it and its Base correspondent. The single violations of each of the two right edges total two violations of BPROX/R,SYLL. The contiguous candidate in (47) has one right Reduplicant edge, [e], that has two syllable boundaries intervening between it and its Base correspondent. Thus (47) also has two violations of BPROX/R,SYLL.

46) [ma].[ma].[fe].[fe] 2(edges)×1(boundary) = 2 violations

47) [ma.fe].[ma.fe] 1(edge)×2(boundaries) = 2 violations

Again, because the two candidates tie on BPROX/R,SYLL, low-ranking O-CONTIGUITY is left to eliminate the splitting candidate (48).

Table 8. PROXIMITY Pathology Prevented

/RED-mafe/	BPROX/R,SYLL	O-CONTIGUITY
-> $[ma.fe]_R$-.$[ma.fe]_B$	2	
$[ma]_R$-.$[ma]_B$-.$[fe]_R$-.$[fe]_B$	2	2!

Because splitting up the Reduplicant and the Base increases the number of edge elements in the Reduplicant, the decrease in each corresponding pair's violations of BPROX is offset by an increase in the number of relevant pairs whose violations are counted by BPROX. And, because the loci of violation are phonological boundaries, splitting the Reduplicant and Base cannot get rid of intervening boundaries between the edge segments. BPROX does not prevent splitting from

happening at all, but rather minimizes the extent of the splitting. I will show in Sections 3 and 4 attested examples of splitting that are successfully accounted for with Proximity constraints.

3. BOUNDARY-PROXIMITY in Saisiyat

In this section, I provide an analysis of two reduplicative patterns in Saisiyat: Ca- reduplication and Progressive reduplication. I first show how BPROX/SYLL accounts for the pattern of Ca- reduplication in Saisiyat, which does not show order-disruption. I then extend the analysis to Progressive reduplication, in which the order of segments is disrupted. I account for this by ranking BPROX/SYLL over Contiguity, so that order-disruption is tolerated in order to allow for correspondents to occupy the same syllable.

3.1 Saisiyat Ca- Reduplication

In Ca- reduplication, BPROX not only accounts for which consonant gets copied (accounting for Marantz's Generalization), but also motivates the minimal and atemplatic size of the Reduplicant. While either BPROX/SEG or BPROX/SYLL can account for this pattern, I use the latter to explicate the analysis. To account for the position of the copied consonant in this pattern, I first must deal with a relevant matter of Saisiyat phonology, the infixing of vowel-initial prefixes.

3.1.1 Infixing in Saisiyat

In Saisiyat, words must begin with a consonant, though onsetless syllables are allowed medially; I capture this with the prosodic edge-sensitive constraint ONSET/WD, which demands that every ProsodicWord begin with a consonant (Flack 2009).

48) ONSET/WD: Assign a violation mark for an onsetless word-initial syllable.

To satisfy ONSET/WD, vowel-initial words surface with an initial epenthetic glottal stop (49).

49) /oræl/ -> [ʔo.ræl] ‘rain’

ONSET/WD must dominate DEP-C to repair initially onsetless words with glottal-stop epenthesis.

Table 9. ONSET/WD >> DEP-C

/oræl/	ONSET/WD	DEP-C
-> ʔo.ræl		*
o.ræl	*!	

To the same end, vowel-initial prefixes /om-/ ‘AgentFocus’ and /in-/ ‘Perfective/PatientFocus’ are infixed inside an initial consonant (50-51).

50) /om-ħayap/ -> [ħ-o.m-a.yap] ‘fly-AF’

51) /in-bamøħ/ -> [b-i.n-a.møħ] ‘grow-Prf.PF’ (= “grown plant”)

To account for the infixing, ONSET/WD and DEP-C must be ranked above whatever constraint motivates the leftward tendency of the relevant affixes (i.e., the prefixed ones). Following McCarthy (2003), I use two categorical constraints for these affixes, PREFIX(-af-), which demands that no segment precede the relevant affix (i.e., the affix is leftmost), and PREFIX/SYLL(-af-), which demands that no syllable precede the relevant affix.

52) PREFIX(-af-): Assign a violation mark if a (relevant) affix is preceded by a segment in the Prosodic Word

53) PREFIX/SYLL(-af-): Assign a violation mark if a (relevant) affix is preceded by a syllable in the Prosodic Word (McCarthy 2003)

The presence of infixation in Saisiyat show that Prefix(-af-) is violated; however, PREFIX/SYLL(-af-) is never violated, i.e., all infixes occupy the first syllable (at least in part). Ranking

ONSET/WD and DEP-C over PREFIX(-af-) correctly predicts that vowel-initial prefixes are infixed.[2]

Table 10. Vowel-Initial Prefix: Infixing

/om-ħayap/	ONSET/WD	DEP-C	PREFIX(-om-)
-> ħ-om-ayap			*
om-ħayap	*!		
ʔom-ħayap		*!	

Further infixation of vowel-initial prefixes is prevented by the constraint PREFIX/SYLL(-af-). It does not matter where this constraint is ranked with PREFIX(-af-), since they are in a stringency ranking (Prince 1998). Wherever it is ranked, it eliminates deeper infixation.

Table 11. Vowel-Initial Prefix: No Deep Infixing

/om-ħayap/	PREFIX(-om-)	PREFIX/SYLL(-om-)
-> ħ-o.m-a.yap	*	
ħa.y-o.m-ap	*	*!

Consonant-initial prefixes, such as /ka-/ 'Nominalization' are prefixed, not infixed (54), so the infixing of vowel-initial prefixes cannot be accounted for by the general ranking of ANCHOR-ROOT-LEFT above PREFIX(-af-). Note that this cannot purely be an effect of NOCODA, as in McCarthy and Prince's (1993) account of Tagalog um-infixation, because vowel-initial prefixes are infixed even when a coda is created (55). Thus NOCODA cannot favor the infixed candidate (55) over the prefixing candidate.

[2] I assume that all prefixes and leftward infixes have the same ranking of Prefix(-af-). While this is not necessarily demanded by the grammar, it is certainly the null hypothesis here.

54) /ka-kiʂkaat-an/ -> [ka-.kiʂ.ka.a.t-an] ‘Nom-read-Loc’ (= “school”)

55) /om-ʂβət/ -> [ʂ-om-.βət] ‘hit-AF’

Table 12. Consonant-Initial Prefix: No Infixing

/ka-kiʂkaat-an/	ONSET/WD	DEP-C	PREFIX(-ka-)
-> ka-kiskaat-an			
ki-ka-skaat-an			*!

The demand for a word to begin with a consonant (non-epenthetic if possible) is crucial to account for the position of the copied consonant in the patterns below.

3.1.2 Ca- Reduplication

Now that I have established the need for an initial onset in Saisiyat, I can account for Ca-reduplication using BPROX/SYLL. The presence of the vowel [a] cannot be an effect of The Emergence of The Unmarked (TETU: McCarthy and Prince 1994), because [ə], not [a], is default epenthetic vowel in Saisiyat. For example, complex onsets in Saisiyat are avoided through [ə]-epenthesis (56).

56) /ʂβət-ən/ -> [ʂə.βə.t-ən] ‘hit-PF’

If this vowel were epenthetic, due to a TETU ranking, then the vowel between the initial consonant and its copy would be [ə]. Therefore I assume, following Alderete et al. (1999), that the fixed segment [a] is a separate prefix /a-/, which prevents the Reduplicant from copying a Base vowel, as in (57).

57) /a,RED,ħila/ -> [ħa.ħi.la]

While Alderete et al. do not explicitly state whether the fixed segment is part of the Reduplicant in the output, I assume that the C alone constitutes the Reduplicant, so that the output [ħaħila] in (57) is parsed as $ħ_{RED}$-a-ħila. While there is no evidence that the /a/ and the /RED/ are exponents of separate morphemes that carry different semantic meanings, the definition of RED is that it is empty of material in the input, and thus cannot have segmental prespecification. I therefore follow Wolf (2008) in assuming the possibility of multiple exponence, whereby a single morpheme can have more than one morph that expresses its meaning in the phonological component. In this analysis, both the /a/ and the /RED/ are exponents of the same morpheme (variously "instrument," "reciprocal," etc.), but they are two different morphs. Since the [a] is a separate morph in the input, it must still be separate from the Reduplicant in the output (Wolf 2008). I assume that the [a] is underlyingly a prefix, not an infix, and its output position is an effect of the general Saisiyat process of infixation outlined above, like the other vowel-initial prefixes /om/ and /in/.

BPROX/SYLL accounts for the copying of the root-initial consonant [ħ], rather than the root-internal consonant [l], as in the candidate [l-a-ħila] (note that following Kennedy's (2005) analysis, I do not use BR-ANCHOR constraints). The copies of [ħ] in [ħ-a-.ħi.la] are only one syllable apart, while the copies of [l] in [l-a-.ħi.la] are two syllables apart; thus BPROX/SYLL favors the former.[3]

[3] A coda consonant in the root-initial syllable is never copied into the Reduplicant (e.g., *[k-a-ʂək.laʔ] instead of [ʂ-a-ʂək.laʔ]). This can be accounted for using BProx/Syll, since the copies of [k] in [k-a-ʂək.laʔ] are further apart by segments than are the copies of [ʂ] in [ʂ-a-ʂək.laʔ]. In Section 4, I elaborate on the interaction of BProx/Seg and BProx/Syll in Pima Plural Reduplication.

Table 13. BPROX/SYLL and Marantz's Generalization

/a,RED,ħila/	BPROX/SYLL
-> ħ-a-.ħi.la	*
l-a-.ħi.la	**!

I account for the position of the reduplicated consonant to the left of the prefix [a-], rather than to its right as in the candidates [a-ħ-.ħila] and [ʔa-ħ-.ħila], by using the constraint ranking established to account for infixation of other vowel-initial prefixes: ONSET-WD >> DEP-C >> PREFIX(-af-). PREFIX(-a-) and PREFIX(RED) do not need to be ranked with respect to each other: since /a/ is a vowel, it must follow the Reduplicant, which is a consonant.[4]

Table 14. Infixation of Fixed Segment

/a,RED,ħila/	ONSET/Wd	DEP-C	PREFIX(-a-)	PREFIX(RED)
-> ħ-a-.ħi.la			*	
a-ħ-.ħila	*!			*
ʔa-ħ-.ħila		*!		*

The copies of [ħ] in [ħ-a-.ħi.la] are in distinct (though adjacent) syllables, which violates BPROX/SYLL once. To eliminate the challenger [ħ-a-ħ.li.a], which does not violate BPROX/SYLL (as both copies of [ħ] are in the same syllable), another constraint must outrank BPROX/SYLL. This form violates LINEARITY (McCarthy and Prince 1994, 1995), as the input sequence /...il.../ surfaces as [...li...] (this is necessary to get the second copy of [h] into the coda of the initial syllable).

[4] The Reduplicant must be a consonant because of both BProx (which in effect demands that the Reduplicant copy the Root-initial segment, which is a consonant) and general Saisiyat phonotactics, which prefers the consonant-reduplicating [ħ-a-ħila] to the vowel-reduplicating [a-.i-.ħila].

58) LINEARITY: S_1 (e.g., the Reduplicant) is consistent with the precedence structure of S_2 (e.g., the Base), and vice-versa. (McCarthy and Prince 1995)

Thus, LINEARITY must outrank BPROX/SYLL.

Table 15. LINEARITY >> BPROX/SYLL

/a,RED,ħila/	LINEARITY	BPROX/SYLL
-> ħ-a-.ħi.la		*
ħ-a-ħ.li.a	*!	

The candidate [ħ-a-ħ.ʔi.la] respects PROXIMITY/SYLLABLE and LINEARITY, but fatally violates DEP-C in order to force the copies of [ħ] into the same syllable.

Table 16. DEP-C >> BPROX/SYLL

/a,RED,ħila/	DEP-C	BPROX/SYLL
-> ħ-a-.ħi.la		*
ħ-a-ħ.ʔi.la	*!	

In order for only the first consonant to reduplicate, BPROX/SYLL must outrank BR-MAX, which demands that the Reduplicant be a complete copy of the Base. This ranking eliminates the candidate [ħi.l-a-.ħi.la], which copies more material at the expense of moving the edge of the Reduplicant another syllable away from its correspondent.

Table 17. BPROX/SYLL >> BR-MAX

/a,RED,ħila/	BPROX/SYLL	BR-MAX
-> ħ-a-.ħi.la	*	a...ila
ħi.l-a-.ħi.la	**!	a...a

The complete ranking in (59) accounts for Ca- reduplication. Note that PROXIMITY/SYLLABLE is crucial to keep the Reduplicant minimal; templatic constraints cannot account for this size restriction, because the single-consonant Reduplicant is subtemplatic. Moreover, an economy constraint, such as *SEGMENT, would necessitate a trickier ranking: it would have to outrank BR-MAX to keep copying minimal, but then be outranked by M-PARSE, so that the Reduplicant would surface at all.

59) ONSET/WD, DEP-C, LINEARITY >> BPROX/SYLL >> BR-MAX

The master tableau shows how this ranking chooses the correct winner.

Table 18. Ca- Reduplication

/a-RED-ħila/	ONSET/WD	LINEARITY	DEP-C	BPROX/SYLL	BR-MAX
-> ħ-a-ħila				*	ila
ħ-a-ħ.li.a		*!			ila
ħ-a-ħ.ʔila			*!		ila
a-.ħ-iħ.la	*!	*!			ila
ʔa-.ħ-iħ.la		*!	*!		ila
l-a-.ħi.la				**!	ħia
ħi.l-a-.ħi.la				**!	a

3.2 Order-Disruption in Saisiyat Progressive Reduplication

I now turn to Progressive reduplication in Saisiyat, which I will account for using the crucial ranking of BPROX/SYLL over CONTIGUITY. Since I have already shown above that ONSET/WD and DEP-C are high-ranked in Saisiyat, I will not show any candidates that violate it, i.e., all viable candidates must begin with a non-epenthetic consonant. In this pattern, the AgentFocus infix (variously realized as [-om-], [-əm-], or [-øm-]) is split by one of the correspondents of the copied single consonant, as in (60-62) (Zeitoun and Wu forthcoming).

60) /om, RED-kitaʔ/ -> [k-o-k-.m-i.taʔ]

61) /əm, RED-rəmə/ -> [r-ə-r-.m-əmə]

62) /øm, RED-ħæŋiħ/ -> [ħ-ø-ħ-.m-æ.ŋiħ]

In this analysis it does not matter precisely which occurrence of the copied consonant is the Reduplicant, and which is part of the Base. For the purposes of simplicity, I assume that Reduplicant is the first occurrence, i.e., the initial consonant, throughout the analysis. BPROX/SYLL treats both parsings identically, as in both, the Reduplicant and its Base correspondent (e.g., the two instances of [k] in (60)) are in the same syllable.

I now proceed to show the crucial position of BPROX/SYLL in the grammar of Saisiyat. The Reduplicant is a single consonant, e.g., [k] in (60), and is thus both minimal and subtemplatic. The second occurrence of the consonant (e.g., the second [k] in (60)) splits the infix [om], so that O-CONTIGUITY (which demands that output morphemes not be split) is violated twice by (60): once by the splitting of the infix [om], and once by the splitting of the root [kitaʔ]. This invites the challenger candidate in (63), which respects O-CONTIGUITY.

63) $[k]_R[om\text{-}.ki.taʔ]_B$

To choose (60) over (63), the constraint BPROX/SYLL must dominate O-CONTIGUITY. While the corresponding [k]s are dominated by the same syllable in (60), so that no syllable boundary intervenes between them, the corresponding [k]s are in separate syllables in (63), and are thus separated by a syllable boundary. Thus, the winning output respects BPROX/SYLL completely, while the challenger crucially violates it once.

Table 19. BProx/Syll >> O-Contiguity

/om,RED,kitaʔ/	BPROX/SYLL	O-CONTIGUITY
-> $[k]_R[o\text{-}k\text{-}.m\text{-}i.taʔ]_B$		**
$[k]_R[om\text{-}.ki.taʔ]_B$	*!	

There is no other constraint that will favor the winner over the challenger. Prosodically these candidates are identical, both having three syllables, so an economy constraint such as *SYLLABLE cannot decide between them. Both candidates copy the same amount of material (the single consonant [k]) from the Base, so BR-MAX also cannot decide a winner. Moreover, the challenger's syllable transition [m.k], in which the sonority falls, is less marked than that of the winner's [k.m], in which the sonority rises. The constraint CORR-SYLL-ROLE (McCarthy and Prince 1994; Gafos 1996; Kenstowicz 2005), which demands that correspondents play the same role in the syllable, can be used to account for cases of minimal, discontiguous reduplication (Nuger 2006). However, the constraint actually favors the challenger over the winner, since the [k]s are both onsets in the former but an onset and a coda in the latter. Thus a BPROX constraint on the distance between correspondents is the only constraint that can eliminate the challenger. BPROX/SYLL specifically is necessary to account for this Saisiyat pattern.

I next account for why the Reduplicant is minimal, i.e., why only the one consonant [k] is copied. The challenger in (64) copies more material than the winner, but each edgemost element in the Reduplicant is the same distance away from its Base correspondent in both candidates in segments: in both cases, one segment intervenes.

64) $[ko]_R.[k\text{-}o.m\text{-}i.ta\text{ʔ}]_B$

(64) does better than the winner on BR-MAX, as it copies two segments of the Base rather than just one. (64) also does better on O-CONTIGUITY, since the infix [om] is not split by one of the copies of [k]. Therefore a higher-ranked constraint than these must favor the winning output over (64). This constraint cannot be a version of BPROX evaluating violations by segment (i.e., that assesses a violation mark for every segment boundary intervening between correspondents), since the copies of [k] are separated by one segment ([o]) in both the challenger and the winner. It is thus crucial that BPROX can refer to syllable boundaries: in (64), the copies of [k] are in different syllables, so it violates BPROX/SYLL. In the winner, both copies of [k] are in the same syllable, so it does not violate BPROX/SYLL; thus this constraint is needed to favor the winner over the challenger.

Table 20. BPROX/SYLL >> BR-MAX

/om,RED,kitaʔ/	BPROX/SYLL	BR-MAX
-> $[k]_R[o\text{-}k\text{-}.m\text{-}i.ta\text{ʔ}]_B$		o...mitaʔ
$[ko]_R.[k\text{-}o.m\text{-}i.ta\text{ʔ}]_B$	*!	mitaʔ

Another output candidate to consider is [m-om-.ki.taʔ], where the consonant of the infix is reduplicated instead of that of the root. This candidate does not violate BPROX/SYLL, and moreover does not violate O-CONTIGUITY, unlike the winner. To eliminate this candidate, I

invoke the tendency of Reduplicant to copy from root material and not affixal material. To capture this tendency, I use a subconstraint of BR-MAX, BR-MAX-ROOT (see, e.g., McCarthy and Prince 1995, Benua 1997 for differentiating root faithfulness constraints from general versions), which in effect prefers copying root material over affix material.

65) BR-MAX-ROOT: every element in the Reduplicant must have a correspondent in the Root in the Base.

Ranking BR-MAX-ROOT over O-CONTIGUITY eliminates this challenger.

Table 21. BR-MAX-ROOT >> O-CONTIGUITY

/om,RED,kitaʔ/	BR-MAX-ROOT	O-CONTIGUITY
-> k-o-k-.m-i.taʔ		**
m-om-.kitaʔ	*!	

There is another interesting form of this reduplicative pattern where the reduplicant cannot fully satisfy BPROX/SYLL. In (66), the Reduplicant is the first CV of the Base, so the corresponding instances of [ʂ] are separated by a syllable boundary, thus violating BPROX/SYLL.

66) /om,RED,ʂβət/-> [ʂo-.ʂ-om-.βət]

Challenging candidates that satisfy BPROX/SYLL, however, violate other constraints. The candidates in (67) violate Saisiyat phonotactics, which enforce a maximum CV(:)C syllable. No complex codas or onsets are permissible even in non-reduplicative contexts, so *COMPLEX is high-ranked. The candidates in (67) violate *COMPLEX, and thus cannot surface.

67) [ʂ-o-ʂ-m-.βət], [ʂ-o-ʂ-.m-βət]

Table 22. *COMPLEX >> BPROX/SYLL

/om,RED,ʂβət/	*COMPLEX	BPROX/SYLL
-> ʂo-.s-om-.βət		*
ʂ-o-ʂ-m-.βət	*!	
ʂ-o-ʂ-.m-βət	*!	

To prevent complex margins in non-reduplicated words, a schwa is epenthesized to break up the cluster (68). This is captured by ranking *COMPLEX over DEP-V.

68) /ʂβət/ -> [ʂə.βət]

Table 23. *COMPLEX >> DEP-V

/ʂβət/	*COMPLEX	DEP-V
-> ʂə.βət		*
ʂβət	*!	

However, a schwa is not epenthesized to satisfy BOUNDARY-PROXIMITY/SYLLABLE, as in the challenger in (69).

69) $[ʂ]_R[o\text{-}ʂ\text{-}.m\text{-}əβət]_B$

Thus DEP-V, which is violated by the schwa epenthesis in (69), must dominate BPROX/SYLL.

Table 24. DEP-V >> BPROX/SYLL

/om,RED,ʂβət/	DEP-V	BPROX/SYLL
ʂo-.ʂ-om-.βət		*
ʂ-o-ʂ-.m-ə.βət	*!	

Neither can a vowel from the root be copied into this position in order to satisfy BPROX/SYLL, as in the challenger in (70).

70) $[ʂ]_R[o\text{-}ʂ\text{-}.m]_B[ə]_R.[βət]_B$

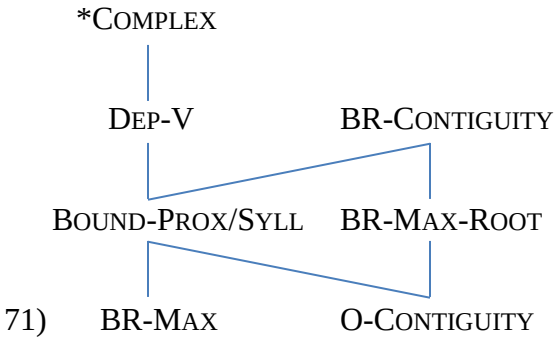
This constraint violates BR-CONTIGUITY, as the actual Base and Reduplicant portions are both split (as opposed to only morphemes within the Base), and thus discontiguous. Therefore BR-Contiguity must also dominate BOUNDARY-PROximity/SYLLABLE. In addition, because (70) copies more root material, BR-CONTIGUITY must dominate BR-Max-Root as well.[5]

Table 25. BR-CONTIGUITY >> BPROX/SYLL

/om,RED,ʂβət/	BR-CONTIGUITY	BPROX/SYLL	BR-MAX-ROOT
$[ʂo]_R.[ʂ\text{-}om\text{-}.βət]_B$		*	*
$[ʂ]_R[o\text{-}ʂ\text{-}.m]_B[ə]_R.[βət]_B$	**!		

The complete ranking in (71) accounts for Saisiyat Progressive reduplication. Note again how BPROX/SYLL accounts for both the minimal size of the reduplicant and the infixing of the Base correspondent into the infix [om]. No other constraint is available that can account for this infixing; an economy constraint cannot favor the CONTIGUITY-violating winner $[k]_R[o\text{-}k\text{-}mitaʔ]_B$ over the CONTIGUITY-respecting challenger $[k]_R[om\text{-}kitaʔ]_B$.

[5] The very existence of BR-Contiguity in the grammar seems to favor a prefixed Reduplicant like $[k]_R[o\text{-}k\text{-}mitaʔ]_B$ over an infixed Reduplicant like $[ko]_B[k]_R[mitaʔ]_B$. Since these two forms only differ in morphological affiliation and not in phonological content, only a morphologically-sensitive constraint like BR-Contiguity (or perhaps Prefix(RED)) can tell them apart. I cannot think of such a constraint that favors an infixed Reduplicant over a prefixed Reduplicant.



71)

The master tableau shows how this ranking selects the winners, [k-o-k-.m-ita?] (from /om,RED,kita?/) and [ŋo-.ŋ-om-.βət] (from /om,RED,ŋβət/).

Table 26. Progressive Reduplication in Saisiyat

/om,RED,kita?/	*COM-PLEX	BR-CONTIG	DEP-V	BPROX/SYLL	BR-MAX-ROOT	BR-MAX	O-CONTIG
-> [k]$_R$[o-k-.m-i.ta?]$_B$					ita?	o...mita?	**
[k]$_R$[om-.kita?]$_B$				*!	ita?	om...ita?	
[ko]$_R$.[k-o.m-i.ta?]$_B$				*!	ita?	mita?	*
[ki]$_R$.[k-o.m-i.ta?]$_B$				*!	ta?	o...ita?	*
[m]$_R$[om-.kita?]$_B$					kita?!	o...kita?	
/om,RED,ŋβət/	*COM-PLEX	BR-CONTIG	Dep-V	BPROX/SYLL	BR-MAX-ROOT	BR-MAX	O-CONTIG
-> [ŋo]$_R$.[ŋ-om-.βət]$_B$				*	βət	m...βət	*
[ŋ]$_R$[o-ŋ-m-.βət]$_B$	*!				βət	o...mβət	**
[ŋ]$_R$[o-ŋ-.m-βət]$_B$	*!				βət	om...βət	**
[ŋ]$_R$[o-ŋ-.m-ə.βət]$_B$			*!		βət	o...məβət	**
[ŋ]$_R$[o-ŋ-.m]$_B$[ə]$_R$.βət]$_B$		*!			βt		

4. BOUNDARY-PROXIMITY in Pima Plural Reduplication

BOUNDARY-PROXIMITY is able to give a simple, elegant analysis to Plural Reduplication in Pima. In the most common form of this pattern, an initial consonant is reduplicated, with one instance remaining word-initial and the second instance occupying the coda of the first syllable, as in (72-74) (Riggle 2004, 2006).

72) [ma.vit] ‘lion (sg.)’ -> [mam.vit] ‘lion (pl.)’

73) [nak] ‘earlobe (sg.)’ -> [nank] ‘earlobe (pl.)’

74) [kos.vuɭ] ‘cocoon (sg.)’ -> [koks.vuɭ] ‘cocoon (pl.)’

However, in certain contexts, an initial CV sequence is copied (75-80). These contexts all involve avoiding particular codas, including laryngeal codas (75-76), palatal nasal codas (77), and codas that violate sonority restrictions (e.g., they have sonority plateaus (78) or reversals (79), or the fall in sonority is not large enough (80)).

75) [ho.dai] ‘rock (sg.)’ -> [ho.ho.dai] ‘rock (pl.)’; *[hoh.dai]

76) [ʔi.put] ‘dress (sg.)’ -> [ʔi.ʔi.put] ‘dress (pl.)’; *[ʔiʔ.put]

77) [ɲɨ.pod] ‘night hawk (sg.)’ -> [ɲɨ.ɲɨ.pod] ‘night hawk (pl.)’; *[ɲɨɲ.pod]

78) [namks] ‘joint (sg.)’ -> [na.namks] ‘joint (pl.)’; *[nanmks]

79) [kan.ʤɨɭ] ‘candle (sg.)’ -> [ka.kan.ʤɨɭ] ‘candle (pl.)’; *[kakn.ʤɨɭ], [kak.nʤɨɭ]

80) [ɭan.ʤɨ.ki] ‘lentil (sg.)’ -> [ɭa.ɭan.ʤɨ.ki] ‘lentil (pl.)’; *[ɭaɭn.ʤɨ.ki]

The appropriate descriptive generalization (from Riggle 2006) is that the only the initial consonant is copied if the copy is a licit coda, in which case it occupies the coda of the first syllable. If copying only the initial consonant cannot result in a licit structure (e.g., because it results in a laryngeal or palatal nasal coda, or violates Sonority Sequencing), then an adjacent

vowel is also copied. In general, palatal nasal codas and Sonority Sequencing violations are possible in Pima, so their prohibition here must be due to The Emergence of The Unmarked.

4.1 Parsing the Reduplicated Forms

The question now arises of the parsing of the single-C reduplicated forms, i.e., which copy of the initial consonant is in the Reduplicant and which is in the Base. For the form in (72) [mam.vit], there are three possible parsings, as shown in (81a-c) (cf. Riggle 2006, who only acknowledges two possibilities, (81a) and (81c)).

81) a. $[ma]_R[m.vit]_B$ b. $[m]_R[am.vit]_B$ c. $[ma]_B[m]_R[vit]_B$

In (81a), the Reduplicant is prefixed to the Base, and consists of the first CV of the word. Because of the syncope of the vowel [a] in the Base, the Reduplicant contains material that is not present in the Base, namely, that vowel [a]. To account for this, the Reduplicant needs access to the input form of the word, /mavit/, so it can copy the [a]. This approach necessitates the use of McCarthy and Prince's (1995) Full Model of Reduplication, including IR-Faithfulness constraints in addition to the IO-Faithfulness and BR-Faithfulness required by the Basic Model (McCarthy and Prince 1995). As I put forth in Section 1, I do not use the Full Model for reasons of parsimony, in addition to the general arguments against the Full Model (e.g., Idsardi and Raimy 1997, Struijke 2002, Inkelas and Zoll 2005) on both empirical and theoretical grounds. Thus if it is possible to account for this reduplicative pattern without letting the Reduplicant have access to the input, such an account will be superior to one that requires such access.

The parsings (81b-c) do not necessitate the Reduplicant having access to the input, because the Reduplicant only consists of the single copied consonant. In both parsings, the positions of elements is disrupted, violating Faithfulness constraints. In (81b), the linear order of the Base

[amvit] is changed from the input /mavit/, so that the order of the segments /ma/ is reversed, violating LINEARITY. In (81c), the Reduplicant [m] breaks the Base into two parts, [ma] and [vit], thus violating BR-CONTIGUITY. Additionally, the infixation of the Reduplicant prefix in (81c) misaligns it from the left edge, violating PREFIX(RED). More theoretically, (81c) assumes that the Base is the output strings both preceding and following the Reduplicant. This constitutes an extension of the normal definition of the Base (which the parsings in (81a-b) adhere to): "the output string of segments to which the reduplicant is attached; the following string if the reduplicant is a prefix, and the preceding string if the reduplicant is a suffix" (Kager 1999). While this does constitute an extension of the typical definition of the Base, it is not necessarily undesirable in itself.

Here, I remain agnostic as to which parsing, (81b) $[m]_R[amvit]_B$ or (81c) $[ma]_B[m]_R[vit]_B$, is preferable on its own rate. In fact, I give accounts for both parsings that are mostly similar; for (81b), LINEARITY is crucially dominated, while for (81c), both BR-CONTIGUITY and PREFIX(RED) are crucially dominated. Both parsings otherwise have the same violation profiles, including on phonotactic and prosodic constraints and BR-MAX.

4.2 Analysis: BPROX/SYLL

In the sub-pattern exemplified by [mam.vit], the Reduplicant is a single consonant, and so is both minimal and atemplatic. Moreover, the input sequence /mavit/ is not preserved in the output form: this sequence is interrupted by the second copy of [m]. Accounting for this form requires the ranking to disfavor the minimally reduplicating challenger where the sequence /mavit/ is preserved in the output (82).

82) $[ma]_R.[ma.vit]_B$

By preserving the ordering and contiguity of the input sequence, (82) satisfies both LINEARITY and O-CONTIGUITY. Moreover, (82) beats the winning output on BR-MAX, as it copies more Base material. A constraint is thus needed to favor the winner over this challenger. BPROX/SYLL is in fact such a constraint: in the winner, the corresponding copies of [m] are in the same syllable, while in (82), a syllable boundary intervenes between these copies. Thus BPROX/SYLL must dominate constraints on faithfulness to the input sequence.

Which faithfulness constraints must be dominated depends on the parsing of the winner. If the first instance of [m] composes the Reduplicant, then BPROX/SYLL must dominate LINEARITY.

Table 27. BPROX/SYLL >> LINEARITY

/RED,mavit/	BPROX/SYLL	LINEARITY
-> $[m]_R[am.vit]_B$		*
$[ma]_R.[ma.vit]_B$	*!	

If the second instance of [m] composes the Reduplicant, then BPROX/SYLL must dominate both BR-CONTIGUITY and PREFIX(RED).

Table 28. BPROX/SYLL >> BR-CONTIGUITY, PREFIX(RED)

/RED,mavit/	BPROX/SYLL	BR-CONTIGUITY	PREFIX(RED)
-> $[ma]_B[m]_R.[vit]_B$		*	*
$[ma]_R.[ma.vit]_B$	*!		

In both [mam.vit] and [ma.ma.vit], the Reduplicant edge segments and their Base correspondents are separated by a single segment. Thus, the subconstraint of BPROX evaluating violations by intervening segment boundaries (BPROX/SEG) cannot favor the winner over the challenger,

because it is violated equally by both candidates. BPROX/SYLL therefore is absolutely necessary here to decide in favor of the winner.

BPROX/SYLL must also dominate BR-MAX for both parsings, as $[ma]_R[mavit]_B$ copies more material from the Base than either $[m]_R[amvit]_B$ or $[ma]_B[m]_R[vit]_B$, and thus BR-MAX favors the former.

Table 29. BPROX/SYLL >> BR-MAX

/RED,mavit/	BPROX/SYLL	BR-MAX
-> $[m]_R[am.vit]_B$		avit
-> $[ma]_B[m]_R.[vit]_B$		avit
$[ma]_R.[ma.vit]_B$	*!	vit

I now turn to the sub-pattern in which a CV sequence is copied rather than a single consonant. In these forms, such as (83), BPROX/SYLL is violated because the edge elements ([h] and [o] in (83)) of the Reduplicant are separated from their correspondents in the Base by a syllable boundary.

83) /RED,hodai/ -> [ho-.ho.dai]

Constraints dominating BPROX/SYLL must disfavor the challenging candidate in (84) that respects BPROX/SYLL (as in (84) both copies of [h] are in the same syllable).

84) [hoh.dai]

In all the cases of CV Reduplicants in Pima Plural reduplication, the BPROX/SYLL-respecting challenger violates a phonotactic constraint; (84), for instance, violates $*LAR]_{Syll}$ (Riggle 2006). Ranking $*LAR]_{Syll}$ above BPROX/SYLL picks the winning CV Reduplicant in (74).

Table 30. PHONOTACTICS >> BPROX/SYLL

/RED,hodai/	*LAR]$_{Syll}$	BPROX/SYLL
-> [ho]$_R$.[ho.dai]$_B$		*
[h]$_R$[oh.dai]$_B$	*!	
[ho]$_B$[h]$_R$[.dai]$_B$	*!	

*LAR]$_{Syll}$ is generally unviolated in Pima, and thus dominates IO-Faithfulness. The other phonotactic constraints that force CV Reduplication, *ɲ]$_{Syll}$ and SSP (the SONORITY SEQUENCING PRINCIPLE), are violated in non-reduplicative contexts (85-86), and thus must be dominated by IO-Faithfulness (Riggle 2006).

85) [to.toŋ] 'ant'

86) [totpk] 'purr'

For simplicity of exposition, I ignore IO-Faithfulness and collapse these constraints as PHONOTACTICS. Thus the ranking for Pima Plural Reduplication (in the simplest cases) is as in (87-88), where (87) accounts for parsing the Reduplicant as a prefix and (88) for parsing it as an infix.

87) PHONOTACTICS >> PROXIMITY/SYLLABLE >> LINEARITY, BR-MAX

88) PHONOTACTICS >> PROXIMITY/SYLLABLE >> BR-CONTIGUITY, PREFIX(RED), BR-MAX

I illustrate these ranking with the two master tableaux below.[6]

[6] Monosyllabic forms like [nank] from [nak] 'earlobe', which have coda clusters, require a way to regulate the position of the second instance of the copied consonant (here, [n]). This can be done with either the SSP or Boundary-Proximity/Segment, which I illustrate below.

Table 31. Pima Plural Reduplication (Prefixing Parsing)

/RED,mavit/	PHONOTACTICS	BPROX/SYLL	LINEARITY	BR-MAX
-> $[m]_R[am.vit]_B$			*	a...vit
$[ma]_R.[ma.vit]_B$		*!		vit
/RED,hodai/	PHONOTACTICS	BPROX/SYLL	LINEARITY	BR-MAX
-> $[ho]_R.[ho.dai]_B$		*		dai
$[h]_R[oh.dai]_B$	*!		*	o...dai

Table 32. Pima Plural Reduplication (Infixing Parsing)

/RED,mavit/	PHONOTACTICS	BPROX/ SYLL	BR- CONTIGUITY	PREFIX(RED)	BR- MAX
-> $[ma]_B[m]_R.[vit]_B$			*	*	a...vit
$[ma]_R.[ma.vit]_B$		*!			vit
/RED,hodai/	PHONOTACTICS	BPROX/ SYLL	BR- CONTIGUITY	PREFIX(RED)	BR- MAX
-> $[ho]_R.[ho.dai]_B$		*			dai
$[ho]_B[h]_R.[dai]_B$	*!		*	*	o...dai

4.3 Further Sub-patterns: BPROX/SEG

There are two sub-patterns of Plural reduplication in Pima that require closer attention. These sub-patterns make further BPROX demands on elements in correspondence: rather than merely needing to be close to one another in terms of syllables, the copies also need to be close to one another in terms of segments. When the first nucleus of the root is a diphthong, the copied

portion is the initial consonant and the first member of the diphthong, as in (89-91). Note that this copying is not minimal, as not just the consonant is copied (vs. (92)), and the copies are in different syllables, violating BPROX/SYLL (vs. (93)).

89) [kua.di] ‘twin (sg.)’ -> [ku.kua.di] ‘twin (pl.)’

90) [piast] ‘party (sg.)’ -> [pi.piast] ‘party (pl.)’

91) [tai.vig] ‘firefly (sg.)’ -> [ta.tai.vig] ‘firefly (pl.)’

92) *[ku.ka.di], *[pi.past], *[ta.ti.vig]

93) *[kuak.di], *[piapst], *[tait.vig]

Because BPROX/SYLL is respected in the challenger candidates (92) but violated in the winners (88-90), another, higher-ranking constraint is needed to favor the winners over the challengers. Notice that while the copies of the initial consonant in the winners are further away from each other in terms of syllables than are the copies in the challengers, they are actually closer in terms of segments. For instance, in (89) the copies of [k] are only separated by one segment [u], while in the corresponding challenger in (93), they are separated by two segments ([ua]). Thus, a subconstraint of BPROX sensitive to segmental rather than syllable boundaries is needed to favor the winner [ku.kua.di] over the challenger [kuak.di]. I define this constraint below:

94) BOUNDARY-PROXIMITY/SEGMENT (BPROX/SEG): Assign a violation mark for every segmental boundary that intervenes between the left/rightmost element in the Reduplicant and its correspondent in the Base.

This constraint simply substitutes the category ‘segment’ for ‘syllable’; it does not change the framework of BPROX, which evaluates category boundaries intervening between edgemost

segments and their correspondents. Thus it avoids the pathology encountered by the segment-counting formulation of PROXIMITY, as shown in Section 2 above.

Returning to the Pima data, ranking BPROX/SEG above BPROX/SYLL accounts for the position of the internal copied consonant in (89-91). In [ku.kua.di], only two segment boundaries intervene between the copies of [k], while in [kuak.di], three segment boundaries do so. I show this below using "_" to mark segmental boundaries:

95) [k_u_kuadi] 2 violations of BPROX/SEG

96) [k_u_a_kdi] 3 violations of BPROX/SEG

The tableau below shows how the ranking of BPROX/SEG above BPROX/SYLL is necessary to select the winner.

Table 33. BPROX/SEG >> BPROX/SYLL

/RED,kuadi/	BPROX/SEG	BPROX/SYLL
-> $[ku]_R.[kua.di]_B$	**	*
$[k]_R[uak.di]_B$	***!	

Note that in the forms without diphthongs, as in Section 4.2 above, the CV- copying candidates and C-copying candidates tie on BPROX/SEG (97-98), so BPROX/SYLL is left to choose C-copying candidates where phonotactically permissible.

97) m_a_m.vit 2 violations of BPROX/SEG

98) m_a_.ma.vit 2 violations of BPROX/SEG

Table 34. Monophthongal Forms

/RED,mavit/	BPROX/SEG	BPROX/SYLL
-> $[m]_R[am.vit]_B$	**	
$[ma]_R.[ma.vit]_B$	**	*!

The challengers in (92), where only the single consonant and not a CV sequence is copied (e.g., [ku.ka.di]), are eliminated by low-ranking constraints on faithfulness to the input sequence of material, as they change this sequence without fewer violations of BPROX/SEG or BPROX/SYLL to compensate. If the first copy of the consonant is the Reduplicant, then LINEARITY is violated in the Base ([ukadi] vs. /kuadi/). If the second copy of the consonant is the Reduplicant, then both PREFIX(RED) and BR-CONTIGUITY are violated (the Reduplicant is infixed inside the Base). Because these constraints are ranked below BPROX/SYLL, this is not a problem for either $[m]_R[am.vit]_B$, which violates LINEARITY, or $[ma]_B[m]_R.[vit]_B$, which violates PREFIX(RED) and BR-CONTIGUITY. BPROX/SYLL favors these winners over candidates that respect the lower-ranked constraints.

Table 35. CV- Reduplication in Diphthongal Forms

/RED,kuadi/	BPROX/SEG	BPROX/SYLL	LINEARITY	BR-CONTIGUITY	PREFIX (RED)
-> $[ku]_R.[kua.di]_B$	**	*			
$[k]_R[u.ka.di]_B$	**	*	*!		
$[ku]_B.[k]_R[a.di]_B$	**	*		*!	*!

Next I move on to the second subpattern of Pima Plural Reduplication that requires a crucial ranking of BPROX/SEG. When a word begins in a complex onset (only found in loanwords), only

the second member is copied (99-102). The distribution of single C versus CV copying is the same as above: if single-C copying can create a licit coda, it is preferred (99-100), while if it cannot create a licit coda, CV copying results (101-102).

99)	[tɬo.gi] ‘truck (sg.)’	->	[tɬoɬ.gi] ‘truck (pl.)’
100)	[kɬa.vo] ‘nail (sg.)’	->	[kɬaɬ.vo] ‘nail (pl.)’
101)	[tɬam.ba] ‘tramp (sg.)’	->	[tɬa.ɬam.ba] ‘tramp (pl.)’
102)	[pɬan.ʤa.kud] ‘an iron (sg.)	->	[pɬa.ɬan.ʤa.kud] ‘an iron (pl.)’

To capture the single-C copying cases, BPROX/SEG is needed to select the second member of the complex onset (e.g., (99), repeated in (103)) rather than the first member (e.g., [tɬot.gi] in (104)), which always violates it once more. Copying the whole complex onset along with the following vowel (e.g., [tɬo.tɬo.gi] in (105)) also violates BPROX/SEG more, as well as BPROX/SYLL.

103)	tɬ_o_ɬ.gi	2 violations of BPROX/SEG
104)	t_ɬ_o_t.gi	3 violations of BPROX/SEG
105)	t_ɬ_o_.tɬo.gi	3 violations of BPROX/SEG, 1 of BPROX/SYLL

The ranking in Table (33) above easily accounts for this case, shown in Table (36).

Table 36. Single-C Reduplication in Complex Onset Forms

/RED,tɬogi/	BPROX/SEG	BPROX/SYLL
-> $[t]_B[ɬ]_R[oɬ.gi]_B$	**	
$[t]_R[ɬot.gi]_B$	***!	
$[tɬo]_R.[tɬo.gi]_B$	***!	*

The CV copying cases also requires the presence of BPROX/SEG to favor the winner (e.g., (101), repeated in (106)) over a challenger that copies the complex onset (e.g., [t͡ɬa.t͡ɬam.ba] in (107)).

106) t͡ɬ_a_.͡ɬam.ba 2 violations of BPROX/SEG

107) t_͡ɬ_a_.t͡ɬam.ba 3 violations of BPROX/SEG

Note that in the winner (106), the Reduplicant is necessarily infixed: the leftmost consonant, [t], is not copied (i.e., it only occurs once), and thus must be part of the Base, not the Reduplicant. A non-infixed challenger, where the Reduplicant is leftmost in the word (e.g., [͡ɬa-.t͡ɬam.ba] in (108), [t͡ɬa-.t͡ɬam.ba] in (109), or [ta-.t͡ɬam.ba] in (110)) must therefore be eliminated. The first and second challengers (108) violate BPROX/SEG once more each than the winner (109).

108) ͡ɬ_a_.t_͡ɬam.ba 3 violations of BPROX/SEG

109) t_͡ɬ_a_.t͡ɬam.ba 3 violations of BPROX/SEG

The third challenger (110) violates BPROX/SEG as much as the winner according to the left edge of the Reduplicant; however, evaluated according to the right edge of the Reduplicant (111-112), this challenger (112) does violate BPROX/SEG more than does the winner (111).

110) t_a_.t͡ɬam.ba 2 violations of BPROX/L,SEG

111) t͡ɬa_.͡ɬ_am.ba 2 violations of BPROX/R,SEG

112) ta_.t_͡ɬ_am.ba 3 violations of BPROX/R,SEG

Therefore, adjusting the above ranking in Tables (33-36) to specify BPROX/R,SEG, this case too is accounted for.

Table 37. CV Reduplication in Complex Onset Forms

/RED,tɬamba/	BPROX/R,SEG	BPROX/SYLL
-> tɬa.ɬam.ba	**	*
ta.tɬam.ba	***!	*
ɬa.tɬam.ba	***!	*
tɬa.tɬam.ba	***!	*

Note that BPROX/SEG by itself cannot account for the placement of the second copy of the consonant in the coda of the initial syllable in the Single-C reduplicative pattern. The copies of the initial consonant in the winner (113) are the same distance from each other in segment as the copies in the challenger (114).

113) m_a_m.vit 2 violations of BPROX/SEG

114) m_a_.ma.vit 2 violations of BPROX/SEG

Thus BPROX/SYLL is absolutely necessary to favor (114) over (113): the winner satisfies it perfectly, while the challenger violates it once. Therefore both subconstraints of BPROX are necessary to account for Pima Plural reduplication.

BPROX/SEG also does favor challenger candidates that copy from a Base-internal consonant (115-116) over winners that copy from a Base-initial consonant (117-118). These challengers only violate BPROX/SEG once, while the winners violate it more than once.

115) mav_.vit 1 violation of BPROX/SEG

116) hod_.dai 1 violation of BPROX/SEG

117) m_a_m.vit 2 violations of BPROX/SEG

118) ho_.h_o.dai 2 violations of BPROX/SEG

Since BPROX/SEG must outrank BPROX/SYLL to favor winners like [ku.kua.di] over challengers like [kuak.di], the greater violation of BPROX/SYLL in (115) vis-à-vis (117) cannot eliminate the challenger. Furthermore, the challenger in (116) and the winner in (118) both violate BPROX/SYLL once, so it cannot be called in to favor the winner over the challenger even in the absence of BPROX/SEG. Another constraint dominating BPROX/SEG are needed to eliminate the challengers in favor of the winners. Neither PREFIX(RED) nor BR-CONTIGUITY can do so, because these would wrongly favor left-aligned, non-splitting challengers like [ta-.tl̪am.ba] above over infixing winners like [t-l̪a-.l̪am.ba]. To eliminate these challengers, I invoke the OCP (Goldsmith 1976, McCarthy 1986; see, e.g., Urbanczyk 1995, Keer 1999) against adjacent identical structures, specifically OCP(SEGMENT), which captures the general cross-linguistic tendency to avoid these sequences.

119) OCP(SEGMENT): assign a violation mark for every sequence of adjacent identical segments.

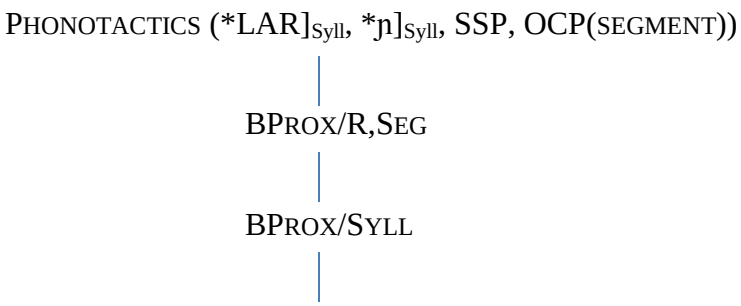
Ranking OCP(SEGMENT) over BPROX/SEG eliminates the internally-reduplicating challengers (115-116) in favor of the winners (117-118), which reduplicate root-initial consonants.

Table 38. OCP(SEGMENT) >> BPROX/SEG

Input	Output	OCP(SEGMENT)	BPROX/SEG
/RED,mavit/	-> mam.vit		**
	mav.vit	*!	*
/RED,hodai/	-> ho.ho.dai		**
	hod.dai	*!	*

I group OCP(SEGMENT) with the other phonotactic constraints relevant to Pima reduplication. Thus the complete ranking for Pima Plural Reduplication in all its forms is as follows (120):

120)



The following master tableau shows how this ranking captures the whole range of subpatterns in Pima Plural Reduplication. This ranking works equally well with both the prefixal or infixal parse of the reduplicative forms, since all of the Position constraints are low-ranking. The ranking of BPROX/SYLL over Position generates the general pattern of Single-C reduplication ([mam.vit]). High-ranked PHONOTACTICS constraints can prevent Single-C reduplication for certain forms, forcing CV-reduplication ([ho.ho.dai]). BPROX/R,SEG picks the second member of a complex onset to be copied, not the first ([tlol.gi], [tla.lam.ba]); its ranking over BPROX/SYLL favors placing the second instance of the copied consonant after the first member of a diphthong ([ku.kua.di]).

Table 39. Pima Plural Reduplication

Input	Output	PHONOTACTICS	BPROX/R,SEG	BPROX/SYLL	POSITION
/RED,mavit/	-> mam.vit		**		*
	ma.ma.vit		**	*!	
	mav.vit	*!	*	*	*
/RED,hodai/	-> ho.ho.dai		**	*	
	hoh.dai	*!	**		*
	hod.dai	*!	*	*	*
/RED,kuadi/	-> ku.kua.di		**	*	
	kuak.di		***!		*
	ku.ka.di		**	*	*!
/RED,tɬogi/	-> tɬoɬ.gi		**		*
	tɬot.gi		***!		*
/RED,tɬamba/	-> tɬa.ɬam.ba		**	*	*
	ta.tɬam.ba		***!	*	

5. Discussion

I have provided analyses of the above patterns of order-disrupting reduplication in Saisiyat and Pima using the constraints BPROX/SEG and BPROX/SYLL. These constraints account for both the minimal size of the copied material, which is often atemplatic, and the placement of the corresponding copies, which disrupts the faithful order of segments. In this section, I compare this analysis to other analyses of minimal, order-disrupting reduplication, and show that this

analysis is superior on both theoretical and empirical grounds. I then explore possible extensions of BPROX constraints to other patterns of reduplication.

5.1 Reduplicant Infixation: Riggle (2006)

Riggle (2006) provides an alternate analysis of Pima Plural reduplication, in which the Reduplicant is always an infix. Infixing of the Reduplicant is demanded by the undominated constraint ANCHOR-V_1, which demands that the initial vowel of the stem be the initial vowel of word. This constraint, which dominates the constraint RED-L that demands prefixing reduplication, forces the Reduplicant to be placed directly to the right of the stem-initial vowel, as in (121-122).

121) /RED,mavit/ -> $[ma]_B[m]_R.[vit]_B$

122) /RED,hodai/ -> $[ho]_B.[ho]_R.[dai]_B$

Riggle accounts for the Reduplicant copying the Base-initial consonant with a subconstraint of BR-MAX indexed to the initial onset, BR-MAX-O_1. The minimal size of the Reduplicant is captured by the economy constraint *STRUC-SYLL, which prevents gratuitous syllables from being added to the reduplicated form. As in the analysis presented here, high-ranking Phonotactics constraints such as *LAR]$_{Syll}$, *ɲ]$_{Syll}$, and the Sonority Sequencing Principle (SSP) are responsible for the distribution of single-C and CV Reduplicants. Evaluating RED-L gradiently accounts for the position of the Reduplicant infix directly after the first member of a diphthong, rather than after the whole diphthong (123).

123) /RED,kuadi/ -> $[ku]_B.[k]_R[ua.di]_B$

Lastly, Riggle accounts for the copying of second member of a complex onset, as in (124-125), with the constraint LOCALITY, which penalizes material intervening between corresponding segments unless that material is itself in B-R correspondence.

124) /RED,tlogi/ -> $[tlo]_B[l]_R.[gi]_B$

125) /RED,tlamba/ -> $[tla]_B.[la]_R[m.ba]_B$

While this analysis successfully accounts for Pima Plural reduplication, it is considerably more complicated than the BPROX analysis. There are three main issues with the infixation analysis. First, the infixation analysis requires an extension of the faithfulness constraints (IO-)Anchor and BR-MAX beyond their usual definitions. In the BPROX analysis, a simple ranking of two BPROX constraints above typical Position-enforcing constraints (PREFIX(RED), BR-CONTIGUITY, LINEARITY) accounts for both the placement and the size of copied material. The infixation analysis requires several constraints to do this work. The position of the copies is enforced by two positional faithfulness constraints, ANCHOR-V_1 and BR-MAX/O_1, which together with the alignment constraint RED-L results in a copy of the word-initial consonant surfacing to the right of the first vowel of the Base. Because of these constraints, Riggle (2006) argues that these faithfulness constraints must be able to be sensitive to strong positions, i.e., they can be positionally faithful, (see Beckman 1998 for positional faithfulness).

Having positionally-faithful versions of these constraints in itself is not necessarily problematic, but it does constitute an extension of the typical formulation of reduplication. While Riggle (2006) does not explicitly propose this, allowing the constraints ANCHOR and BR-MAX to be indexed to any strong positions (i.e., not just the first vowel or the initial onset) predicts a range of infixation of Reduplicants, as well as further patterns of copying. For example, high-ranked

ANCHOR-SYLL$_1$ and BR-MAX-PEAKFOOT might compel a Reduplicant that copies the peak Foot of the word to be infixed after the initial syllable. Whether such an extension of reduplicative constraints is empirically necessary remains to be seen.

The second issue with the infixation analysis involves its use of economy constraints and gradient constraints. The infixation analysis accounts for the minimal size of the Reduplicant using the economy constraint *STRUC-SYLL, which penalizes every syllable in a word. Gouskova (2003) argues that such constraints are theoretically undesirable, as they do not penalized only marked structure in contrast to unmarked structure (e.g., codas vs. open syllables), but all instances of structure that has no unmarked counterpart (e.g., syllables vs. no syllables); only the absence of structure completely satisfies economy constraints. Moreover, Gouskova argues that economy constraints are not only unnecessary to account for effects like syncope, but are outright harmful: for instance, they predict unattested patterns of iambic syncope that targets long vowels. BPROX, on the other hand, accounts for the minimal size and the position of the Reduplicant without appealing to economy.

The infixation account for the position of the Reduplicant infix in forms with diphthongs requires RED-L (the constraint forcing the Reduplicant leftward) to be a gradient constraint, as it favors $[ku]_B.[k]_R[uadi]_B$ over $*[kua]_B[k]_R.[di]_B$. McCarthy (2003) and Gouskova (2003) both argue against gradient constraints, i.e., constraints that must be able to count potentially infinitely. The BPROX analysis uses categorical constraints that force leftwardness of the Reduplicant: PREFIX(RED) and PREFIX/SYLL(RED) (McCarthy 2003). PREFIX(RED) forces true prefixation, while PREFIX/SYLL(RED) prevents the Reduplicant from being infixed after the first syllable. In the BPROX analysis, PREFIX(RED) is sometimes crucially violated (depending on the parsing of the reduplicated form), but PREFIX/SYLL(RED) is never violated, which prevents deep infixation.

The infixation analysis cannot use PREFIX/SYLL(RED) to prevent deep infixation, as it violated by $[ku]_B.[k]_R[uadi]_B$ but not $*[kua]_B[k]_R.[di]_B$, and PREFIX(RED) is always violated. Thus the infixation analysis crucially must rely on gradiently evaluating leftwardness, counting segments, to account for this pattern.

The third issue is one of parsimony: while the infixation analysis uses the other types of constraints above to account for Pima reduplication, it also must use the notion of proximity. The constraint LOCALITY that Riggle uses in fact is very similar to BPROX: both constraints penalize certain intervening material between correspondents and thus both account for the copying of the second member of a complex onset rather than the first. Since the infixation analysis must use this distance-evaluating constraint as well as positionally-indexed reduplication constraints and an economy constraint, this analysis is more complicated than the BPROX analysis, which dispenses with the need for the latter two types of constraints. The BPROX analysis, in other words, can do the same amount of work as the infixation analysis with fewer constraints: BPROX by itself accounts for the minimality, the position, and the content of the Reduplicant. Moreover, this analysis does not crucially demand that the Reduplicant be infixed, but is compatible with both a prefixing and infixing parsing, unlike the infixation analysis. Therefore the BPROX analysis is the more elegant one, while arguably remaining the more conservative.

5.2 ADJACENCYBR: Lunden (2004)

Lunden (2004) provides an analysis of similarly minimal and order-disrupting reduplicative patterns in several languages, including the Austronesian languages Dobel, Agta, and Marshallese. She uses a family of ADJACENCYBR constraints that demand that material in the Reduplicant is adjacent to its correspondent in the Base. These constraints specifically include

ADJACENCYBR-BY-SEGMENT, which demands that every segment in the Reduplicant be next to its correspondent Base segment, as well as ADJACENCYBR-BY-SYLLABLE and ADJACENCYBR-BY-FOOT, which make similar demands on Reduplicant syllables and Feet, respectively. ADJACENCY-BR constraints account for several different phenomena in reduplication: Marantz's Generalization, minimal size, discontiguity, and templatic effects. The ability to account simultaneously for both the position and size of copied material is similar to BPROX.

Because ADJACENCYBR constraints only demand that material be adjacent, however, they cannot account for the single-C copying in Saisiyat and Pima. In these patterns (126-127), the Reduplicant ([k], [m]) is not segment-adjacent to its correspondent in the Base: the vowel of the Base intervenes between the two copies. Thus these forms violate ADJACENCYBR-BY-SEGMENT.

126) /om,RED,kitaʔ/ -> [k-o-k-.m-i.taʔ]

127) /RED,mavit/ -> [m-am.vit], [ma-m-.vit]

Challengers with CV Reduplicants (128-129) also violate ADJACENCYBR-BY-SEGMENT, but do not violate ADJACENCYBR-BY-SYLLABLE, because the Reduplicant syllable ([ko], [ma]) is adjacent to its Base correspondent.

128) [ko-.k-o.mi.taʔ]

129) [ma-.ma.vit]

ADJACENCYBR thus cannot distinguish between the single-C winners and the CV challengers; in fact, on other positional constraints (e.g., LINEARITY, CONTIGUITY, PREFIX(RED)), the CV challengers fare better than the winners. Thus, while the ADJACENCYBR analysis has similar theoretical benefits as the BPROX analysis, it crucially cannot account for the Saisiyat and Pima

data. The BPROX analysis therefore does better than the ADJACENCYBR analysis in terms of empirical coverage.

5.3 Future Directions for BPROX

There are several future directions for BPROX constraints. First, while BPROX can be sensitive to segmental and syllable boundaries, it is not certain whether they can be sensitive to any other types of boundary. For example, are BPROX/FOOT constraints ever necessary to account for certain reduplicative patterns? I have shown that both BPROX/SEG and BPROX/SYLL constraints are empirically necessary, but it is unclear at this point whether this is true of BPROX/FOOT constraints.

Another issue is the strong prediction made by BPROX/SEG that reduplication can be realized as gemination or length in some languages. Since BPROX/SEG penalizes any segmental boundaries between correspondents, it can only be satisfied by the coalescence of the Reduplicant and its Base correspondent into a long or geminate segment, where there is no intervening segmental boundary. Several languages do express morphemes through gemination of a root segment (e.g., in the verbal morphology of Semitic languages), so this is not entirely implausible. An analysis of such morphological gemination as reduplication using BPROX/SEG is thus necessary.

A further theoretical issue is whether BPROX constraints can account for templatic effects, such as syllable-sized Reduplicants. Since Lunden's (2004) ADJACENCYBR constraints, which do similar work to BPROX constraints, can account for templatic effects, it seems likely that BPROX constraints could do so as well. Saisiyat, for instance, has a pattern of syllable-size reduplication which prefixes heavy syllables when possible and light syllables when phonotactics prevent a heavy syllable from being copied. To compel the copying of a full syllable's worth of material, a

constraint must dominate BPROX/SYLL that would force the introduction of an intervening syllable boundary. A constraint demands that this particular Reduplicant must dominate a syllable head (e.g., nucleus) could be useful here: if a nucleus is copied, then another syllable boundary must be introduced. Such constraints have been proposed, such as MAX-MP$_{\text{SYLL-HEAD}}$ (Walker 2002). Ranking BPROX/L,SYLL over BR-MAX would limit the Reduplicant to a single syllable, while ranking BR-MAX over BPROX/R,SYLL would allow the onset of a Base-internal syllable to be copied to create a coda for the Reduplicant. This is a rough sketch, but it offers a tentative idea of how BPROX constraints could account for templatic reduplication.

6. Conclusion

In this paper I have argued for an account of patterns of minimal and order-disrupting reduplication in the unrelated languages Saisiyat and Pima using BOUNDARY-PROXIMITY (BPROX) constraints. BPROX constraints penalize phonological boundaries intervening between the edges of a Reduplicant and their Base correspondents. This is a reformulation of Kennedy's (2005) definition of PROXIMITY needed to avoid a pathology of unattested splitting reduplication, in which Reduplicants are ideally split into smaller and smaller parts. The reformulation of BPROX that I argue for predicts a narrower ranger of order-disruption in reduplication. BPROX so defined accounts for such order disrupting in the above reduplicative patterns. Thus I conclude on both empirical and theoretical grounds that BPROX constraints provide the best analysis of attested patterns of minimal and order-disrupting reduplication.

REFERENCES

Alderete, John, Jill Beckman, Laura Benua, Amalia Gnanadesikan, John J. McCarthy & Suzanne Urbanczyk. 1999. Reduplication with fixed segmentism. Linguistic Inquiry (30). 327–364.

Beckman, Jill. 1998. Positional faithfulness, Ph.D. dissertation, University of Massachusetts at Amherst.

Benua, Laura. 1997. Transderivational identity: Phonological relations between words, Ph.D. dissertation, University of Massachusetts at Amherst. ROA-259.

Flack, Kathryn. 2009. Constraints on onsets and codas of words and phrases. Phonology 26: 269-302.

Gafos, Adiamantos. 1996. The articulatory basis of locality in phonology. Baltimore, MD: Johns Hopkins University dissertation. [Published, New York: Garland, 1999]

Gafos, Adamantios. 1998. A-templatic reduplication. Linguistic Inquiry (29). 515–527.

Goldsmith, John A. 1976. Autosegmental phonology. Ph.D. dissertation, MIT.

Goodenough, W., & H. Sugita. 1980. Trukese-English dictionary.

Gouskova, M. 2003. Deriving economy: Syncope in optimality theory. Ph.D. dissertation, University of Massachusetts at Amherst. ROA-610.

Inkelas, Sharon & Cheryl Zoll. 2005. Reduplication: Doubling in morphology. Cambridge: Cambridge University Press.

Kager, Rene. 1999. Optimality theory. Cambridge: Cambridge University Press.

Keer, E. 1999. Geminates, the OCP, and the Nature of CON. Ph.D. dissertation, Rutgers University. ROA-350.

Kennedy, Robert. 2005. A formal replacement for reduplicative anchoring. Unpublished Ms., http://www.linguistics.ucsb.edu/faculty/rkennedy/papers/kennedy_proximity.pdf.

Kenstowicz, Michael. 2005. Paradigmatic Uniformity and Contrast. In Laura J. Downing, T.A.Hall, and Renate Raffelsiefen (eds.), Paradigms in Phonological Theory, 145-169. Oxford: Oxford University Press.

Lunden, S.L. Anya. 2004. Reduplicant Placement, Anchoring and Locality. Unpublished Ms., University of California, Santa Cruz.

Marantz, Alec. 1982. Re reduplication. Linguistic Inquiry (13). 435–482.

McCarthy, John J. 1986. OCP effects: Gemination and antigemination. Linguistic Inquiry (17). 207–263.

McCarthy, John. 2003. OT constraints are categorical. Phonology 20: 75-138.

McCarthy, John J., and Alan Prince. 1993. Prosodic Morphology I: Constraint interaction and satisfaction. Ms., University of Massachusetts, Amherst, and Rutgers University, New Brunswick, N.J.

McCarthy, John J. & Alan Prince. 1994a. Two lectures on prosodic morphology. Unpublished Ms., University of Massachusetts, Amherst & Rutgers University (ROA-59).

McCarthy, John J. & Alan Prince. 1994b. The emergence of the unmarked: Optimality in prosodic morphology. Unpublished Ms., University of Massachusetts, Amherst & Rutgers University (ROA-13).

McCarthy, John J. & Alan Prince. 1995. Faithfulness and reduplicative identity. In Jill N. Beckman, Laura Walsh Dickey & Suzanne Urbanczyk (eds.) Papers in Optimality Theory, 249–384. Amherst: GLSA.

Mester, Armin., & Jane Padgett. 1994. Direct syllabification in generalized alignment. Phonology at Santa Cruz, 3, 79–85.

Nelson, Nicole. 2003. Asymmetric anchoring. Ph.D. Dissertation, Rutgers University.

Nuger, Justin. 2006. Discontiguous Reduplication. ROA 831-0506.

Odden, David. 1994. Adjacency parameters in phonology. Language 70.289-330.

Prince, Alan. 1998. Two lectures on Optimality Theory. Handout from Phonology Forum 1998, Kobe University.

Prince, Alan & Paul Smolensky. 1993/2004. Optimality Theory: Constraint interaction in generative grammar. Unpublished Ms., Rutgers University & University of Colorado, Boulder. Published 2004, Malden, MA & Oxford: Blackwell.

Raimy, Eric, & William Idsardi. 1997. A minimalist approach to reduplication in optimality theory. In K. Kusumoto (Ed.), Proceedings of the North Eastern Linguistics Society 27 (pp. 369–382). Amherst: GLSA.

Riggle, Jason. (2004). Nonlocal reduplication. In K. Moulton, & M. Wolf (Eds.), Proceedings of the 34th annual meeting of the North Eastern Linguistic Society (pp. 485–496). Amherst: GLSA.

Riggle, Jason: 2006, 'Infixing Reduplication in Pima and its Theoretical Consequences', Natural Language and Linguistic Theory 24/3, 857-891

Rose, Sharon. 2000. Rethinking geminates, long-distance geminates and the OCP. Linguistic Inquiry 31.85-122.

Rose, Sharon and Rachel Walker. 2004. A typology of consonant agreement as correspondence. Language, 80, 475–531.

Spaelti, Philip. 1997. Dimensions of variation in multi-pattern reduplication. Ph.D. dissertation, University of California, Santa Cruz.

Struijke, Caro. 2000. Existential faithfulness: A study of reduplicative TETU, feature movement, and dissimilation. Ph.D. dissertation, University of Maryland, College Park.

Suzuki, Keiichiro. 1998. A typological investigation of dissimilation . Tuscon, AZ: University of Arizona dissertation

Urbanczyk, Suzanne. 1995. Double reduplications in parallel. In J. Beckman, L.Walsh Dickey & S. Urbanczyk (Eds.), University of Massachusetts occasional papers in linguistics 18: Papers in optimality theory (pp. 499–532). Amherst: GLSA.

Urbanczyk, Suzanne. 1996. Patterns of reduplication in Lushootseed. Ph.D. dissertation, University of Massachusetts at Amherst.

Urbanczyk, Suzanne. 2006. Reduplicative form and the root-affix asymmetry. Natural Language and Linguistic Theory (24) . 179–240.

Walker, Rachel. 2000. Nasal reduplication in Mbe affixation. Phonology 17: 65-115.

Walker, Rachel. 2002. Yuhup prosodic morphology and a case of augmentation. Proceedings of NELS 32, ed. by Masako Hirotani, pp. 551-562. University of Massachusetts, Amherst. GLSA.

Walker, Rachel and Bella Feng. 2004. A ternary model of morphology-phonology correspondence. Proceedings of WCCFL 23, ed. by Vineeta Chand, Ann Kelleher, Angelo J. Rodríguez, and Benjamin Schmeiser, pp. 773-786.

Wolf, Matthew. 2008. Optimal Interleaving: Serial Phonology-Morphology Interaction in a Constraint-Based Model. PhD dissertation. University of Massachusetts Amherst.

Zeitoun, Elizabeth and Chen-huei Wu. 2005. Saisiyat reduplication revisited. Concentric 31.2:31-56.

Zoll, C. (1994). Subsegmental parsing: Floating features in Chaha and Yawelmani. In J. Merchant, J. Padgett & R. Walker (Eds.), Phonology at Santa Cruz (Vol. 3) (pp. 47–56).

Zuraw, Kie. 2002. Aggressive reduplication. Phonology (19) . 395–439.