Contiguity in Tawala Reduplication

Sam Zukoff, Leipzig University
samzukoff@gmail.com · www.samzukoff.com

mfm 28 · Manchester, UK · May 26–28, 2021

1 Introduction

- Hicks Kennard (2004) [HK] develops an atemplatic analysis of reduplication in Tawala (Papua New Guinea, Oceanic, Austronesian; Ezard 1997), set within Base-Reduplicant Correspondence Theory (BRCT; McCarthy & Prince 1995).

⋆ Today, I will show that HK's analysis needs to be revised slightly in order to account for two small problems:

(1) An unconsidered alternative candidate for one of the consonant-initial patterns motivates a more articulated version of **Contiguity-BR** (McCarthy & Prince 1995), which relativizes violations to consonants and vowels.

(2) The treatment of repeated identical syllables requires restricting the relevant constraint to **word-initial position**.

→ These changes have the overall effect of re-characterizing the system as targeting *minimal reduplication* (e.g. Hendricks 1999, Zukoff 2016; cf. Saba Kirchner 2010, 2013), rather than a foot-sized reduplicant.

- I conclude by considering the ramifications of this approach for what would otherwise appear to be gradient Contiguity violations.

2 Data preview

- The Tawala durative exhibits four distinct reduplicant shapes, whose distribution is phonologically predictable (see HK, drawing on Ezard 1980, 1997; see also Haugen & Hicks Kennard 2011 [HHK]):

(3) *Reduplicant-shape alternations in Tawala*

Type A: $C_1V_1.V_2$-initial bases reduplicate as C_1V_2
Type B: $C_1V_1.C_2V_2$-initial bases reduplicate that whole string
Type C: VC-initial bases reduplicate VC-
Type D: Roots beginning in a repeated CV sequence "reduplicate" by doubling the first root vowel[1]

- These patterns are schematized and exemplified in (4):

(4) *Tawala reduplicant shapes by base type*

	Base shape	Red. shape	Example forms			
A.	$C_1V_1.V_2X$	→ C_1V_2-	e.g. *be.i.ha*	→ <u>*bi*</u>-*be.i.ha*	'search/be searching'	[HK:312]
B.	$C_1V_1.C_2V_2X$	→ $C_1V_1.C_2V_2$-	e.g. *hu.ne.ya*	→ <u>*hu.ne*</u>-*hu.ne.ya*	'praise/be praising'	[HK:307]
C.	V_1C_1X	→ $V_1.C_1$-	e.g. *a.tu.na*	→ <u>*a.t*</u>-*a.tu.na*	'rain/be raining'	[HHK:12]
D.	$C_1V_1.C_1V_1X$	→ *V_1-doubling*	e.g. *gu.gu.ya*	→ *g*-<u>*u*</u>.-*u.gu.ya*	'preach/be preaching'	[HK:305]

[1] It is not completely clear whether the output of the doubling in Type D is a single long vowel [V:] or a sequence of heterosyllabic short vowels [V.V].

3 Consonant-initial roots: Type A & Type B reduplication

→ I will first focus on Types A and B, because HK's analysis turns out to not quite work for them.

- Type B is the most common root shape, and has many examples (5). Many of these have morphologically complex bases, but this does not appear to have any impact on reduplication.

(5) *More examples of Type B reduplication* (Ezard 1980:147; cf. Ezard 1997:41)

Simplex		**Reduplicated (durative)**	
hopu	→	*hopu-hopu*	'to go down'
geleta	→	*gele-geleta*	'to arrive'
hune-ya	→	*hune-hune-ya*	'to praise (tr.)'
kima-ya	→	*kima-kima-ya*	'to bite (tr.)'
paliwele-ya	→	*pali-paliwele-ya*	'to speak to someone'
hanahaya	→	*hana-hanahaya*	'to bite'
bahanae	→	*baha-bahanae*	'to speak' ("talk-go")
kawamoina	→	*kawa-kawamoina*	'to proclaim true' ("proclaim-true")
nugotuhu	→	*nugo-nugotuhu*	'to think' ("mind-#")
hinimaya	→	*hini-hinimaya*	'to be ashamed' ("skin-feel")
menamaga	→	*mena-menamage*	'to be two-faced' ("tongue-many")
lupahopu	→	*lupa-lupahopu*	'to jump down' ("jump-down")

- There are much fewer CVV-initial roots. The primary pattern is Type A reduplication (6), but it must be admitted that there are just as many "exceptions" (7). I analyze only the Type A forms.

(6) *More examples of Type A reduplication* (Ezard 1980:147; cf. Ezard 1997:43)

Simplex		**Reduplicated (durative)**	
ga.e	→	*ge-ga.e*	'to go up'
ho.u.ni	→	*hu-ho.u.ni*	'to put it'
be.i.ha	→	*bi-be.i.ha*	'to search'
to.u	→	*tu-to.u*	'to weep'
wa.o	→	*wo-wa.o*	'to dig a hole for planting'

(7) *Other reduplication patterns for CVV roots* (Ezard 1980:147, Ezard 1997:43)

Simplex		**Reduplicated (durative)**	
C_1V_1-reduplication			
ne.i	→	*ne-ne.i*	'to come'
ge.i	→	*ge-ge.i*	'to come up'
—	→	*ko-ko.e*	'to finish'
$C_1V_1V_2$-reduplication			
ho.e-ya	→	*ho.e-ho.e-ya*	'to open (tr.)'
bu.i	→	*bu.i-bu.i*	'to turn over'
wo.e	→	*wo.e-wo.e*	'to paddle'
C_1i-reduplication			
pe.u	→	*pi-pe.u*	'to fall'

3.1 HK's analysis

- HK derives Type A primarily through the operation of two constraints:

- The first constraint is *Repeat (8), which bans adjacent identical syllables.
  → I am going to use a more specific version of this constraint, *Repeat(initial) (9), for reasons which will become clear below.

(8) ***Repeat:*** Assign one violation * for each pair of adjacent identical syllables. (HK:310; cf. Yip 1995)

(9) ***Repeat(initial):*** Assign one violation * for each *word-initial* pair of adjacent identical syllables.

- The other constraint is Align-Root-L (10), which prefers minimizing the length of the reduplicant.

(10) **Align-Root-L:** Assign one violation * for each segment which intervenes between the left edge of the root and the left edge of the word. (HK:309; cf. McCarthy & Prince 1993a, Hendricks 1999)

- These constraints must outrank Contiguity-BR (11), the constraint requiring contiguous copying, in order to generate Type A's discontiguous copying (12d).

(11) **Contiguity-BR** ("*Don't skip–BR*"): (HK:308; cf. McCarthy & Prince 1995)
a. Assign one violation * if the reduplicant doesn't correspond to a contiguous substring of the base.
b. For a reduplicant string $r_1 \ldots r_n$ standing in correspondence with a base string $b_1 \ldots b_n$, assign one violation * for each segment between b_1 and b_n which lacks a correspondent in $r_1 \ldots r_n$.

(12) *Type A reduplication: CV.V bases*

/red, beiha/			*Repeat(init)	Align-Root-L	Contiguity-BR
a.		be.i.ha-be.i.ha		5!	
b.		be.i-be.i.ha		3!	
c.		be-be.i.ha	*!	2	
d.	☞	bi-be.i.ha		2	*

- Align-Root-L prefers short reduplicants, favoring (12c,d) over (12a,b).
- (12c) incurs a *Repeat(init) violation for its initial repetition ($[\text{be}]_\sigma[\text{be}]_\sigma$), while (12d) incurs a Contig-BR violation because [bi] is not a contiguous substring of the base.

→ As long as *Repeat(init) outranks Contiguity-BR, we correctly select (12d).

⋆ However, this ranking wrongly predicts discontiguous copying also for Type B, i.e. candidate (13d), which was not considered by HK (nor by Haugen & Hicks Kennard 2011).

(13) *Type B reduplication: CV.CV bases*

/red, huneya/			*Repeat(init)	Align-Root-L	Contiguity-BR
a.		hu.ne.ya-hu.ne.ya		6!	
b.	☹	hu.ne-hu.ne.ya		4!	
c.		hu-hu.ne.ya	*!	2	
d.	💣	he-hu.ne.ya		2	*(*)

→ With the current constraints, there should be no difference in the constraint interaction, so we should continue to select the C_1V_2- candidate (13d).
  - But do note that we have a potential difference with respect to Contiguity: if we adopt (some version of) the "gradient" definition (11b), we would have one more violation in (13d) than (12d)...

3.2 My proposed fix: relativized CONTIGUITY

- The two patterns are distinguished by the nature of their (would-be) discontiguity:

(14) a. Type A skips only vowels (base V_1): *bi-b*[*e*]*.i.ha*
b. For Type B, the problematic discontiguous candidate (13d) also skips a consonant (base C_2) in addition to a vowel (base V_1): **he-h*[*u.n*]*e.ya*

→ We can take advantage of this distinction if we relativize CONTIG-BR to consonants (15) and vowels (16):

(15) **CONTIGUITYC-B(→)R** ("*Don't skip C's–BR*"):
For a reduplicant string $r_1 \ldots r_n$ standing in correspondence with a base string $b_1 \ldots b_n$, assign one violation * for each **consonant** between b_1 and b_n which lacks a correspondent in $r_1 \ldots r_n$.

(16) **CONTIGUITYV-B(→)R** ("*Don't skip V's–BR*"):
For a reduplicant string $r_1 \ldots r_n$ standing in correspondence with a base string $b_1 \ldots b_n$, assign one violation * for each **vowel** between b_1 and b_n which lacks a correspondent in $r_1 \ldots r_n$.

- If we sandwich the size restrictor constraint ALIGN-ROOT-L between the relativized CONTIG constraints as shown in (17), we derive the right results (18, 19).

(17) **Ranking:** CONTIGC-BR ≫ ALIGN-ROOT-L ≫ CONTIGV-BR

- Splitting up CONTIG has no effect on Type A (because there's no medial consonant to skip), so the evaluation looks exactly the same as before:

(18) *Type A reduplication with relativized* CONTIGUITY

/RED, beiha/	*REPEAT(init)	CONTIGC-BR	ALN-RT-L	CONTIGV-BR
a. be.i.ha-be.i.ha			5!	
b. be.i-be.i.ha			3!	
c. be-be.i.ha	*!		2	
d. ☞ bi-be.i.ha			2	*

→ But for Type B, the new high-ranked CONTIGC-BR can rule out the discontiguous copying candidate because its discontiguity includes a consonant (unlike Type A).

- *REPEAT(init) and CONTIGC-BR now rule out both minimal copying candidates (19c,d), and so the next shortest possible (C-contiguous) reduplicant (19b) is selected.

(19) *Type B reduplication with relativized* CONTIGUITY

/RED, huneya/	*REPEAT(init)	CONTIGC-BR	ALN-RT-L	CONTIGV-BR
a. hu.ne.ya-hu.ne.ya			6!	
b. ☞ hu.ne-hu.ne.ya			4	
c. hu-hu.ne.ya	*!		2	
d. he-hu.ne.ya		*!	2	*
e. hu.e-hu.ne.ya		*!	3	

- You can't finagle a short reduplicant by copying a non-initial syllable because ANCHOR-L-BR (20), which requires copying from the left edge, outranks ALIGN-ROOT-L (21).

(20) **ANCHOR-L-BR:** Assign one violation * if the leftmost segment of the reduplicant does not correspond to the leftmost segment of the base. (HK:307; cf. McCarthy & Prince 1995, Shaw 2005)

(21) *Type B reduplication and* ANCHOR-L-BR

/RED, huneya/	ANCHOR-L-BR	ALIGN-ROOT-L
a. ☞ hu.ne-hu.ne.ya		4
b. ne-hu.ne.ya	*!	2
c. u-hu.ne.ya	*!	1

3.3 A note on stress and feet

- HK includes a constraint requiring a left-aligned foot in reduplicated words.
  - She asserts that this helps generates disyllabic copying in Type B.
  - But, as can be seen from my revised analysis, this is not necessary. (And I don't think it actually follows from her account to begin with.)

⋆ The way in which it still may be relevant is that reduplicated words do show a strange stress pattern.

- Tawala's stress pattern is as follows (Ezard 1997:44–45, HK:305–306):

(22)	a.	Primary stress on the penult	[bá.da] 'man', [te.wá.la] 'child'
	b.	Secondary stress on alternating syllables to the left	[kè.du.lú.ma] 'woman'

- The one exception to (22b) is found in reduplication: there is a requirement that the initial syllable of the reduplicant be stressed, even if this leads to a lapse (23a–c) or a clash (23d).

(23) *Stress in reduplication* (Ezard 1997:44, HK:306–307)

a.	i-**dè**.wa-de.wá.ya	(*i-de.**wà**-de.wá.ya)	'he/she/it is doing it'
b.	ina-**bù**.li-bu.lì.li.má.i	(*ina-bu.**lì**-bu.lì.li.má.i)	'he/she/it will be running here'
c.	**kà**.da-ka.dá.u	(*ka.**dà**-ka.dá.u)	'be traveling'
d.	**à**.p-á.pu	(***a**.p-á.pu)	'be baking'

- While this could surely be accounted for with feet (as long as it interacts in the right with copying), we can do this straightforwardly with the following foot-free stress constraints (cf. Zukoff 2021 on Ponapean):

(24) *Stress constraints*[2]

a. **StressL-Red:** Assign one violation * if the reduplicant-initial syllable is unstressed.
b. ***Clash:** Assign one violation * if for each pair of adjacent stressed syllables.
c. ***Lapse:** Assign one violation * if for each pair of adjacent unstressed syllables.

- If StressL-Red outranks the constraints on alternating rhythm, we will generate clashes and lapses just in reduplication:

(25) *Clashes in reduplication*

/red, apu/				StressL-Red	Align-Root-L	*Clash	*Lapse
a.	☞	à.p-á.pu	[2-10]		2	*	
b.		a.p-á.pu	[0-10]	*!	2		
c.		à.pu-á.pu	[20-10]		3!		

(26) *Lapses in reduplication*

/red, dewaya/				StressL-Red	Align-Root-L	*Clash	*Lapse
a.	☞	dè.wa-de.wá.ya	[20-010]		4		*
b.		de.wà-de.wá.ya	[02-010]	*!	4		
c.		dè.wa.yà-de.wá.ya	[202-010]		6!		

- But note from losing candidates (25c) and (26c), where the reduplicant is extended to avoid the stress problems, that the stress constraints have no impact on what gets copied.

→ This reiterates that the languages is *not* treating the (binary) foot as the target shape in reduplication.

[2] We can derive fixed penultimate stress with high-ranked NonFinality (*don't stress the final syllable*) and *LapseRight (*don't have two unstressed syllables at the right edge*). The one exception to penultimate primary stress (Ezard 1997:45) is when the antepenult has a lower vowel than the penult and the penult is onsetless or has an onset [l]. Whatever the right formulation of the markedness constraint based on vowel height, that constraint outranks *LapseRight, allowing for stress retraction to the antepenult. Because of this stress pattern, we get a clash also in [bì-bé.i.ha].

4 Vowel-initial roots: Type C reduplication

- The Type C pattern, where vowel-initial roots copy their initial VC- string, exemplified further in (27), follows completely from existing rankings, as shown in (28).

(27) *More examples of Type B reduplication* (Ezard 1980:147; cf. Ezard 1997:42)

Simplex		**Reduplicated (durative)**	
a.pu	→	*a.p-a.pu*	'to bake'
e.no	→	*e.n-e.no*	'to sleep'
a.m̩	→	*a.m-a.m̩*	'to eat'
u.ma	→	*u.m-u.ma*	'to drink'
a.tu.na	→	*a.t-a.tu.na*	'to rain'
o.to.wi	→	*o.t-o.to.wi*	'to make an appointment'

(28) *Type C reduplication: VC-copying*

	/RED, atuna/	*REPEAT(init)	ANCHOR-L-BR	ALIGN-ROOT-L
a.	a.tu.-a.tu.na			3!
b. ☞	a.t-a.tu.na			2
c.	a.-a.tu.na	*!		1
d.	t-a.tu.na		*!	1

∗ With these constraints, ONSET (29) turns out to be unnecessary, even though we might have expected it to be responsible for eliminating (28a) and (28c), as it is in HK's analysis.

(29) **ONSET:** Assign one violation * for each onsetless syllable. (HK:306; cf. Itô 1989, Prince & Smolensky 2004)

→ Instead, *REPEAT(init) can do the job.

5 Type D reduplication, *REPEAT, and TETU

- We've now used *REPEAT (or the more specific *REPEAT(init)) to account for:

(30) a. The lack of C_1V_1- reduplication in consonant-initial roots (Types A & B), and
b. The lack of V_1- reduplication for vowel-initial roots (Type C).

- HK (and HHK) also uses this constraint to help analyze Type D reduplication, where V_1 doubles inside the base just in case the base begins in two identical syllables.
- The attested instances of this pattern are shown in (31):

(31) *More examples of Type D reduplication* (Ezard 1980:148, Ezard 1997:44, HK:305)

Simplex		**Reduplicated (durative)**	
gu.gu.ya	→	*gu.u.gu.ya*	'preach/be preaching'
to.to.go	→	*to.o.to.go*	'be sick/be being sick'
ta.ta.wa	→	*ta.a.ta.wa*	'tremble/be trembling'
te.te	→	*te.e.te*	'cross/be crossing (a bridge)'
ki.ki	→	*ki.i.ki*	'strangle/be strangling'

5.1 *Repeat outside of reduplication

- However, *Repeat (both the specific and the more general version) is freely violated outside of reduplication (HHK:24–26), including within roots, across compound boundaries, and at other base-affix junctures.
  - This is illustrated for the root /totogo/ → [to.to.go] 'be sick' (Ezard 1997:33, HK:305) in (32):

(32) *Repeat(init) *violations permitted outside of reduplication*

/red, totogo/		Max-IO	Dep-IO	Ident-IO	*Repeat(init)
a.	☞ to.to.go				*
b.	to.t**i**.go			*!	
c.	to.*pa*.to.go		*!*		
d.	to.go	*!*			

⋆ This means that the avoidance of repeated identical (initial) syllables in reduplication in Tawala is an instance of *the emergence of the unmarked* (TETU; McCarthy & Prince 1994), as argued by HK & HHK.

5.2 Type D reduplication

- HK & HHK analyze the Type D vowel-doubling pattern as an extreme instantiation of TETU: the reduplicant surfaces as an infixed copy of base-V_1 in order to break up the root's repeated syllables.
  → Infixal reduplication provides a unique way to satisfy *Repeat that is not available in non-reduplicative constructions (via violation of Align-Red-L (33)).

(33) **Align-Red-L:** Assign one violation * for each segment which intervenes between the left edge of the reduplicant and the left edge of the word. (HK:307)

(34) *Type D reduplication: V-doubling*

/red, $g^1u^2g^3u^4y^5a^6$/		Faith-IO	*Repeat (init)	Anch-L-BR	Aln-Red-L	Aln-Rt-L	*Repeat
a.i.	$\underline{g^1u^2.g^3i^4}$.-$g^1u^2.g^3i^4.y^5a^6$	*!				4	
a.ii.	$\underline{u^2.g^3}$-$u^2.g^3u^4.y^5a^6$	*!				2	*
b.i.	$\underline{g^1u^2.g^3u^4}$.-$g^1u^2.g^3u^4.y^5a^6$		*!			4	***
b.ii.	$\underline{g^1u^2}$.-$g^1u^2.g^3u^4.y^5a^6$		*!			2	**
b.iii.	$\underline{g^1u^4}$.-$g^1u^2.g^3u^4.y^5a^6$		*!			2	**
c.i.	$\underline{u^2}$.-$g^1u^2.g^3u^4.y^5a^6$			*!		1	*
c.ii.	$\underline{y^5a^6}$.-$g^1u^2.g^3u^4.y^5a^6$			*!		2	*
c.iii.	g^1-$\underline{u^4.y^5}$-$u^2.g^3u^4.y^5a^6$			*!	1		
d.i. ☞	g^1-$\underline{u^2}$-.$u^2.g^3u^4.y^5a^6$				1		
d.ii.	g^1-$\underline{u^2.g^3}$-$u^2.g^3u^4.y^5a^6$		*!		1		**

Highlights:

- Trying to do Type B reduplication (34b.i) obvious gives you *Repeat(init) violation (in fact three *Repeat violations for four identical syllables in a row).
- Trying to do Type A reduplication (34b.iii) doesn't even fix the *Repeat(init) violation in this case, because $V_1 = V_2$. (It also picks up a ContigC-BR violation, as would a candidate *[g^1-$\underline{u^2.y^5}$-$u^2.g^3u^4.y^5a^6$].)
- You could turn it into Type C reduplication (34a.ii) by deleting the root-initial consonant, but this fatally violates Faith-IO.
- The (c) candidates all violate Anchor-L-BR because they start with a copy of something other than the base segment immediately to their right.[3]

[3] In order to prefer desired candidate (34d.i) over candidates (34c.i) and (34c.ii), it must be the case that the base of reduplication initiates with the immediately following segment (e.g. McCarthy & Prince 1993b, Urbanczyk 1996). If the base instead comprised the entire non-reduplicative string (Lunden 2004) or some other constituent (Shaw 2005, Haugen 2009), the two candidates would have an equivalent violation profile on the current constraints (i.e. both violating Anchor-L-BR), and the tie would be broken, wrongly in favor of (34c.i), by lower-ranked Onset, or (34c.ii), by Max-BR.

→ As long as we have the crucial rankings in (35), we correctly derive candidate (34d.i) $[g^1\text{-}\underline{u^2}.\text{-}u^2.g^3u^4.y^5a^6]$, which infixes a copy of V_1 between base-C_1 and base-V_1.

- ∗ (Some version of) CONTIGUITY-IO must also be low-ranked, since the base no longer corresponds to a contiguous string in the input.

(35) **Ranking:** *REPEAT(init), ANCHOR-L-BR ≫ ALIGN-RED-L

- This type of infixation is the only way to avoid the *REPEAT(init) violation without modifying the base, and so it is selected, even though it results in a non-canonical reduplicant shape and reduplicant position.[4]

5.3 Why not general *REPEAT?

- As can be verified from the tableau in (34), the general *REPEAT constraint, if ranked in the position of the more specific *REPEAT(init), would be sufficient to select the correct output.
- The reason we need the more specific *REPEAT(init) is because we *do* find non-initial repetitions in reduplicated forms, something which would not be predicted by high-ranked *REPEAT:

(36) *Predictions about V-doubling for the different *REPEAT's* (X = at least one segment)

a. ***REPEAT(init):** V-doubling infixation to avoid *a word-initial* repetition
✓ $/\#C_1V_1C_2V_2(X)/ \rightarrow [\#C_1\text{-}\underline{V_1}\text{-}V_1C_2V_2(X)]$, ✗ $/XC_1V_1C_2V_2(X)/ \rightarrow [XC_1\text{-}\underline{V_1}\text{-}V_1C_2V_2(X)]$

b. ***REPEAT:** V-doubling infixation to avoid *any* repetition
✓ $/\#C_1V_1C_2V_2(X)/ \rightarrow [\#C_1\text{-}\underline{V_1}\text{-}V_1C_2V_2(X)]$, ✓ $/XC_1V_1C_2V_2(X)/ \rightarrow [XC_1\text{-}\underline{V_1}\text{-}V_1C_2V_2(X)]$

- ∗ This is what McCarthy (2003), Yu (2007) would call "hyper-infixation".

- There is at least one relevant base which can disambiguate between these two predictions, in favor of *REPEAT(init) (i.e prediction (36a)):

(37) *kilolo* 'urinate' → *<u>kilo</u>-kilolo* 'urinating' (**kil-<u>o</u>-o-lo*) [Ezard 1997:61, HK:307]

- We can see the argument most clearly if we try getting rid of *REPEAT(init):

- Since the reduplicant is by default a prefix, we know that ALIGN-RED-L ≫ ALIGN-ROOT-L.
- From the Type D cases, we know that some version of *REPEAT must dominate ALIGN-RED-L.

→ Implementing these rankings, we incorrectly derive infixation (38c):

(38) *Can't derive* /kilolo/ → [<u>kilo</u>-kilolo] *(Type B reduplication) without* *REPEAT(init)

/RED, $k^1i^2l^3o^4l^5o^6$/	*REPEAT	ALIGN-RED-L	ALIGN-ROOT-L
a. ☹ $\underline{k^1i^2.l^3o^4}.\text{-}k^1i^2.l^3o^4.l^5o^6$	*!		4
b. $\underline{k^1i^2}.\text{-}k^1i^2.l^3o^4.l^5o^6$	*!*		2
c. 💣 $k^1i^2.l^3\text{-}\underline{o^4}.\text{-}o^4.l^5o^6$		3	

- This shows that the general *REPEAT constraint must rank below ALIGN-RED-L. If this were the only active *REPEAT constraint, we would no longer be able to generate Type D infixation at all.

[4] In a less surface-oriented theory of reduplication — for example, McCarthy & Prince's (1995) "Full Model" with Input-Reduplicant faithfulness, or Inkelas & Zoll's (2005) Morphological Doubling Theory — it might be possible to view the optimal output as a different parse: $[\underline{g^1u^2}.\text{-}u^2.g^3u^4.y^5a^6]$, with a C_1V_1- reduplicant and deletion of the (underlying) base-initial C_1 (see Zukoff 2021:Appendix A.2). This is not a possible candidate in the "Basic Model" of BRCT, because the reduplicant would be "copying" something that is non-existent. (The reduplicant-initial [g] could only be epenthetic, and thus it should have an unmarked quality.)

⋆ Allowing *Repeat(init) to rank high while general *Repeat is ranked low (or non-existent), we derive the correct results for /kilolo/:

(39) /kilolo/ → [kilo-kilolo] *(Type B reduplication) with* *Repeat(init)

/red, $k^1i^2l^3o^4l^5o^6$/	*Repeat (init)	Align-Red-L	Align-Root-L	*Repeat
a. ☞ $k^1i^2.l^3o^4.\text{-}k^1i^2.l^3o^4.l^5o^6$			4	*
b. $k^1i^2.\text{-}k^1i^2.l^3o^4.l^5o^6$	*!		2	**
c. $k^1i^2.l^3\text{-}o^4.\text{-}o^4.l^5o^6$		3!		

• Using *Repeat(init) also comports with the one attested vowel-initial root with identical V_1 and V_2, which attests Type C reduplication that creates medial identical syllables:

(40) *o.to.wi* 'make an appointment' → *o.t-o.to.wi* [Ezard 1980:147; Inkelas & Zoll 2005:95, HHK:26]

> ∗ This form is cited in Ezard (1980), an early paper on reduplication in Tawala, but not in Ezard (1997), the subsequent Tawala grammar. HHK (26) suggest that this might mean that the form is erroneous. The only aspect of the analysis hinging on this form is whether we can establish a crucial ranking between *Repeat and Align-Root-L.

• In (41), we see that general *Repeat must be ranked *below* Align-Root-L, or else candidate (41a), which additionally copies V_2 to avoid the medial repetition, would be preferred to desired candidate (41b).

(41) *Type C reduplication for* /otowi/ *with* *Repeat(init)

/red, otowi/	*Repeat(init)	Anch-L-BR	Aln-Red-L	Aln-Rt-L	*Repeat
a. o.to.-o.to.wi				3!	
b. ☞ o.t-o.to.wi				2	*
c. o.-o.to.wi	*!			1	*
d. t-o.to.wi	*!	*!		1	*
e. w-o.to.wi		*!		1	
f. o.t-o.-o.wi			2!		

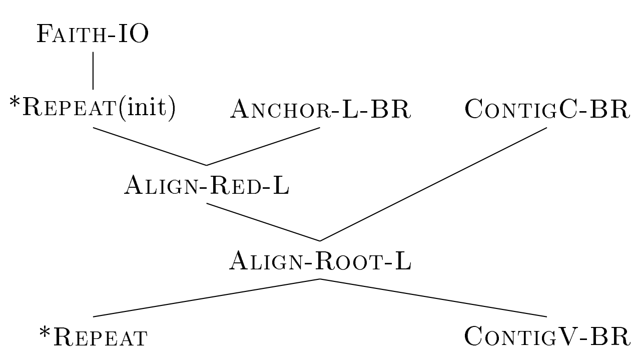
⋆ Whether or not we need higher-ranked *Repeat(init) to rule out candidates in this instance, we know from Type B that the operative *Repeat constraint must outrank Align-Root-L.

→ Therefore, this form provides additional evidence that we need *Repeat(init) rather than general *Repeat.

6 Conclusion and discussion

• The rankings motivated above are summarized in (42):

(42) *Ranking summary for Tawala reduplication*



- I have proposed two revisions to Hicks Kennard's (2004) analysis:

(43) **Splitting Contiguity-BR** into separate constraints, one banning consonant skipping, the other banning vowel skipping.
↪ This fixes HK's unrecognized prediction of Type A reduplication for $C_1V_1C_2V_2$-initial bases.

(44) Restricting the active *Repeat constraint to **initial position**.
↪ This fixes HK's unrecognized prediction of Type D reduplicative infixation to repair non-base-initial repetitions.

- These changes allow us to understand the system as preferring the shortest possible reduplicant, subject to the needs of higher-ranked constraints.
  - This is consistent with the a-templatic approach to reduplication (e.g. Spaelti 1997, Hendricks 1999, Riggle 2006, Zukoff 2016).
  - This is also consistent with the observed stress facts, such that reduplication-specific stress (or footing) requirements can induce exceptional stress while not having an effect on reduplicant shape.

⋆ The relativized Contiguity approach not only solves the Type B copying problem, it clarifies HK's "gradient" evaluation of Contiguity.

- For example, HK (314) uses Contiguity-BR to prefer copying V_2 [bi-beiha] (45b) rather than V_3 *[ba-beiha] (45c) in Type A forms.

(45) *Selecting the reduplicative vowel with gradient* Contiguity (HK's approach)

/red, beiha/			*Repeat	Contig-BR	
a.		be-be.i.ha	*!		
b.	☞	bi-be.i.ha		*	(e)
c.		ba-be.i.ha		**!*	(e,i!,h)

⋆ However, if we adopted the traditional definition (cf. (11a)), we actually shouldn't be able to distinguish between the two:

(46) *Traditional definition of* (Input) Contiguity (HK:308; cf. McCarthy & Prince 1995:123)
Assign one violation * if the reductplicant doesn't correspond to a contiguous substring of the base.

→ On the other hand, my new definition, repeated here (cf. (11b, 15, 16) above), spells out a method for categorical violation assignment over multiple loci.

(47) **Contiguity(X)-B(→)R** ("*Don't skip X's–BR*"):
For a reduplicant string $r_1 \ldots r_n$ standing in correspondence with a base string $b_1 \ldots b_n$, assign one violation * for each **segment/C/V/X** between b_1 and b_n which lacks a correspondent in $r_1 \ldots r_n$.

- With this in hand, for a Type A base, ContigC-BR will rule out $V_{n>2}$-copying (48c) because it skips any/all subsequent consonant.
  - Even if we had any example with no subsequent consonants, ContigV-BR would assign a violation for skipping V_2 and any subsequent vowels.

(48) *Selecting the reduplicative vowel with relativized* Contiguity

/red, beiha/			*Repeat	ContigC-BR		ContigV-BR	
a.		be-be.i.ha	*!				
b.	☞	bi-be.i.ha				*	
c.		ba-be.i.ha		*!	(h!)	**	(e,i)

→ This shows that this revised approach addresses multiple analytical questions within Tawala, as well as giving us traction on our theoretical understanding of Contiguity.

- There is perhaps an alternative approach to the role of ContigC-BR:
  → A faithfulness constraint on **consonant-vowel transitions** (49):

(49) **Ident[CVtrans]/V-BR:** If V_b and V_r both have CV transitions, they must be identical.

- For a Type B base (e.g. /kimaya/), this constraint would be violated by C_1V_2-copying (50c), because base-V_2 ([a]) follows one consonant in the base ([m]) but a different consonant in the reduplicant ([k]).

(50) *Type B reduplication with* Ident[CVtrans]/V-BR

/red, kimaya/	*Repeat(init)	Ident[CVtrans]/V-BR	Align-Root-L
a. ☞ ki.ma-ki.ma.ya			4
b. ki-ki.ma.ya	*!		2
c. **ka**-ki.**ma**.ya		*!	2

- On the other hand, for a Type A base (e.g. /beiha/), this constraint would be *vacuously satisfied* by C_1V_2-copying (51c), because base-V_2 ([i]) *doesn't follow a consonant in the base.*

(51) *Type A reduplication with* Ident[CVtrans]/V-BR

/red, beiha/	*Repeat(init)	Ident[CVtrans]/V-BR	Align-Root-L
a. be.i-be.i.ha			3!
b. be-be.i.ha	*!		2
c. ☞ **bi**-be.**i**.ha			2
d. **ba**-be.i.**ha**		*!	2

∗ We may have a problem with [h]'s though: presumably, [h] doesn't induce CV transitions on a following vowel.
  - In (51), we need Ident[CVtrans]/V-BR to penalize (51d) *[**ba**-be.i.**ha**] the same way that it penalized (50c) *[**ka**-ki.**ma**.ya].
  - Likewise, we need it to penalize *[**he**-hu.**ne**.ya] the same as *[**ka**-ki.**ma**.ya].

⋆ So, maybe we should stick with the Contiguity-based approach.

References

Ezard, Bryan. 1980. Reduplication in Tawala. *Kivung: Journal of the Linguistic Society of Papua New Guinea* 12(2):145–160. https://www.langlxmelanesia.com/llm-archive.

———. 1997. *A Grammar of Tawala: An Austronesian Language of the Milne Bay Area, Papua New Guinea*. Canberra: Pacific Linguistics.

Haugen, Jason D. 2009. What is the Base for Reduplication? *Linguistic Inquiry* 40(3):505–514. doi:10.1162/ling.2009.40.3.505.

Haugen, Jason D. & Cathy Hicks Kennard. 2011. Base-Dependence in Reduplication. *Morphology* 21(1):1–29. doi:10.1007/s11525-010-9154-5.

Hendricks, Sean Q. 1999. Reduplication without Template Constraints: A Study in Bare-Consonant Reduplication. PhD Dissertation, University of Arizona.

Hicks Kennard, Catherine. 2004. Copy but Don't Repeat: The Conflict of Dissimilation and Reduplication in the Tawala Durative. *Phonology* 21(3):303–323. doi:10.1017/S0952675704000296.

Inkelas, Sharon & Cheryl Zoll. 2005. *Reduplication: Doubling in Morphology*. Cambridge, UK: Cambridge University Press.

Itô, Junko. 1989. A Prosodic Theory of Epenthesis. *Natural Language & Linguistic Theory* 7(2):217–259. doi:10.1007/BF00138077.

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

McCarthy, John J. 2003. OT Constraints are Categorical. *Phonology* 20(1):75–138.

McCarthy, John J. & Alan Prince. 1993a. Generalized Alignment. In Geert Booij & Jaap van Marle (eds.), *Yearbook of Morphology 1993*, 79–153. Kluwer. doi:10.1007/978-94-017-3712-8_4.

———. 1993b. Prosodic Morphology I: Constraint Interaction and Satisfaction. *Linguistics Department Faculty Publication Series* 14 (2001 version). http://scholarworks.umass.edu/linguist_faculty_pubs/14.

———. 1994. The Emergence of the Unmarked: Optimality in Prosodic Morphology. In Mercè Gonzàlez (ed.), *NELS 24: Proceedings of the North East Linguistic Society*, 333–379. Amherst, MA: Graduate Linguistics Student Association. http://works.bepress.com/john_j_mccarthy/43.

———. 1995. Faithfulness and Reduplicative Identity. In Jill Beckman, Suzanne Urbanczyk & Laura Walsh Dickey (eds.), *Papers in Optimality Theory* (University of Massachusetts Occasional Papers in Linguistics 18), 249–384. Amherst, MA: Graduate Linguistics Student Association. http://works.bepress.com/john_j_mccarthy/44.

Prince, Alan & Paul Smolensky. 2004. *Optimality Theory: Constraint Interaction in Generative Grammar*. Malden, MA: Blackwell Publishing.

Riggle, Jason. 2006. Infixing Reduplication in Pima and its Theoretical Consequences. *Natural Language & Linguistic Theory* 24(3):857–891.

Saba Kirchner, Jesse. 2010. Minimal Reduplication. PhD Dissertation, University of California, Santa Cruz.

———. 2013. Minimal Reduplication and Reduplicative Exponence. *Morphology* 23(2):227–243.

Shaw, Patricia A. 2005. Non-Adjacency in Reduplication. In Bernard Hurch (ed.), *Studies on Reduplication*, 161–210. New York: Mouton de Gruyter.

Spaelti, Philip. 1997. Dimensions of Variation in Multi-Pattern Reduplication. PhD Dissertation, University of California, Santa Cruz.

Urbanczyk, Suzanne. 1996. Morphological Templates in Reduplication. In Kiyomi Kusumoto (ed.), *NELS 26: Proceedings of the North East Linguistic Society*, 425–440. Amherst, MA: Graduate Linguistics Student Association.

Yip, Moira. 1995. Repetition and its Avoidance: The Case of Javanese. Ms., University of California, Irvine. http://arizona.openrepository.com/arizona/handle/10150/227280.

Yu, Alan C. L. 2007. *A Natural History of Infixation*. Oxford: Oxford University Press.

Zukoff, Sam. 2016. Stress Restricts Reduplication. In Adam Albright & Michelle Fullwood (eds.), *Supplemental Proceedings of the Annual Meeting on Phonology 2014*, 1–12. Washington, DC: LSA. doi:10.3765/amp.v2i0.3742.

———. 2021. Reduplication in Ponapean (and Tawala): Re-examining "Base-Dependence". Paper Presented at AFLA 28, Montreal & Singapore, May 25–28, 2021. https://www.samzukoff.com/afla2021handout.