Reduplicant Shape Alternations in Ponapean and Tawala: Re-evaluating Base-Dependence

Sam Zukoff
samzukoff@gmail.com · www.samzukoff.com

September 27, 2022

Abstract

Most theories of reduplication, including Base-Reduplicant Correspondence Theory (McCarthy & Prince 1995), predict the existence of "base-dependent" reduplicant shape alternations, wherein reduplicant shape crucially depends on information present only in the surface reduplicant+base string (Haugen & Hicks Kennard 2011). Inkelas & Zoll (2005) assert that Morphological Doubling Theory (MDT) does not. This paper seeks to clarify the state of affairs regarding base-dependent reduplicant shape alternations by diving deeply into the analyses of Ponapean (Rehg & Sohl 1981; Kennedy 2003) and Tawala (Ezard 1997; Hicks Kennard 2004), which have both been identified as potential instances of base-dependence. I conclude that these languages' patterns *do* constitute base-dependence, yet they nevertheless *are* amenable to MDT analysis. Therefore, MDT's claims regarding base-dependent reduplicant shape alternations cannot be sustained and are *not* a reliable means of distinguishing between MDT and other theories of reduplication after all.

1 Introduction

The Austronesian languages Ponapean (Rehg & Sohl 1981) and Tawala (Ezard 1997) both display intricate phonologically-driven reduplicant shape alternations. These patterns are particularly noteworthy because they have the hallmarks of "base-dependence" (Inkelas & Zoll 2005:92–95, Haugen & Hicks Kennard 2011):

(1) **Base-dependence** [for reduplicant shape]: the shape of the reduplicant appears to crucially depend on information present only in the *surface reduplicant+base string*.

Most theories of reduplication, including Base-Reduplicant Correspondence Theory (BRCT; McCarthy & Prince 1995, 1999), are equipped or indeed designed to accommodate base-dependent reduplication patterns. On the other hand, Morphological Doubling Theory (MDT; Inkelas & Zoll 2005) is argued to predict the *absence* of base-dependence in reduplication (ibid.:92). If these predictions are accurately stated, then base-dependence constitutes a domain for distinguishing between these theories of reduplication.

Inkelas & Zoll (2005:94–96, incl. n. 18) identify Ponapean and Tawala as two *prima facie* instances of base-dependent reduplicant shape alternations, but assert that both admit to analysis via alternative interpretations in MDT, though the analyses are not sketched out in full. Nevertheless, Haugen & Hicks Kennard (2011), when considering additional evidence from Tawala, argue that the reduplicant shape alternations in that language do in fact exhibit base-dependence, and are unanalyzable in MDT. Zukoff (2020b) makes the same sort of argument for Ponapean.

This paper seeks to clarify the state of affairs regarding base-dependence in these languages by digging more deeply into the analytical possibilities available in the different theories. The argument to be advanced in this paper splits the difference between the two sides of the argument: while the reduplicant shape alternations in Ponapean and Tawala *do* constitute base-dependence by any reasonable definition, they in fact *are* amenable to MDT analysis. By invoking independently argued-for technology relating to prosodic constituency, the parallelist BRCT analyses can be imported into MDT in a way that preserves their base-

dependent character. I conclude that Inkelas & Zoll's (2005) claim that MDT cannot capture base-dependent reduplicant shape alternations cannot be sustained. This means that reduplicant shape alternations, even those which superficially appear to be base-dependent, are *not* a reliable means of distinguishing between MDT and other theories of reduplication after all.

This paper is structured as follows. In Section 2, I propose a slightly revised version of Zukoff's (2020b) BRCT analysis of Ponapean. In Section 3, I sketch the architecture of MDT, and show more precisely how Ponapean instantiates base-dependence. In Section 4, I work through a viable MDT analysis of the Ponapean data, using prosodic constituency, that undermines the claim that base-dependence is not predicted by MDT. (Appendix A lays out an alternative MDT analysis of Ponapean based on distributed deletion operations, illustrating that there are indeed multiple ways of deriving seemingly base-dependent patterns in MDT.) In Section 5, I develop competing BRCT and MDT analyses of reduplication in Tawala, and show that the same argument holds, and additionally that the MDT analysis requires an undesirable amount of powerful machinery. In Section 6, I review my argument that base-dependence, in terms of reduplicant-shape alternations at least, is not a suitable grounds for distinguishing between BRCT and MDT.

2 Reduplication in Ponapean: a BRCT analysis

In Ponapean (or Pohnpeian; Pohnpei, Micronesian, Oceanic; Rehg & Sohl 1981), durative aspect is marked by a prefixal partial reduplication pattern (ibid.:§3.3.4, also §2.9.5). This reduplicant predictably alternates in length between one and two moras, as previewed in (2). Note that the 2-mora reduplicants have a variety of segmental shapes, whose distribution is also predictable (described in Rehg & Sohl 1981, Rehg 1984, Goodman 1995; further analyzed in, e.g., Blevins & Garrett 1992, 1993, Kennedy 2002, 2003, Davis 2003, Kurisu 2013, Zukoff 2020a). I will collapse over these alternations for the purposes of this paper.[1]

(2) *Ponapean reduplication*

	Base length			
	1-mora base	**2-mora base**	**3-mora base**	**4-mora base**
1-mora reduplicant		*du-duup* *la-laud* *ke-kens*		*to-tooroor* *lu-luum^{w}uum^{w}* *so-soupisek*
2-mora reduplicant	*paa-pa* *tepi-tep* *don-dod*	*dun-dune* *sipi-siped* *rer-rere*	*duu-duupek* *mee-meelel* *lil-linenek*	*rii-riaala* *lil-lirooro* *lidi-liduwii*

I argue that this alternation in the moraic length of the reduplicant can be derived through a relatively simple interaction between stress and phonotactics (building on Kennedy 2002, 2003), using constraints whose domains of evaluation happen to span the base and the reduplicant. In order for this analysis to work, the module of grammar where the length/shape of the reduplicant is calculated must have the following properties:

(3) a. It must have access to the *surface properties* (i.e. predictable stress) of the base.
b. It must have access to the reduplicant's *position* relative to the base.

This is precisely what is intended by Inkelas & Zoll's (2005) "base-dependence". In this section, I lay out my analysis, which is lightly couched in BRCT, but probably compatible with other constraint-based theories that allow for base-dependence.

[1] In this paper, I will not consider vowel-initial or syllabic-consonant-initial forms (consult, e.g., Kennedy 2003:93–100). It is not clear whether these patterns follow completely from any of the analyses below, but they appear to be consistent with the constraints and rankings proposed here.

2.1 A phonological preliminary: Stress and accent in Ponapean

Rehg (1993:29) describes the Ponapean stress and accent system as follows: "*High pitch occurs on the penultimate mora, while primary stress is on the final mora; secondary stress occurs on alternate preceding morae*". I will assume that we can scale up from this short description, but these facts should be verified by future fieldwork. I will (with one exception) assign stress algorithmically according to this characterization. Abstracting away from the tonal realization of the accent and focusing purely on the position of stress, we can summarize the stress pattern as in (4), and analyze it with the (foot-free) constraints in (5) and (6).[2]

(4) *Stress pattern*

a.	Primary stress on rightmost mora	[StressR$_\mu$ (5a)]
b.	R→L alternating secondary stress by mora	[*Clash$_\mu$ (5b), *Lapse$_\mu$ (5c)]
c.	Medial coda C's are moraic, final coda C's are not	[*C$_\mu$# (6a) ≫ WxP (6b)]

(5) *Stress constraints*

a.	**StressR$_\mu$:**	Assign one violation * if the final mora is unstressed.	(*$\breve{\mu}$#)
b.	***Clash$_\mu$:**	Assign one violation * for each sequence of two *stressed* moras.	(*$\acute{\mu}\acute{\mu}$)
c.	***Lapse$_\mu$:**	Assign one violation * for each sequence of two *unstressed* moras.	(*$\breve{\mu}\breve{\mu}$)

(6) *Weight constraints*

a. ***C$_\mu$#:** Assign one violation * for each moraic word-final consonant.
b. **Weight-by-Position (WxP):** Assign one violation * for each non-moraic coda consonant.

The strictly alternating rhythm means that the stress status of the *initial mora* of a base is directly dependent on the moraic length of the base, as captured in (7). This difference will be crucial in explaining the distribution of reduplicant shape.

(7) a. *Odd* mora count bases ⇒ *stress* on the initial mora
b. *Even* mora count bases ⇒ *no stress* on the initial mora (stress on the peninitial mora)

2.2 Reduplicant shape alternation: Data and generalizations

Building on McCarthy & Prince (1986), Kennedy (2002, 2003) shows that considerations of stress and syllable weight in the base factor into determining the length of the reduplicant (in moras). I argue that reduplicant length is explained entirely by the stress and the weight of the *base-initial syllable*. We can begin to see this by arranging the data in terms of the mora count of the base and the mora count of the reduplicant, as laid out in (8) (chart adapted from Kennedy 2002:225, Kennedy 2003:80):

[2] On foot-free constraint-based approaches to stress, see, e.g., Elenbaas & Kager (1999), Gordon (2002), Stanton (2014, 2015); following Prince (1983), Selkirk (1984), *a.o.*

(8) *Ponapean reduplication: shape alternations by moras* (reduplicants bolded)[3]

	Odd	Even	Odd	Even
	1-mora base	**2-mora base**	**3-mora base**	**4-mora base**
1-mora reduplicant		**dù**-duúp **là**-laúd **kè**-keńs		**tò**-toò.roór **lù**-luù.m^{w}uúm^{w} **sò**-soù.pi.sék
2-mora reduplicant	**pàa**-pá **tè.pi**-tép **dòn**-dód	**dùn**-du.né **sì.pi**-si.péd **rèr**-re.ré	**dùu**-dùu.pék **mèe**-mèe.lél **lìl**-lì.ne.nék	**rìi**-ri.àa.lá **lìl**-li.ròo.ró **lì.di**-li.dù.wií
			6-mora base → 1-mora reduplicant:	**wà**-waàn.tùu.ké

A clear generalization emerges when looking at the mora count of the base: *odd* mora count bases always have *2-mora* reduplicants, but *even* mora count bases may have either a *1-mora* reduplicant or a *2-mora* reduplicant. Recall that stress is strictly alternating from right to left by mora, which means that odd mora count bases have initial-mora stress, whereas even mora count bases have peninitial-mora stress. Therefore, this generalization about mora count can actually be reduced to stress:

(9) a. Bases with *initial-mora stress* always have *2-mora* reduplicants.
b. Bases with *peninitial-mora stress* have either a *1-mora* reduplicant or a *2-mora* reduplicant.

Among the even mora count (i.e. peninitial-mora stress) bases (extracted in (10)), there is a consistent difference that determines which reduplicant length occurs: if it has an initial *light* syllable (i.e. consonant + short vowel), it always has a *2-mora* reduplicant; if it has an initial (super)*heavy* syllable (i.e. a complex rhyme), it always has a *1-mora* reduplicant.

(10) *Ponapean reduplication: even mora count bases* (base-initial syllables bolded)

	Even		
	2-mora base	**4-mora base**	**6-mora base**
1-mora reduplicant *Initial heavy syllable*	dù-**duúp** là-**laúd** kè-**keńs**	tò-**toò**.roór lù-**luù**.m^{w}uúm^{w} sò-**soù**.pi.sék	wà-**waàn**.tùu.ké
2-mora reduplicant *Initial light syllable*	dùn-**du**.né sì.pi-**si**.péd rèr-**re**.ré	rìi-**ri**.àa.lá lìl-**li**.ròo.ró lì.di-**li**.dù.wií	

Before proceeding to the analysis, we can take note of one further generalization that will help us leverage the stress facts (following Kennedy 2002:226). According to the assumptions laid out above (see esp. fn 3), all reduplicants bear a stress on their leftmost mora. This is true of all forms in (8)/(10), regardless of the length/composition of the reduplicant or the base.

2.3 Components of the analysis

We can boil the above generalizations down into an analysis with four component parts:

(11) a. A preference for shorter (i.e. monomoraic) reduplicants [Align-Root-L ≫ Max-BR]
b. A requirement that the reduplicant bear first-mora stress [StressL-Red ≫ *Lapse$_{\mu}$]

[3] Zukoff (2020b) assumes that the 2-mora reduplicants to even mora count bases have stress on the second mora of the reduplicant, as per Rehg's (1993) stress algorithm. In this paper, I will assume, following Kennedy (2003), that these forms instead have stress on the *first* mora of the reduplicant. This will bring the BRCT analysis in line with the MDT analyses proposed below. But note that my BRCT analysis is consistent with second-mora stress, requiring only a single ranking change; the MDT analyses, on the other hand, would have more trouble with that. This is an empirical question that deserves investigation.

c. A ban on moraic clash [*Clash$_\mu$]
d. A ban on adjacent identical light syllables [*Repeat(light)]

I motivate the preference for shorter reduplicants (11a) with the "size restrictor" constraint Align-Root-L (12), following Hendricks's (1999) "compression model" of reduplicant shape.[4] Note that I define the unit of alignment (Hyde 2012) as the "timing slot" rather than the segment. This is because we will need long vowels, diphthongs, and short vowel + coda consonant sequences to count the same. To effectuate the preference for shorter reduplicants, Align-Root-L must outrank Max-BR (13), the constraint preferring complete copying.

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

(13) **Max-BR:** Assign one violation * for each segment in the base which lacks a correspondent in the reduplicant. (McCarthy & Prince 1995)

(14) **Ranking:** Align-Root-L ≫ Max-BR

I enforce the reduplicant stress requirement (11b) via the constraint StressL-Red in (15). This is the only context where (moraic) stress lapses can occur (cf. (8)). Therefore, we need this constraint to outrank *Lapse$_\mu$ (5c).[5]

(15) **StressL-Red:** Assign one violation * for each reduplicant whose leftmost mora is not stressed.

(16) **Ranking:** StressL-Red ≫ *Lapse$_\mu$

The ban on moraic clash (11c) follows from the same *Clash$_\mu$ constraint (5b) (repeated here) needed for the basic stress pattern. This constraint (when coupled with the effect of StressL-Red) motivates reduplicant extension for bases with initial stress.

(17) ***Clash$_\mu$:** Assign one violation * for each sequence of two adjacent *stressed* moras.

The ban on adjacent identical light syllables (11d) is encoded with the constraint *Repeat(light) (discussed further in Section 2.4.3). This will motivate reduplicant extension for bases with initial light syllables.

(18) ***Repeat(light):** Assign one violation * for each sequence of two adjacent identical light syllables (i.e. *[C$_\alpha$V̆$_\beta$]$_\sigma$[C$_\alpha$V̆$_\beta$]$_\sigma$). (cf. Yip 1995, Hicks Kennard 2004)

These last three constraints all outrank the size restrictor constraint (and *Lapse$_\mu$), as shown in (19). The main take-away from this analysis will be that the combined effect of these constraints can *override* the preference for short reduplicants, yielding otherwise dispreferred 2-mora reduplicants for particular base shapes.

(19) **Ranking:** StressL-Red, *Clash$_\mu$, *Repeat(light) ≫ Align-Root-L, *Lapse$_\mu$

2.4 Reduplicant shape alternation: Analysis

This analysis divides up the data into three distinct cases (20). I will also focus on a sub-type of (20c) where the base itself contains a repetition (Section 2.4.4).

(20) a. 1-mora reduplicants to even mora count bases: *the default case* [Section 2.4.1]
b. 2-mora reduplicants to odd mora count bases: *extended by stress* [Section 2.4.2]
c. 2-mora reduplicants to light-syllable-initial bases: *extended by* *Repeat [Section 2.4.3]

[4] On size restrictor constraints and the "a-templatic" approach to reduplicant shape generally, see, e.g., Spaelti (1997), Hendricks (1999), Riggle (2006), Zukoff (2016, 2017). Templatic constraints could also generate this effect (Red = μ ≫ Red = 2μ; Zukoff 2016), but such an approach would ultimately be incompatible with MDT in this case.

[5] If we were to assume that 2-mora reduplicants to even mora count bases have peninitial rather than initial stress (see fn. 3), this ranking would be reversed. This assumption and this ranking are consistent with the rest of the BRCT analysis.

2.4.1 Even mora count bases → 1-mora reduplicants: the default case

When StressL-Red, *Clash$_\mu$, and *Repeat(light) can all be satisfied, the default preference for a monomoraic reduplicant is actualized. This happens only when the following two conditions are met simultaneously: (i) the base has an *even number of moras*, such that the leftmost base stress falls on the base's peninitial mora; and (ii) the base begins with a *heavy or superheavy syllable*, such that a monomoraic reduplicant won't yield adjacent identical light syllables *when concatenated with the base*. The forms from (8) above that meet this description are pulled out in (21).

(21) *1-mora reduplicants*

	Even		
	2-mora base	**4-mora base**	**6-mora base**
1-mora reduplicant *Initial heavy syllable*	**dù**-duúp **là**-laúd **kè**-keńs	**tò**-toò.roór **lù**-luù.m^{w}uúm^{w} **sò**-soù.pi.sék	**wà**-waàn.tùu.ké

I illustrate the analysis of these cases in (23) with a monosyllabic, bimoraic base with a long vowel: [**dù**-duúp]. This tableau shows four candidate outputs that fully cross two variables, as shown in (22). Most of the tableaux in the rest of this section follow the same format. Candidates are accompanied by a numerical schematization of stress in terms of moras, where "1" indicates primary stress, "2" indicates secondary stress, and "0" indicates no stress.

(22) *Candidate comparisons*

	Initial stress?	
Red. length	**No**	**Yes**
1-mora	(23a)	(23b)
2-mora	(23c)	(23d)

(23) *Even mora count bases with initial heavy syllables yield 1μ reduplicants*

/red, duup/	*(stress profile)*	StressL-Red	*Clash$_\mu$	*Repeat	*Lapse$_\mu$	Aln-Rt-L
a. du-duúp	[0-01]	*!			*	2
b. ☞ dù-duúp	[2-01]					2
c. duù-duúp	[02-01]	*!				3
d. dùu-duúp	[20-01]				*!	3!

In (23), none of the candidates violate *Clash$_\mu$, because the base-initial mora is unstressed. Likewise, none of the candidates violate *Repeat(light), because the base-initial syllable is heavy. Candidates (23a) and (23c) don't stress the reduplicant-initial mora, and thus fatally violate StressL-Red. This leaves (23b) and (23d), pulled out in (24), as the only candidates that satisfy all the top-ranked constraints.

(24) *Candidates that satisfy top-ranked constraints*

/red, duup/		*Lapse$_\mu$	Align-Root-L
b. ☞ dù-duúp	[2-01]		2
d. dùu-duúp	[20-01]	*!	3!

The longer (2-mora) reduplicant in (24d) does worse on both of the lower-ranked constraints than does the shorter (1-mora) reduplicant in (24b). Candidate (24d) violates *Lapse$_\mu$ because of the sequence of reduplicant-final unstressed mora + base-initial unstressed mora. Candidate (24d) also incurs an extra violation of Align-Root-L because of the extra timing slot contributed by its reduplicant's extra mora. In this case, both of these constraints help select the 1-mora reduplicant in (24b).

2.4.2 Odd mora count bases → 2-mora reduplicants: StressL-Red + *Clash$_\mu$

When we move to odd mora count bases (25), the stress considerations flip: the strictly alternating rhythm places a stress on the first mora of the base. (This is the primary stress when the base is monomoraic, but a secondary stress in longer odd mora count bases.)

(25) *Odd mora count bases*

	Odd	
	1-mora base	**3-mora base**
2-mora reduplicant	**pàa**-pá	**dùu**-dùu.pék
	tè.pi-tép	**mèe**-mèe.lél
	dòn-dód	**lìl**-lì.ne.nék

Crucially, this now causes the stressed 1-mora reduplicant candidate (26b) to incur a *Clash$_\mu$ violation, because the reduplicant stress is now adjacent to the base-initial stress. We could fix that problem by not stressing the reduplicant (26a), but this would violate StressL-Red. As long as StressL-Red and *Clash$_\mu$ outrank Align-Root-L, it will be better in this case to tolerate the longer (2-mora) reduplicant (26d) than to mess up the stress pattern. This shows that the reduplicant is extended to 2 moras in case it can optimize the stress pattern. (Candidate (26c) is entirely pathological, in that it fails to stress the reduplicant-initial mora, it creates a clash, and it extends the reduplicant.)

(26) *Odd mora count bases yield 2μ reduplicants*

/red, duupek/		StressL-Red	*Clash$_\mu$	*Lapse$_\mu$	Align-Root-L
a.	du-dùu.pék [0-201]	*!			2
b.	dù-dùu.pék [2-201]		*!		2
c.	duù-dùu.pék [02-201]	*!	*!		3
d. ☞	dùu-dùu.pék [20-201]				3

This case allows us to glean several additional crucial aspects of the analysis. (The following tableaux only consider candidates that satisfy StressL-Red and *Clash$_\mu$.) First, the language does not diverge from its regular stress pattern in order to coerce a shorter reduplicant (i.e., better satisfaction of Align-Root-L). Candidate (27b) retracts the final stress to the penultimate mora (violating StressR$_\mu$), allowing for a stressed 1-mora reduplicant with alternating rhythm. Candidate (27c) eschews base-initial stress, allowing for a stressed 1-mora reduplicant without a clash, but at the expense of a medial lapse (violating *Lapse$_\mu$). Since the 2-mora reduplicant is optimal, we know that StressR$_\mu$ and *Lapse$_\mu$ outrank Align-Root-L.

(27) *Stress is more important than a short reduplicant*

/red, duupek/		StressR$_\mu$	*Lapse$_\mu$	Align-Root-L
a. ☞	dùu-dùu.pék [20-201]			3
b.	dù-duú.pek [2-010]	*!		2
c.	dù-duu.pék [2-001]		*!	2

Second, extending the reduplicant out beyond 2 moras is never helpful. Candidates (28b) and (28c) show that extending the reduplicant to 3 moras worsens the stress pattern, in addition to increasing violation of Align-Root-L. Candidate (28d) reveals that Align-Root-L is actually doing work in the analysis: it prefers the 2-mora reduplicant (28a) to the 4-mora reduplicant (28d), which have equally good stress patterns.[6]

[6] The sort of epenthesis represented in candidate (28d) is tolerated under the right phonotactic conditions, and never prevents the proper reduplicant length from surfacing; see, e.g., Kurisu (2013).

(28) *Extending the reduplicant further never helps*

/RED, duupek/		StressL-Red	*Lapse$_\mu$	Align-Root-L
a. ☞ dùu-dùu.pék	[20-201]			3
b. dùu.pe-dùu.pék	[200-201]		*!	5
c. duù.pe-dùu.pék	[020-201]	*!		5
d. dùu.pè.ki-dùu.pék	[2020-201]			7!

Lastly, the comparison between winning candidate (28a) and losing candidate (28d) shows why the *Repeat constraint must be restricted to *light* syllables: *Repeat must outrank Align-Root-L in order to generate the behavior of base-initial *light* syllables (see immediately below). If it also penalized adjacent identical *heavy* syllables, it would assign a violation to (28a). This would be enough to select (28d), which avoids the repeated identical heavy syllables by extending the reduplicant out to 4 moras (extra Align-Root-L violation). Therefore, we must be dealing with a version of *Repeat that is limited to light syllables.[7]

2.4.3 Light-syllable-initial bases → 2-mora reduplicants: *Repeat(light)

This stress-based account of reduplicant-extension does not get us the whole range of 2-mora reduplicants. It predicts that all and only odd mora count bases will have 2-mora reduplicants. However, we also find 2-mora reduplicants for even mora count bases with an initial *light syllable* (i.e., $[CV]_\sigma$), as exemplified in (29). This effect can be captured with the constraint *Repeat(light) (repeated in (30)), inspired by Yip's (1995) more general *Repeat constraint, via Hicks Kennard's (2004) analysis of Tawala (see Section 5).

(29) *Even mora count bases with initial light syllables*

	Even	
	2-mora base	**4-mora base**
2-mora reduplicant *Initial light syllable*	**dùn**-du.né	**rìi**-ri.àa.lá
	sì.pi-si.péd	**lìl**-li.ròo.ró
	rèr-re.ré	**lì.di**-li.dù.wií

(30) ***Repeat(light):** Assign one violation * for each sequence of two adjacent identical light syllables (i.e. *$[C_\alpha\breve{V}_\beta]_\sigma[C_\alpha\breve{V}_\beta]_\sigma$). [repeated from (18) above]

*Repeat(light) will motivate extension to a 2-mora reduplicant in case the base begins in a light syllable (31). Crucially, in this case, *Repeat(light) penalizes the stressed 1-mora reduplicant candidate (31b), because it has adjacent identical light syllables across the juncture. This is precisely parallel to the way that *Clash$_\mu$ assigns violations to the stressed 1-mora reduplicant candidate in odd mora count bases (cf. (26)). Again, the preferred alternative is the 2-mora reduplicant with initial stress (31d), even though this time it also incurs a *Lapse$_\mu$ violation. Note that the reduplicant-final [n] is moraic, and thus relevant for the *Lapse$_\mu$ and *Clash$_\mu$ constraints, because it is in coda position. This shows that the reduplicant is extended to 2 moras not only when that avoids a clash, but also when it can avoid a violation of *Repeat(light). (Additionally, the fact that (31d) is preferred to *[dùn-dú.ne] shows that *StressR$_\mu$ ≫ *Lapse$_\mu$.)

(31) *Even mora count bases with initial light syllables yield 2μ reduplicants*

/RED, dune/		StressL-Red	*Repeat(light)	*Lapse$_\mu$	Align-Root-L
a. du-du.né	[0-01]	*!	*!	*	2
b. dù-du.né	[2-01]		*!		2
c. duù-du.né	[02-01]	*!			3
d. ☞ dùn-du.né	[20-01]			*	3

[7] This explains why McCarthy & Prince's (1986) analysis based on "quantitative complementarity" struggled to account for the entire range of reduplicant shape alternations.

Several additional points can be made with respect to the form [dùn-du.né] and/or *Repeat(light). First, stress must not factor into the relevant calculation of "identical" for *Repeat(light), as the two syllables in losing candidate (31b) do differ in stress. If stress differences led to non-identity, it would incorrectly be selected as the winner. Hicks Kennard (2004:310) makes the same point for *Repeat in Tawala.

Second, Align-Root-L helps distinguish between different ways of achieving a 2-mora reduplicant. Winning candidate (32a) acquires its second mora by copying the nasal consonant from the second onset of the base, and assigning it a mora. Losing candidate (32b) additionally copies the second base vowel, syllabifying the [n] as a (non-moraic) onset. Since (32a) contains one fewer segment (i.e. timing slot) in the reduplicant than does (32b), Align-Root-L correctly selects it as the winner.

(32) *Selecting the right 2-mora reduplicant shape*

/red, dune/		*Repeat(light)	*Lapse$_\mu$	Align-Root-L
a. ☞ dùn-du.né	[20-01]		*	3
b. dù.ne-du.né	[20-01]		*	4!

Lastly, *Repeat(light) is equally applicable in light-syllable-initial *odd* mora count bases, for example root *pàdaák* → durative *pàda-pàdaák* (Rehg & Sohl 1981:93). In these cases, *Repeat(light) and *Clash$_\mu$ *both* advocate for extending the reduplicant to 2 moras.

(33) *Odd mora count bases with initial light syllables yield 2μ reduplicants*

/red, padaak/		StrsL-Red	*Clash$_\mu$	*Repeat	*Lapse$_\mu$	Aln-Rt-L
a. pa-pà.daák	[0-201]	*!		*!	*	2
b. pà-pà.daák	[2-201]		*!	*!		2
c. padà-pà.daák	[02-201]	*!	*!			4
d. ☞ pàda-pà.daák	[20-201]				*	4

2.4.4 *Repeat(light) as TETU

While *Repeat(light) has an observable effect on reduplicants in Ponapean, the language does not show evidence of sensitivity to this constraint in any other domain. That is to say, sequences of adjacent identical light syllables are freely tolerated outside of reduplication (Rehg 1984:325–326, Zukoff 2020a:17). Take, for example, the root in (34), which is composed of two identical light syllables.

(34) root *reré* 'to skin/peel' → durative *rèr-reré* (Rehg & Sohl 1981:80)

By having Input-Output faithfulness rank above *Repeat(light), but the constraints directly regulating reduplicant shape (namely, Align-Root-L) rank below it, we correctly derive this state of affairs (35). Faith-IO rules out various ways of altering the root that could eliminate all consecutive identical light syllables, including deleting the final root vowel (35c) or lengthening the first root vowel (35d). However, since, in BRCT at least, the reduplicant is not subject to IO-faithfulness (McCarthy & Prince 1995), it can expand or contract freely in order to avoid a *Repeat(light) violation, thus preferring (35b) over (35a).

(35) *Repeat(light) *affects the reduplicant but not the base* (TETU)

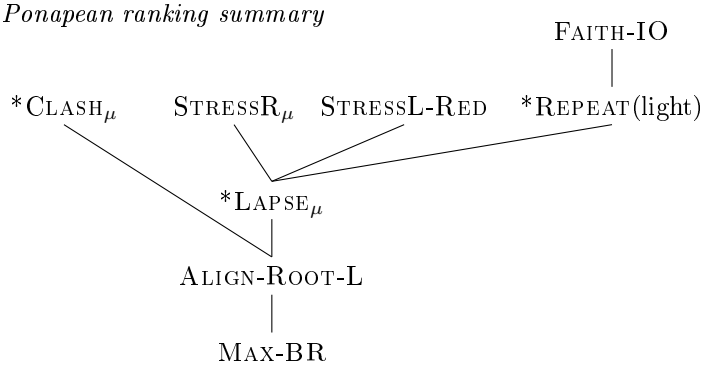
/red, rere/		Faith-IO	*Repeat(light)	Align-Root-L
a. rè-re.ré	[2-01]		**!	2
b. ☞ rèr-re.ré	[20-01]		*	3
c. rèr-rér	[20-1]	*!		3
d. rèe-rèe.ré	[20-201]	*!		3

This type of distribution — i.e. sensitivity only in reduplication — instantiates *the emergence of the unmarked* (TETU; McCarthy & Prince 1994, 1995). This is worth dwelling on because Morphological Doubling Theory treats TETU in a very different way. As a result, the *Repeat(light) facts will have a significant say in what constitutes a viable MDT analysis.

2.5 Local Summary

The complete ranking for the BRCT analysis of Ponapean reduplication is shown in (36):

(36) *Ponapean ranking summary*



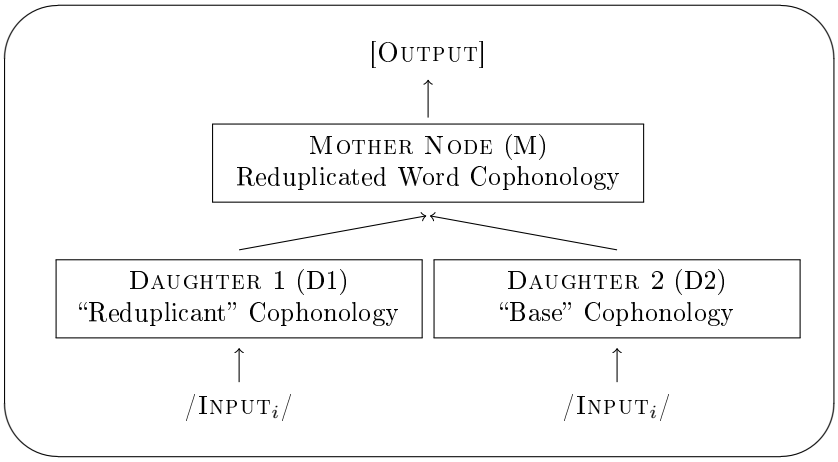
This analysis can be understood in a very straightforward way: reduplicants prefer to be 1-mora long, but they settle for being 2-moras long if that allows them to avoid clashes or light repetitions. This analysis is entirely surface-oriented and transparent. This may be a point in favor of BRCT, but it does not rule out analyses in other frameworks. In the following sections, I will show that a version of this analysis can in fact be implemented in Morphological Doubling Theory, despite its apparent "base-dependence".

3 Base-(in)dependence and the architecture of MDT

3.1 A brief overview of Morphological Doubling Theory

The basic approach to reduplication in MDT, as proposed by Inkelas & Zoll (2005) [henceforth IZ], can be schematized as in (37):

(37) *Reduplication in MDT*



The derivation begins with two identical (or at least semantically equivalent) inputs: /Input$_i$/. The two respective inputs pass through two *separate* derivational nodes (the "daughter" nodes), which are fully independent from one another (i.e. there's no information flow between them). One of these nodes calculates the "reduplicant" (here, D1), the other calculates the "base" (here, D2). These two nodes may have completely distinct phonological grammars, i.e. *cophonologies* (Orgun 1994, 1996, Inkelas, Orgun, & Zoll 1997, et seq.).

The outputs of the daughter nodes jointly form the input to a single derivational node (the "mother" node; M). This node applies its own cophonology (which, again, may be completely distinct) to its input. The mother node concatenates the daughter outputs according to this cophonology, to produce the final

output. There is no explicit distinction in status between material from the respective daughter nodes. That is, there is no formal equivalent of "base" vs. "reduplicant" once we reach the mother node.

IZ's claim that MDT predicts the absence of "base-dependence" is said to follow from this architecture. We can see this when we consider how it allows phonology to apply to its different morphological constituents:

(38) *Possible interactions in MDT*

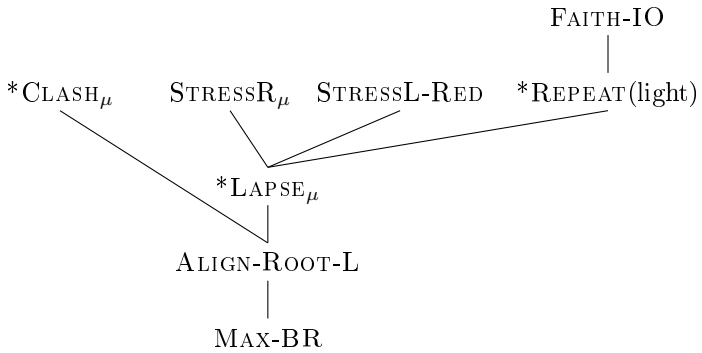
- ✓ Special phonology may apply to the reduplicant before it sees the base. **[D1]**
- ✓ Special phonology may apply to the constituent containing the reduplicant and the base. **[M]**
- ✗ Special phonology may *not* apply to the reduplicant once it sees the base.

If we observe a pattern that truly constitutes this last type of interaction, this should be a problem for MDT. However, we will see that, by appealing to prosodic constituents, Ponapean's apparent base-dependent patterns can be shoehorned into an MDT analysis that actually does recapitulate base-dependence.

3.2 Base-dependence in the analysis of Ponapean reduplication

The BRCT analysis of Ponapean (Section 2) hinges primarily on the operation of *Clash$_\mu$ and *Repeat(light), the constraints which motivate deviation from the default preference for a 1-mora reduplicant:

(39) *Ponapean BRCT-analysis ranking summary* (repeated from (36) above)



For both constraints, the structural descriptions encompass *sequences* (of moras or syllables, respectively). In the case at hand, the relevant sequences *span the base and the reduplicant.* That is to say, a 1-mora reduplicant is disallowed (and thus extended to 2 moras) if:

(40) a. There would be a *clash* across the reduplicant-base juncture, and/or
b. There would be *identical light syllables* across the reduplicant-base juncture.

We cannot reduce either of these to properties of the reduplicant in isolation: light syllable reduplicants and 1-mora stressed reduplicants are both allowed (in fact preferred) in other contexts. Also, since stress is *grammatically assigned* (i.e. predictable), the reduplicant must be aware of *surface* properties of the base, not just underlying ones. This means that the module in which reduplicant length is determined must have access to:

(41) a. The reduplicant's *position* relative to the base, and
b. The *surface* properties of the base

Again, this is precisely "base-dependence" (IZ:95; cf. Haugen & Hicks Kennard 2011). This should tell us something about the architecture of the reduplicative grammar:

(42) *This is compatible with a grammar where:*

- ✓ The base is computed and then the reduplicant is computed (with the base visible), or
- ✓ The base and reduplicant are computed together

Most theories of reduplication have an architecture of one of these types, including BRCT, Stratal OT (Kiparsky 2010), and Serial Template Satisfaction in Harmonic Serialism (McCarthy, Kimper, & Mullin 2012).

(43) *This is **not** compatible with a grammar where:*

✗ The reduplicant is computed and then the base is computed (with the reduplicant visible), or

✗ **The reduplicant and the base are computed separately**

MDT (purportedly) has this latter type of architecture. If we've characterized Ponapean correctly, and MDT's prediction about base-*in*dependence is correct, then MDT shouldn't be able to generate Ponapean. I will now show that, with sufficiently powerful technology, which has been independently argued for, MDT actually *can* generate the Ponapean pattern, but in doing so voids base-dependence in reduplicant shape as a reliable distinguisher between MDT and BRCT.

4 A Mother-Node-based MDT analysis of Ponapean

In this section, I show that the BRCT analysis can effectively be imported wholesale into MDT by locating the analysis in the "Mother Node", i.e. the reduplicated word cophonology. This works because the "BRCT" analysis does not actually rely on any active Base-Reduplicant faithfulness constraints.

The crux of the proposal is that BRCT's central "base" vs. "reduplicant" distinction can be recapitulated in MDT by asymmetric assignment of prosodic constituency between the daughter nodes. Faithfulness referencing these prosodic constituents allows for truncation in the "reduplicant" without generating truncation in the "base". This distinction subsumes the TETU character of the *Repeat(light) effect, which likewise affects the reduplicant but not the base. This is not the only available means of analyzing this pattern in MDT. In Appendix A, I lay out an alternative, more traditional MDT analysis of the Ponapean facts which distributes reduplicant deletion across D1 and M without resorting to prosodic constituency. Nevertheless, the fact that the prosodic constituency analysis is possible has substantial ramifications for our understanding of base-dependence in MDT.

4.1 Prosodic constituents and truncation in MDT: a quick detour into Javanese

One of the tools that IZ make extensive use of in their various analyses is the notion of *prosodic constituents* that are associated with, but distinct from, morphological constituents (IZ:140; following Booij 1985, Nespor & Vogel 1986, Sproat 1986, Inkelas 1990, Booij & Lieber 1993; see Cole 1994, Downing 1998a,b, *a.o.*, on reduplication).

(44) *Prosodic constituents*

a. Morphological root ⇝ Prosodic root (PRoot)
b. Morphological stem ⇝ Prosodic stem (PStem)
c. Morphological word ⇝ Prosodic word (PWord)

IZ (Ch. 5.1, esp. 140–141) introduce the formalism in their analysis of Javanese (Java, Malayo-Polynesian; e.g. Horne 1961, Sumukti 1971, Dudas 1976). They appeal to the PRoot in order to account for why certain affixal segments get copied along with the root in reduplication, as in the example in (45):

(45) *Javanese reduplicated causatives* (IZ:139)[8]
ROOT *uni* 'sound' → CAUSATIVE *ŋ-une-ʔake* → REDUPLICATED *une**ʔ**-une-ʔake* (**une*-une-ʔake*)

Their analysis is that the stem-node which feeds D1 (and D2) builds a PRoot (indicated by {...}) that includes all root segments and all adjacent non-root segments that don't add a syllable (i.e. adjacent consonants), and then D1 deletes all non-PRoot segments (see (51) below). IZ spell out their method for PRoot-parsing (p. 141). They employ the faithfulness constraints in (46), which introduce roots (R) and

[8] IZ suggest that the [ʔ] of the suffix is an independent morpheme, but this does not factor into their analysis.

PRoots (P) into the set of constituents which can be related by faithfulness, ranked as in (47).[9] Segments which are not parsed into PRoots are nonetheless retained in the output of this node. This is crucial for deriving the correct ultimate surface form, because this output serves also as the input to D2 (see below).

(46)
a. **MAX$_{SEG}$-RP:** All root segments should have correspondents in the PRoot.
b. **MAX$_{SYL}$-RP:** All root syllables should have correspondents in the PRoot.
c. **DEP$_{SEG}$-RP:** All PRoot segments should have correspondents in the root.
d. **DEP$_{SYL}$-RP:** All PRoot syllables should have correspondents in the root.
e. **MAX$_{SEG}$-IP:** All input segments should have correspondents in the PRoot.

(47) *PRoot parsing at the stem node*

/uni-ʔake/	DEP$_{SYL}$-RP	MAX$_{SEG}$-IP	DEP$_{SEG}$-RP
a. {u.ne}.ʔa.ke		4!	
b. {u.ne.ʔa.ke}	2!		4
c. ☞ {u.neʔ}.a.ke		3	1
d. {u}.ne.ʔa.ke		6!	

IZ are less explicit about how the deletion step works, but we can extrapolate that there are similar MAX/DEP constraints for segments referencing the PRoot-to-Output relation:

(48) **MAX$_{SEG}$-PO:** Assign one violation * for each segment contained within a PRoot in the input which lacks an output correspondent.

In D1, this MAX constraint must outrank a constraint motivating truncation, e.g. *STRUC (49). This constraint in turn outranks the general MAX-IO constraint, motivating deletion of everything outside the PRoot ((50b) ≻ (50a)) but nothing inside the PRoot ((50b) ≻ (50c)).

(49) ***STRUC[TURE]**: Assign one violation * for each segment in the output.

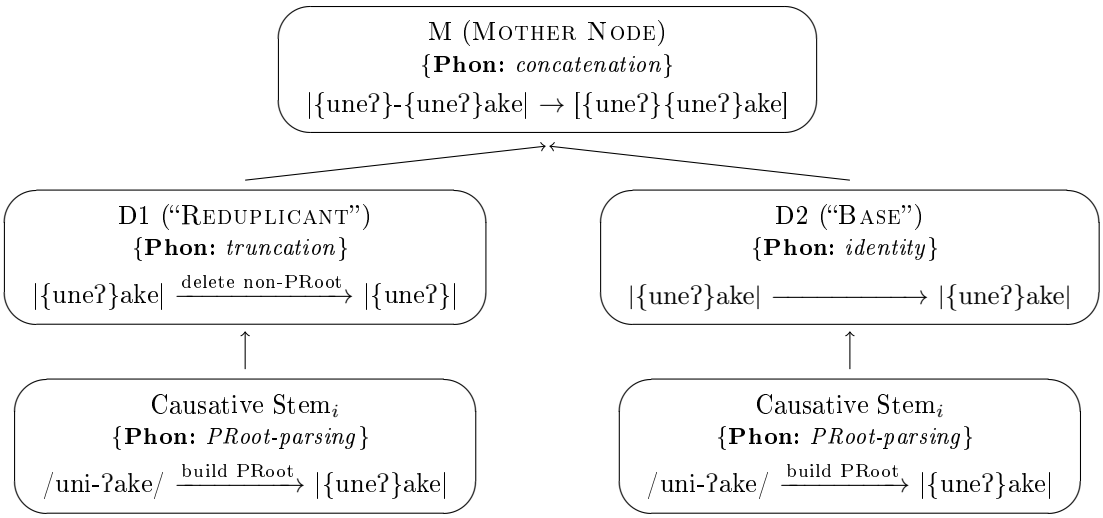
(50) *Deletion of non-PRoot segments in D1*

|{uneʔ}ake|	MAX$_{SEG}$-PO	*STRUC	MAX$_{SEG}$-IO
a. {uneʔ}ake		7!	
b. ☞ {uneʔ}		4	***
c. {une}	*!	3	****

As long as this is the output of D1, and the output of D2 is a faithful concatenated realization of the stem, then we derive the desired result:

[9] Note that their use of syllable-oriented faithfulness constraints presupposes that the root is parsed into syllables prior to this mapping.

(51) *Javanese PRoot-driven truncation in D1* (IZ:140)



IZ use these sorts of prosodic constituents in their analyses of a number of languages: Eastern Kadazan (Sabah, East Malaysia, Malayo-Polynesian; Hurlbut 1988; IZ:152–155); Tagalog (Philippines, Malayo-Polynesian; e.g. Schachter & Otanes 1972; IZ:183–185); Iniseño and Barbareño Chumash (California, Chumashan; Applegate 1972, 1976, Wash 1995; IZ:185–196; cf. McCarthy & Prince 1995). Therefore, we should view this technology as indispensable for MDT, and freely adopt it for new analyses.

4.2 PRoot-constrained deletion in the Mother Node in Ponapean

While IZ's analyses typically make use of PRoots to effectuate truncation in the *daughter nodes* (fed by an earlier node that builds the PRoot), there's no reason why we can't defer the equivalent interaction to the Mother Node. Taking this approach will allow us to import the BRCT analysis into the Mother Node wholesale.

4.2.1 Basics of the analysis

In this analysis, the daughter nodes have the following properties. D2 (the "base") builds a PRoot over all of its input material. (It doesn't matter whether it assigns stress.) D1 *doesn't* build a PRoot, but it assigns a *single* stress, to the leftmost mora (à la StressL-Red). Truncation then occurs in the Mother Node, but it is sensitive to two factors:

(52) a. Faithfulness to PRoot segments, via Max-PO
b. Faithfulness to stressed segments, via Max$_{\acute{\mu}}$-IO (53)

(53) **Max$_{\acute{\mu}}$-IO:** Assign one violation * for each segment associated with a stressed mora in the input which lacks an output correspondent.

While we could use *Struc to trigger deletion like in Javanese, we could also use an alignment constraint referencing PRoots (54). This has the exact same "size-restrictor" effect as Align-Root-L did in the BRCT analysis, as demonstrated in (55).

(54) **Align-PRoot-L:** Assign one violation * for each timing slot which intervenes between the left edge of a PRoot and the left edge of the word.

(55) *Even mora count bases with initial heavy syllables yield 1μ reduplicants*[10]

|láud-{laud}|			MAX-PO	MAX$_{\mu}$-IO	*CLASH$_{\mu}$	*LAPSE$_{\mu}$	ALN-PRT-L	MAX-IO
a.	làud-{laúd}	[20-01]				*!	4!	
b.	làu-{laúd}	[20-01]				*!	3!	1
c. ☞	là-{laúd}	[2-01]					2	2
d.	là-{lúd}	[2-1]	*!		*!		2	3
e.	lu-{lúd}	[0-1]	*!	*!			2	3
f.	{laúd}	[01]		*!				4
g.	{lúd}	[1]	*!	*!				5

Deleting segments from the PRoot (55d,e,g) isn't allowed because it violates undominated MAX-PO. Deleting PRoot segments confers no benefit with the constraints used here, though it would increase satisfaction of *STRUC. Deleting stressed "reduplicant" segments (55e,f,g) isn't allowed because it violates undominated MAX$_{\mu}$-IO. Because all PRoot segments are protected by MAX-PO, it doesn't matter whether the output of D2 contains stress or not, and thus whether deleting them ever violates MAX$_{\mu}$-IO. This leaves only the segments(/timing slots) between the reduplicant-initial stressed mora and the left edge of the PRoot open to deletion. Not deleting these segments (55a,b) results in increased violation of ALIGN-PROOT-L (and, in this case, *LAPSE$_{\mu}$), and therefore the candidate that deletes these segments (55c) is correctly chosen as the winner.

4.2.2 Blocking deletion with *CLASH$_{\mu}$ and *REPEAT(light)

Just as in the BRCT analysis, truncation down to a 1-mora reduplicant (the default preference) is blocked just in case it would violate *CLASH$_{\mu}$ (56) or *REPEAT(light) (57). In odd mora count bases, where stress is regularly assigned to the PRoot-initial mora, truncating the reduplicant down to its stressed initial mora will result in a clash (56c). As long as *CLASH$_{\mu}$ ≫ ALIGN-PROOT-L, it will be preferable *not* to delete the next mora (56b) so as to avoid the clash.

(56) *Odd mora count bases block reduction*

|dúupek-{duupek}|			MAX$_{\mu}$-IO	*CLASH$_{\mu}$	*LAPSE$_{\mu}$	ALN-PRT-L	MAX-IO
a.	dùu.pek-{dùu.pék}	[200-201]			*!	5!	
b. ☞	dùu-{dùu.pék}	[20-201]				3	3
c.	dù-{dùu.pék}	[2-201]		*!		2	3
d.	du-{dùu.pék}	[0-201]	*!			2	3
e.	{dùu.pék}	[201]	*!				5

In bases beginning in light syllables, truncating the reduplicant down to its stressed initial mora will result in identical light syllables across the juncture (57d). As long as *REPEAT(light) ≫ ALIGN-PROOT-L (and *LAPSE$_{\mu}$), it will be preferable *not* to delete the next mora (57b) so as to avoid the light repetition.

(57) *Light-syllable-initial bases block reduction*

|dúne-{dune}|			MAX$_{\mu}$-IO	*REPEAT(light)	*LAPSE$_{\mu}$	ALN-PRT-L	MAX-IO
a.	dù.ne-{du.né}	[20-01]			*	4!	
b. ☞	dùn-{du.né}	[20-01]			*	3	1
c.	duǹ-{du.né}	[02-01]	*!			3	1
d.	dù-{du.né}	[2-01]		*!		2	2

Additional ranking arguments show that, just as in the BRCT analysis, it is more important to apply the normal stress pattern than to have a shorter reduplicant (58) and that rightmost stress is more important than avoiding a lapse (59).

[10] Demotion of D1's primary stress to a secondary stress can be accounted for by requiring the rightmost stress to be the unique primary stress, and having the constraint(s) enforcing this outrank IDENT[stress degree]-IO.

(58) *Stress is mora important than PRoot alignment*

|dúupek-{duupek}|			STRESSR$_\mu$	*LAPSE$_\mu$	ALIGN-PROOT-L
a. ☞	dùu-{dùu.pék}	[20-201]			3
b.	dù-{duu.pék}	[2-001]		*!	2
c.	dù-{duú.pek}	[2-010]	*!		2

(59) *Rightmost stress is more important than lapse avoidance*

|dúne-{dune}|			STRESSR$_\mu$	*LAPSE$_\mu$	ALIGN-PROOT-L
a. ☞	dùn-{du.né}	[20-01]		*	2
d.	dùn-{dú.ne}	[20-10]	*!		2

4.2.3 MAX-PO generates TETU

We've just seen that MAX-PO prevents deletion of "base" material in the general case. It will also block deletion of base material that could repair base-internal and/or base-reduplicant-junctural *REPEAT(light) violations, given the ranking MAX-PO ≫ *REPEAT(light):

(60) MAX$_{\text{PROOT}}$-IO *prevents base reduction for* *REPEAT(light)

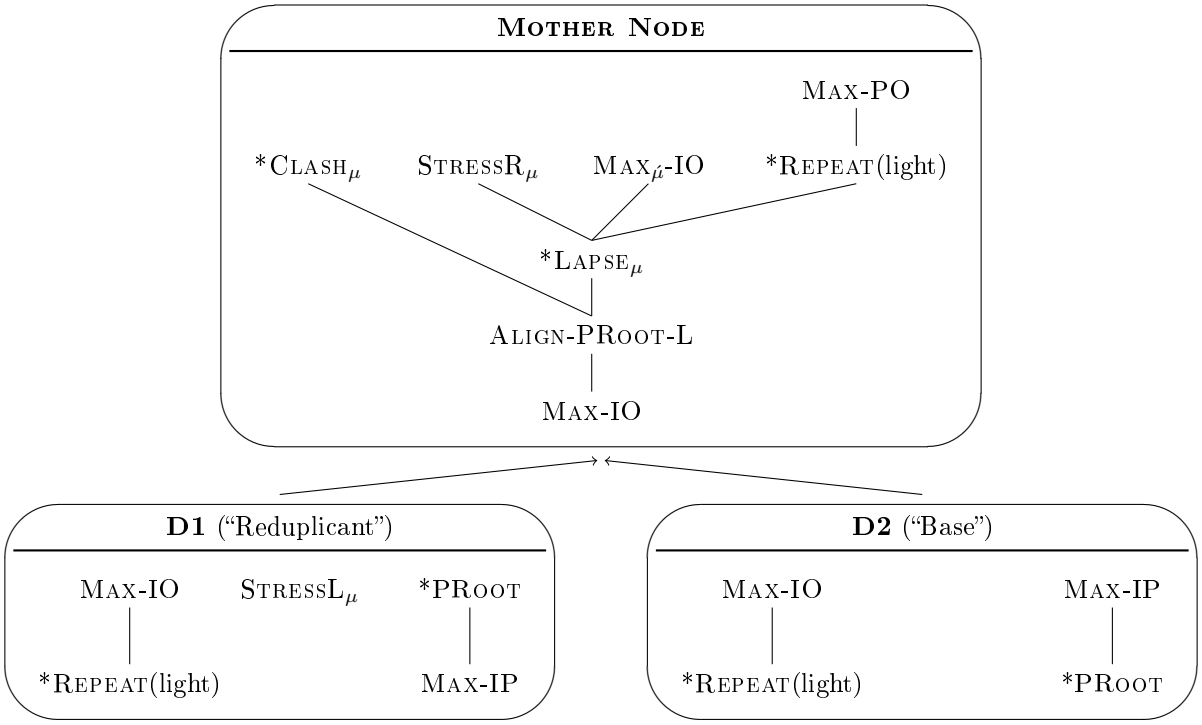
|rére-{rere}|			MAX-PO	*REPEAT(light)	*LAPSE$_\mu$	ALN-PRT-L	MAX-IO
a.	rè.re-{re.ré}	[20-01]		**!*	*	4	
b. ☞	rèr-{re.ré}	[20-01]		*	*	3	1
c.	rè-{re.ré}	[2-01]		**!		2	2
d.	rèr-{rér}	[20-1]	*!			3	2

MAX-PO blocks deletion in the base that could have avoided a *REPEAT(light) violation (60d), just like MAX-IO did in the BRCT analysis. To prevent other repairs in the base which may be tolerated in the reduplicant (i.e. feature change, lengthening, or epenthesis), we can employ other PO-faithfulness constraints as well. This is exactly the same kind of asymmetry between base and reduplicant that leads to TETU effects in BRCT and other theories that explicitly allow for base-dependence.

4.3 Ranking summaries and comparisons

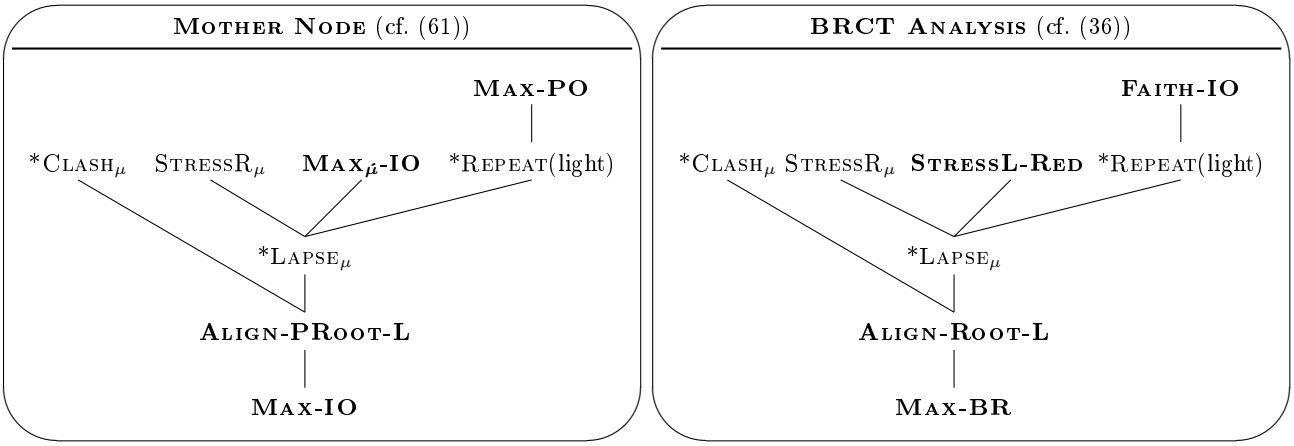
The rankings motivated by the MDT Mother Node analysis (coupled with a sketch of the constraints needed for the daughter nodes) are summarized in (61). The relative ranking of something like *PROOT (*don't have PRoots*) and MAX-IP is what would determine whether or not a node will parse any of its segments into a PRoot.

(61) *Ponapean MDT ranking summary*



From this we can now see that, in fact, the MDT Mother Node is precisely equivalent to the BRCT analysis, with a few constraint definitions changed in order to accommodate the differences between the theories:

(62) *Comparison of MDT's Mother Node with BRCT analysis*



This calls into question whether MDT's claim of being more restrictive in this domain is either accurate or desirable. If one were to deny the validity of the analysis just laid out, then, pending a viable alternative analysis, MDT suffers from undergeneration. If one accepts this analysis, then it is clear that MDT has the power to generate patterns that have all the hallmarks of base-dependent reduplicant shape. If we accept Ponapean to be such a pattern, then IZ's prediction about its absence is incorrect, as is their claim that MDT does not generate it. If one were to assert that this is not truly a base-dependent pattern because it can be captured entirely in the mother node, then it is hard to see how MDT's prediction regarding base-dependent reduplicant shape does not devolve into an unfalsifiable claim. Under any of these avenues, following also

Haugen & Hicks Kennard (2011), it is clear that the notion of base-dependence is a less reliable distinguishing factor between MDT and BRCT than previously assumed.

5 Tawala reduplication

In the preceding sections, we saw that the reduplication pattern in Ponapean was amenable to largely equivalent analyses in BRCT and MDT. While this undermined the claimed superiority of MDT in the domain of base-dependence, it does not stand as an argument for BRCT over MDT. In this section, I will lay out evidence from the cognate reduplication pattern in Tawala (Papua New Guinea, Western Oceanic, Oceanic; Ezard 1997), which has an even stronger base-dependent character and functions around a number of the same phonological principles. However, we will find that, in this case, the analysis diverges much more substantially across the two frameworks, in a way that accentuates the differences between the two. While a convergent MDT analysis is possible, it must make use of even more, and more powerful, machinery to handle what BRCT generates in a relatively simple parallel derivation. When we couple the powerful tools with the undermined predictions, MDT's claims of restrictiveness become harder to substantiate.

5.1 Tawala: a BRCT analysis

Hicks Kennard (2004) [henceforth HK] develops an atemplatic analysis of reduplication in Tawala. In this section, I revise this analysis slightly to account for two small problems. First, I show that an unconsidered candidate for the CVCV pattern requires a more articulated version of CONTIGUITY-BR (McCarthy & Prince 1995), which relativizes violations to consonants and vowels. Second, I show that 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, rather than a foot-sized reduplicant.

5.1.1 The data

The Tawala durative exhibits four distinct reduplicant shapes (63), whose distribution is phonologically predictable (see HK, based on Ezard 1980, 1997; see also Haugen & Hicks Kennard 2011 [HHK]). These patterns are schematized and exemplified in (64). (I omit stress marking for the time being, but will return to this apropos of the MDT analysis in Section 5.2.) I will first focus on Types A and B, because HK's analysis turns out to not quite work for them.

(63) *Reduplicant-shape alternations in Tawala*

Type A: $C_1V_1.V_2$-initial bases reduplicate C_1V_2
Type B: CVCV-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 V[11]

(64) *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]

[11] 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].

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

Type B is the most common root shape, and attests many examples (65). Many of these have morphologically complex bases, but this does not appear to have any impact on reduplication.

(65) *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 (66), but it must be admitted that there are just as many "exceptions" (67). Following the previous literature, I analyze only the Type A forms as being derived from the productive reduplicative grammar.

(66) *More examples of Type A reduplication* (Ezard 1980:147, 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'

(67) *Other reduplication patterns for CVV roots* (Ezard 1980:147, 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'

HK derives Type A primarily through the operation of two constraints. The first constraint is *Repeat (68), which bans adjacent identical syllables. I am going to use a more specific version of this constraint, *Repeat(initial) (69), for reasons which will become clear below. The other constraint is Align-Root-L (70), which prefers minimizing the length of the reduplicant, just as in the BRCT analysis of Ponapean above.

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

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

(70) **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)

Note that Tawala does not have long vowels or heavy syllables. This means that both *Repeat (minus the restriction to initial syllables) and Align-Root-L are effectively exactly equivalent to their Ponapean counterparts. All of the repetitions penalized by *Repeat in Tawala are light syllables, as with *Repeat(light) in Ponapean. All of the segments counted by Align-Root-L are single timing slots, as with the version of Align-Root-L used for Ponapean.

These constraints must outrank Contiguity-BR, the constraint requiring contiguous copying — defined categorically in (71a), but "gradiently" in (71b) — in order to permit Type A's discontiguous copying (72d). Contig-BR penalizes winning candidate (72d), because [bi] is not a contiguous substring of the base. In order for this candidate to be selected as optimal, this constraint must rank below both Align-Root-L, which prefers the shorter reduplicants in (72c,d) over the longer ones in (72a,b), and *Repeat(init), which penalizes (72c) for its initial repetition ($[be]_\sigma[be]_\sigma$).

(71) **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 $r_1 \ldots r_n$.

(72) *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	*

However, this ranking wrongly predicts discontiguous copying also for Type B, i.e. candidate (73d), which was not considered by HK (nor by Haugen & Hicks Kennard 2011). With the current constraints, there should be no difference in the constraint interaction; we should continue to select the C_1V_2- candidate (73d). But do note that we have a potential difference with respect to Contiguity: if we adopt (some version of) the "gradient" definition (71b), there's one more violation in (73d) than (72d).

(73) *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	*(*)

That is to say, the two patterns are distinguished by the nature of their (would-be) discontiguity. Type A skips only vowels (base V_1): *bi-b[e].i.ha*. On the other hand, for Type B, the problematic discontiguous candidate (73d) 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 (74) and vowels (75), in the mode of the gradient definition of Contiguity.

(74) **ContiguityC-B($\rightarrow$)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 $r_1 \ldots r_n$.

(75) **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 (76), we derive the right results (77, 78).

(76) **Ranking:** ContigC-BR ≫ Align-Root-L ≫ ContigV-BR

Splitting up Contig has no effect on Type A (77), because there's no medial consonant to skip, so the evaluation looks exactly the same as before. But for Type B (78), 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 (78c,d), and so the next shortest possible (C-contiguous) reduplicant (78b) is selected.

(77) *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	*

(78) *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	*

One other type of alternative candidate which could yield a short reduplicant without running afoul of *Repeat(init) and/or ContigC-BR would be one whose reduplicant does not include the initial syllable in the first place, as in (80b,c). We can rule these candidates out by ranking Anchor-L-BR (79), which requires copying from the left edge, above Align-Root-L (80).

(79) **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)

(80) *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

5.1.3 A note on stress and feet

HK includes a constraint requiring a left-aligned foot in reduplicated words. She asserts that this helps generate disyllabic copying in Type B. But, as can be seen from my revised analysis, this is not necessary. (It is also not clear whether it follows from or is truly necessary for her account to begin with.) The way in which it still may be relevant is that reduplicated words do show a divergent stress pattern. Tawala's typical stress pattern is as follows (Ezard 1997:44–45, HK:305–306):

(81) *Tawala stress*

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 (81b) is found in reduplication: there is a requirement that the initial syllable of the reduplicant be stressed, even if this leads to a lapse (82a–c) or a clash (82d).

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

Lapses in reduplication			
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'
Clashes in reduplication			
d.	**à**.p-á.pu	(*__a__.p-á.pu)	'be baking'

We could account for this with feet, by saying that a foot is constructed beginning on the first syllable of the reduplicant and terminating before the first syllable of the base. This would be a binary foot in cases like (82a–c) but a unary foot in cases like (82d). Therefore, foot binarity would not seem to be playing any role in determining reduplicant shape. But we can also do this straightforwardly using several of the foot-free stress constraints employed above for Ponapean, operating over syllables rather than moras:

(83) *Stress constraints*

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 (84, 85). But note from losing candidates (84c) and (85c), where the reduplicant is extended to avoid the stress problems, that, unlike in Ponapean, the stress constraints have no impact on what gets copied. This reiterates that the language is *not* treating the (binary) foot as the target shape in reduplication.

(84) *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!		

(85) *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!		

Before moving on, it is worth pausing briefly on the position of primary stress. As mentioned above (81), primary stress is normally penultimate. We can derive this through the interaction between *Lapse and NonFinality (*don't stress the final syllable*): one of the final two syllables must bear a stress, but it can't be the final, so it must be the penult. 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(s) driving this retraction — which I will shorthand as "Retract" — that constraint outranks *Lapse, allowing for stress retraction to the antepenult. Because of this retraction, we get a clash also in [bì-bé.i.ha]. It must therefore also be the case that "Retract" outranks *Clash, as shown in (86).

(86) *Retraction and clashes*

/RED, beiha/				ALIGN-ROOT-L	NONFIN	"RETRACT"	*CLASH	*LAPSE
a.	☞	bì-bé.i.ha	[2-100]	2			*	*
b.		bì-be.í.ha	[2-010]	2		*!		
c.		bì-be.i.há	[2-001]	2	*!			*
d.		bè.i-bé.i.ha	[20-100]	3!				*

5.1.4 Vowel-initial roots

The Type C pattern, where vowel-initial roots copy their initial VC- string, is exemplified further in (87):

(87) *More examples of Type B reduplication* (Ezard 1980:147, 1997:42)

Simplex		**Reduplicated (durative)**	
a.pu	→	*a.p-a.pu*	'to bake'
e.no	→	*e.n-e.no*	'to sleep'
a.ɱ	→	*a.m-a.ɱ*	'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'

This pattern follows completely from existing rankings. Among the shortest possible reduplicants, *REPEAT(init) rules out copying just the initial vowel (88c) and ANCHOR-L-BR rules out copying the base-second consonant (88d). Lower-ranked ALIGN-ROOT-L selects the next shortest properly-anchored reduplicant, which is the desired VC-copying candidate (88b).

(88) *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 (89) turns out to be unnecessary, even though we might have expected it to be responsible for eliminating (88a) and (88c), as it is in HK's analysis.

(89) **ONSET:** Assign one violation * for each onsetless syllable.
(HK:306; cf. Itô 1989, Prince & Smolensky [1993] 2004)

5.1.5 Type D reduplication, *REPEAT, and TETU

We've now used *REPEAT (or the more specific *REPEAT(init)) to account for both the lack of C_1V_1-reduplication in consonant-initial roots (Types A & B), and the lack of V_1- reduplication for vowel-initial roots (Type C). HK (followed by HHK) also uses it to help analyze Type D:

(90) *More examples of Type D reduplication* (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'

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 (91). None of the conceivable means of avoiding the repetition are employed, and the violation is tolerated. This means that the avoidance of repeated identical (initial) syllables in reduplication in Tawala is again an instance of TETU, as argued by HK and HHK.

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

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

HK and 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, i.e., infixation via violation of Align-Red-L (92). The complete analysis is shown in (93) (superscripts indicate correspondence).

(92) **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)

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

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

As can be seen in the tableau, there are a variety of candidates that might reasonably be considered for this derivation. One option is Type B (i.e. CVCV-) reduplication (93b.i). This candidate obviously incurs a *Repeat(init) violation, as well as three *Repeat violations for its four identical syllables in a row. Another option is Type A (i.e. C_1V_2-) reduplication (93b.iii). For roots of this sort, this strategy doesn't fix the *Repeat(init) violation either, 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], not shown in the tableau.) A third option is something like Type C (i.e. V_1-) reduplication (93a.iii), which could be accomplished by deleting the root-initial consonant. However, this fatally violates one of the Input-Output faithfulness constraints, namely Max-IO, which we know from (91) to rank above even *Repeat(init). Additionally, 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. As long as we have the crucial rankings in (94), we correctly derive candidate (93d.i) [g^1-$\underline{u^2}$.-u^2.g^3u^4.y^5a^6], which infixes a

copy of V_1 between base-C_1 and base-V_1. (CONTIGUITY-IO must also be low-ranked, since the base doesn't correspond to a contiguous input string.)

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

Note that, in order to prefer desired candidate (93d.i) over candidate (93c.i), 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 surely be broken, wrongly in favor of (93c.i), by lower-ranked ONSET, or (93c.ii), by lower-ranked MAX-BR.

As can be verified from the tableau in (93), 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:

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

	Underlying initial repetition	***Underlying non-initial repetition***
a.	***REPEAT(init) prediction:** V-doubling infixation to avoid *a word-initial* repetition	
	✓ /#$C_1V_1C_2V_2$(X)/ → [#C_1-$\underline{V_1}$-$V_1C_2V_2$(X)]	☒ /X$C_1V_1C_2V_2$(X)/ → [XC_1-$\underline{V_1}$-$V_1C_2V_2$(X)]
b.	***REPEAT prediction:** V-doubling infixation to avoid *any* repetition	
	✓ /#$C_1V_1C_2V_2$(X)/ → [#C_1-$\underline{V_1}$-$V_1C_2V_2$(X)]	☑ /X$C_1V_1C_2V_2$(X)/ → [XC_1-$\underline{V_1}$-$V_1C_2V_2$(X)]

There is at least one relevant base which can disambiguate between these two predictions: *kilolo* 'urinate' → *kilo-kilolo* 'urinating' (**kil-o-o-lo*) (Ezard 1997:61, HK:307). Since it shows prefixation (Type B reduplication) rather than infixation (Type D reduplication), we know that we must be dealing with *REPEAT(init), not general *REPEAT. We can see the argument most clearly if we try getting rid of *REPEAT(init), as in (96). Since the reduplicant is by default a prefix, we know that ALIGN-RED-L ≫ ALIGN-ROOT-L (cf., e.g., Zukoff 2022). From the Type D cases, we know that some version of *REPEAT must dominate ALIGN-RED-L. Implementing these rankings, we incorrectly derive infixation (96c):

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

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

This shows that the general *REPEAT constraint must rank below ALIGN-RED-L. But if this were the only active *REPEAT constraint, we would no longer be able to generate Type D infixation at all. Allowing *REPEAT(init) to rank high while general *REPEAT is ranked low (or non-existent), we derive the correct results for /kilolo/, as shown in (97). *REPEAT(init) correctly rules out $\underline{C_1V_1}$- reduplication (97b), but does not penalize retaining the underlying non-initial repetition in desired candidate (97a). This allows ALIGN-RED-L to eliminate the infixal candidate (97c).

(97) /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. ☞	$\underline{k^1i^2.l^3o^4}$.-$k^1i^2.l^3o^4.l^5o^6$			4	*
b.	$\underline{k^1i^2}$.-$k^1i^2.l^3o^4.l^5o^6$	*!		2	**
c.	$k^1i^2.l^3$-$\underline{o^4}$.-$o^4.l^5o^6$		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: *o.to.wi* 'make an appointment' →

o.t-o.to.wi (Ezard 1980:147; Inkelas & Zoll 2005:95, HHK:26).[12] In (98), we see that general *Repeat must be ranked *below* Align-Root-L, or else candidate (98b), which additionally copies V_2 to avoid the medial repetition, would be preferred to desired candidate (98a).

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

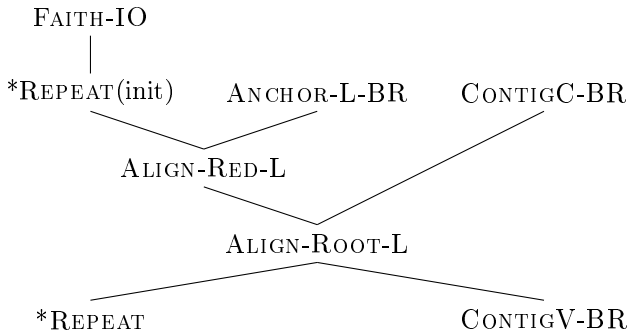
/red, otowi/		*Repeat(init)	Anch-L-BR	Aln-Red-L	Aln-Root-L	*Repeat
a.	o.to.-o.to.wi				3!	
b. ☞	o.t-o.to.wi				2	*
c.	o.-o.to.wi	*!			1	*
d.	t-o.to.wi	*!	*!		1	*
e.	w-o.to.wi		*!		1	
f.	o.t-o-o.wi			2!		

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.

5.1.6 Local summary

The rankings motivated above are summarized in (99):

(99) *Ranking summary for Tawala reduplication*



This analysis revises that of Hicks Kennard (2004) primarily in the following two ways. First, it splits Contiguity-BR into separate constraints, one banning consonant skipping, the other banning vowel skipping. This fixes HK's unrecognized prediction of Type A ($\underline{C_1V_2}$-) reduplication for $C_1V_1C_2V_2$-initial bases. Second, it restricts the active *Repeat constraint to *initial position*. This fixes HK's unrecognized prediction of Type D reduplicative infixation to repair non-base-initial repetitions, as well as $\underline{V_1C_2V_2}$- reduplication for vowel-initial roots where $V_1 = V_2$.

These changes allow us to understand the system as preferring the shortest possible reduplicant (just as in the Ponapean analysis in Section 2), 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.

Before moving on to the MDT analysis, there is one other point to be made regarding Contiguity. The relativized Contiguity approach proposed here not only solves the Type B copying problem, it clarifies HK's "gradient" evaluation of Contiguity. For example, in Type A forms, HK (p. 314) uses Contiguity-BR to prefer copying V_2 [bi-beiha] (100b) rather than V_3 *[ba-beiha] (100c), under the assumption that the latter incurs greater Contiguity violation.

[12] This form is cited in Ezard (1980), an early paper on reduplication in Tawala, but not in Ezard (1997), the subsequent Tawala grammar. HHK (p. 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.

(100) *Selecting the reduplicative vowel with gradient* CONTIGUITY (HK's approach)

/RED, beiha/		*REPEAT(init)	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, categorical definition (cf. (71a)), we actually shouldn't be able to distinguish between the two. On the other hand, my new definition, repeated here (cf. (71b, 74, 75) above), spells out a method for categorical violation assignment over multiple loci.

(101) *Traditional definition of* (INPUT) CONTIGUITY (HK:308; cf. McCarthy & Prince 1995:123):
Assign one violation * if the reduplicant doesn't correspond to a contiguous substring of the base.

(102) **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 (103c) because it skips any/all subsequent consonant(s). Even if we had an example with no subsequent consonants, e.g. a hypothetical root /beia/ (104), CONTIGV-BR would assign a fatal violation for skipping V_2 and any subsequent vowels (104c), because it counts up each locus of violation. This shows that this revised approach addresses multiple analytical questions within Tawala, as well as giving us traction on our theoretical understanding of CONTIGUITY.

(103) *Selecting the reduplicative vowel with relativized* CONTIGUITY

/RED, beiha/		*REPEAT(init)	CONTIGC-BR	CONTIGV-BR
a.	be-be.i.ha	*!		
b. ☞	bi-be.i.ha			*
c.	ba-be.i.ha		*! (h!)	** (e,i)

(104) CONTIGUITY *and hypothetical Type A root* /beia/

/RED, beia/		*REPEAT(init)	CONTIGC-BR	CONTIGV-BR
a.	be-be.i.a	*!		
b. ☞	bi-be.i.a			*
c.	ba-be.i.a			**! (e,i!)

5.2 Tawala: an MDT analysis

When comparing potential BRCT and MDT analyses of Ponapean earlier, we found that both theories could generate the data using essentially the same analysis, simply by translating between the appropriate constraint formulations. One reason that this was possible was because the BRCT analysis did not rely on any active BR-faithfulness constraints. Crucially, then, the faithfulness asymmetry between base and reduplicant in BRCT could be recapitulated by faithfulness to prosodic constituency, *ad hoc* as it may have been. While Ponapean was thus relevant for better understanding the concept of base-dependence, it was not substantially probative in distinguishing between the theories.

On the other hand, because the BRCT analysis of Tawala does lean heavily on BR-faithfulness constraints, there's no easy way to import it directly into MDT like we did for Ponapean. Nevertheless, in the remainder of this section, I will demonstrate that we can establish an internally consistent MDT analysis, again using the PRoot technology and distributing deletion across the different nodes. However, we will see that this analysis, even moreso than the MDT analysis proposed earlier for Ponapean, reveals the extreme power of the technology available to MDT. I will thus argue that BRCT is to be preferred here on the grounds of parsimony of analysis and typological restrictiveness. The bigger picture take-away will again be that MDT can in fact generate base-dependent reduplicant shape alternations.

5.2.1 Data review and analysis preview

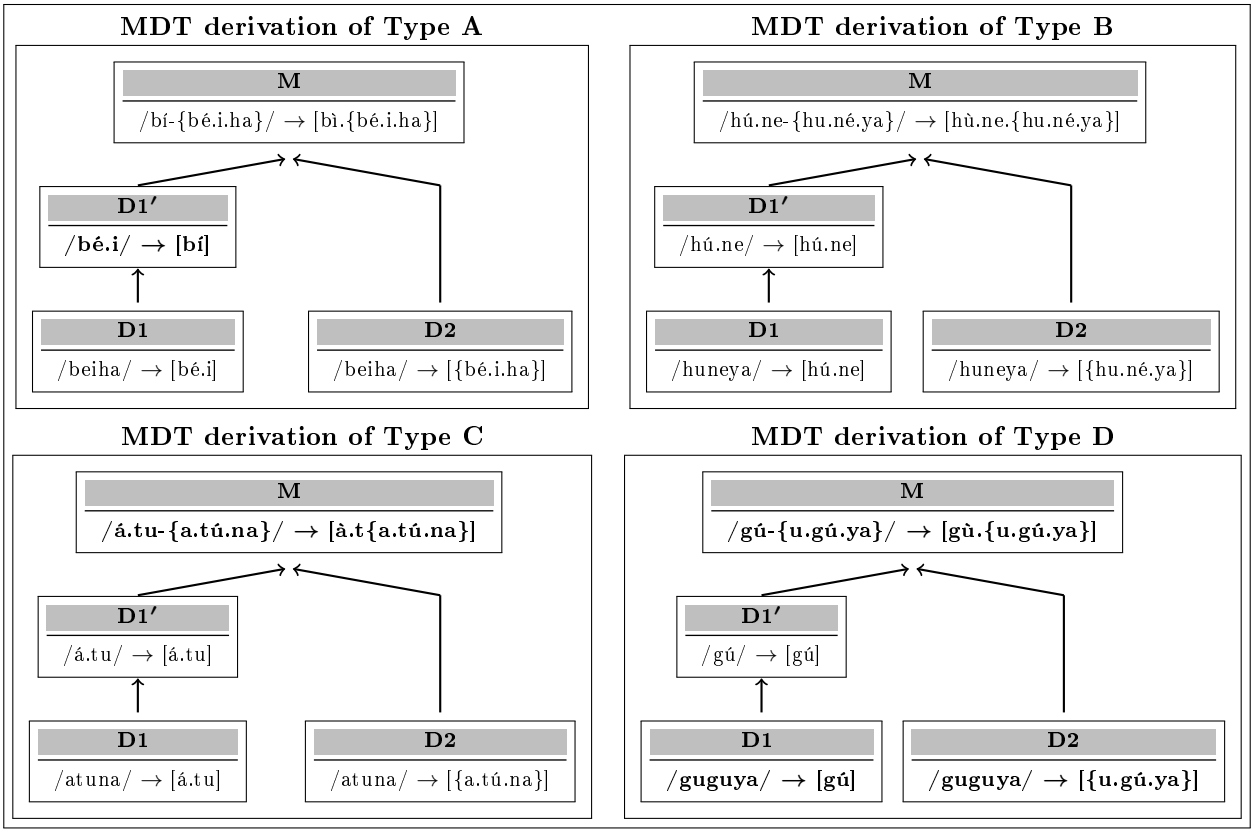
The table in (105) repeats the basic data patterns introduced above, now including stress markings, which will be significant for the MDT analysis. As discussed briefly in Section 5.1.3, primary stress generally falls on the penult (105B,C,D). The exception is when the penultimate vowel is in hiatus with the antepenultimate vowel, and that vowel is lower in height. In that case (105A), primary stress falls on the antepenult. The first syllable of the reduplicant always bears a secondary stress, even if this results in a clash.

(105) *Tawala data* (with stress marked)

	Base	Reduplicated
Type A:	*bé.i.ha*	→ *bì.bé.i.ha*
Type B:	*hu.né.ya*	→ *hù.ne.hu.né.ya*
Type C:	*a.tú.na*	→ *à.ta.tú.na*
Type D:	*gu.gú.ya*	→ *gù.u.gú.ya*

The derivations generated by the MDT analysis are summarized in (106). As before, D1 refers to the "reduplicant" node, D2 refers to the "base" node, and "M" refers to the mother node where the two constituents are concatenated. D1′ is an extra node, intermediate between D1 and M, which must be posited in order to fix a problem with Type A forms. The analysis has four main components: (i) truncation in D1; (ii) Prosodic Root ({...}) parsing in D2; (iii) hiatus resolution in M, conditionally blocked by FAITH; and (iv) V_1-deletion under hiatus in D1′.

(106) *Preview of MDT derivations*



I will now proceed through each step of the analysis, pointing out where crucial operations take place for the four different kinds of derivations.

5.2.2 Truncation in D1

The first step in the MDT analysis (following IZ:95, HHK) is that D1 preferentially truncates the input down to two syllables (107A,B,C). This is effectuated by the ranking in (108) (constraints defined in (109)), and demonstrated for Type B in (110). It is the combined effect of StressL and NonFinality that requires at least two syllables in the truncatum. That is to say, any candidate composed of a single syllable would have to violate one of the two constraints, since that syllable must either be stressed or unstressed. This also correctly places stress on the first syllable.

(107) *Output of D1*

	Input	[D1]
A.	*beiha*	→ *bé.i*
B.	*huneya*	→ *hú.ne*
C.	*atuna*	→ *á.tu*
D.	*guguya*	→ *gú*

(108) **Ranking for 2σ truncation:** StressL, NonFin ≫ *Struc ≫ Max

(109) *Stress constraints for Tawala MDT analysis*

a. **StressL:** Assign one violation * if the leftmost syllable is unstressed. (*[#$\breve{\sigma}$])
b. **NonFinality:** Assign one violation * if the rightmost syllable is stressed. (*[$\acute{\sigma}$#])
c. ***Struc:** Assign one violation * for each segment in the output.
d. **Max-IO:** Assign one violation * for each input segment w/o an output correspondent.

(110) *D1 derivation of Type B* (same for Types A & C)

/huneya/		StressL	NonFin	*Struc	Max-IO
a.	hú.ne.ya [100]			6!	
b. ☞	hú.ne [10]			4	2
c.	hú [1]		*!	2	4
d.	hu [0]	*!		2	4

However, Type D, the crucial case in HHK's argument, *is* truncated to a single syllable. This is effectuated by ranking *Repeat(init) (repeated in (111)) and StressL above NonFin. This ranking is shown in (112) and demonstrated for Type D in (113).

(111) ***Repeat(initial)**: Assign one violation * if the first two syllables are identical. (*[#$\sigma_\alpha\sigma_\alpha$])

(112) **Ranking for 1σ truncation in Type D:** StressL, *Repeat(init) ≫ NonFin

(113) *D1 derivation of Type D*

/gu_1gu_2ya/		StressL	*Repeat(init)	NonFin
a.	$gú_1.gu_2$.ya [100]		*!	
b.	$gú_1.gu_2$ [10]		*!	
c. ☞	$gú_1$ [1]			*
d.	gu_1 [0]	*!		

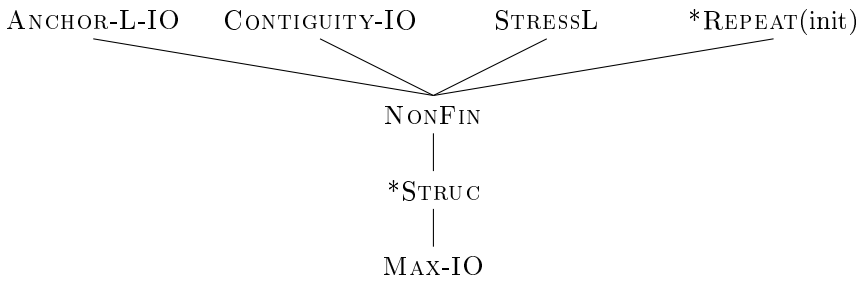
Two other faithfulness constraints are necessary to rule out some viable alternatives. As long as Anchor-L (114a) and Contiguity (114b) are undominated, the other 2σ candidates in (115) will be eliminated. The rankings for D1 are summarized below in (116).

(114) a. **Anchor-L-IO:** Assign one violation * if the leftmost segment in the input does not correspond to the leftmost segment in the output.
b. **Contiguity-IO:** Assign one violation * for each pair of adjacent segments in the output which were not adjacent in the input.

(115) *D1 derivation of Type D*

/gu$_1$gu$_2$ya/		ANCHOR-L	CONTIGUITY	NONFIN
a. ☞ gú$_1$ [1]				*
b. ú$_1$.gu$_2$ [10]	*!			
c. gú$_1$.ga [10]			*!	
d. gú$_1$.ya [10]			*!	
e. gú$_2$.ya [10]	*!			

(116) *Ranking summary for D1*



5.2.3 Type D C_1-deletion in D2

The action in D2 revolves around prosodic parsing. D2's cophonology parses all segments into an output Prosodic Root (cf. IZ:140; Downing 1998a,b; see Section 4 above). This is effectuated with the constraint MAX-INPUT-PROOT (117) (cf. (46e)), as shown in (118). Note that it is crucial that D1 not output a PRoot. Therefore, D1 needs to have a constraint against PRoot's in the output, e.g. *PROOT, which outranks MAX-IP. D2 needs to have the reverse ranking: MAX-IP ≫ *PROOT.

(117) **MAX-IP:** Assign one violation * for each input segment which does not have an output correspondent contained within a Prosodic Root.

(118) MAX-IP *induces PRoot parsing, e.g. for Type A*

/beiha/		MAX-IP
a.	bé.i.ha	5!
b.	{bé.i}.ha	2!
c. ☞	{bé.i.ha}	

The only constraint which forces violation of MAX-IP in D2 is *REPEAT(init). If *REPEAT(init) outranks MAX-IP *and* MAX-IO (119), we generate initial-C deletion for Type D only (120). Since there is not general truncation in D2, we know that *STRUC ranks below at least one of the MAX constraints.

(119) **Ranking for initial-C deletion in Type D:** *REPEAT(init) ≫ MAX-IP, MAX-IO

(120) *REPEAT*(init)*-driven deletion in D2 for Type D*

/gu$_1$gu$_2$ya/		*REPEAT(init)	MAX-IP	MAX-IO	*STRUC
a.	{gu$_1$.gú$_2$.ya}	*!			6
b.	g{u$_1$.gú$_2$.ya}	*!	*		6
c. ☞	{u$_1$.gú$_2$.ya}		*	*	5
d.	{gú$_2$.ya}		**!	**!	4

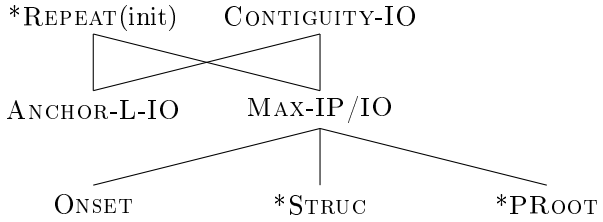
The fact that the *REPEAT(init) violation is repaired by *initial* deletion and not medial deletion shows that CONTIGUITY-IO ≫ ANCHOR-L-IO (121). At least the higher-ranked MAX constraint must also outrank ONSET, because initial-C deletion (121b) is preferred to initial-CV deletion (121c). The low ranking of ONSET in D2 is consistent with the fact that hiatus is tolerated stem-internally in D2 (e.g. {bé.i.ha}).

(121) *REPEAT*(init)*-driven deletion in D2 for Type D*

/gu$_1$gu$_2$ya/		*REPEAT(init)	CONTIG-IO	ANCHOR-L	MAX-IP/IO	ONSET
a.	{gu$_1$.gú$_2$.ya}	*!				
b. ☞	{u$_1$.gú$_2$.ya}			*	*	*
c.	{gú$_2$.ya}			*	**!	
d.	{gú$_1$.ya}		*!		**	

The rankings proposed for D2 are summarized in (122). D2 also enforces the language's normal stress pattern (see Ezard 1997): right-to-left trochees, i.e., penult primary stress + alternating stress leftward from the penult. I will not spell out the stress constraints here (see Section 5.1.3 above), except to note that STRESSL must be low-ranked, unlike in D1.

(122) *Ranking summary for D2*



5.2.4 Restricted hiatus resolution in M

The only process in M is deletion to repair hiatus, which occurs at the reduplicant-base juncture for Type C (126). This process can be modeled using the ranking ONSET ≫ MAX-IO, as ONSET penalizes the faithful hiatus candidate (126a). The second vowel cannot be deleted (126b) because it is underlyingly parsed into a PRoot, and thus protected by MAX-PO (123)(cf. (48)), the constraint protecting underlying PRoot segments. On the other hand, in this case, the reduplicant-final vowel is *not* protected by any such special faithfulness constraint, so the optimal repair is to delete it (126c).

(123) **MAX-PROOT-OUTPUT [MAX-PO]:** Assign one violation * for each vowel segment which was part of a PRoot in the input that does not have an output correspondent.

(124) **MAX$_{\acute{V}}$-IO:** Assign one violation * for each stressed vowel in the input that lacks a correspondent in the output.

(125) **Ranking for hiatus resolution in M:** MAX$_{\acute{V}}$-IO, MAX-PO ≫ ONSET ≫ MAX-IO[13]

(126) ONSET*-driven reduplicant-V$_2$ deletion in M for Type C*

/á.tu-{a.tú.na}/		MAX$_{\acute{V}}$-IO	MAX-PO	ONSET	MAX-IO
a.	à.tu.{a.tú.na}			**!	
b.	à.tu.{tú.na}		*!	*	*
c. ☞	à.t{a.tú.na}			*	*
d.	t{a.tú.na}	*!			*

The ranking ONSET ≫ MAX-IO would predict also deletion of reduplicant-initial vowels in Type C, namely candidate (126d). One way to avoid this outcome would be to include ANCHOR-L-IO ≫ ONSET in the rankings. Alternatively, we could use MAX$_{\acute{V}}$-IO (124), a constraint protecting stressed vowels, since the initial vowel will always be stressed. The reason to use this constraint is that it helps explain why hiatus is left unresolved in Type D (127).

Employing both MAX$_{\acute{V}}$-IO and MAX-PO in our analysis of hiatus means that, when we get to Type D, we find that both of the hiatal vowels are protected by special faithfulness. That is, deletion of the first vowel (127c) is blocked by MAX$_{\acute{V}}$-IO because it is stressed, and deletion of the second vowel (127b) is blocked by MAX-PO because it is underlyingly in a PRoot. For this reason, hiatus must be tolerated (127a) in Type D.

[13] If we assume that adjacency relations are established in the input across morpheme boundaries, then CONTIGUITY-IO would have to rank below ONSET as well to allow deletion in Type C.

MAX-PO also explains why there continues to be no hiatus-driven deletion base-internally in, e.g., Type A (128).

(127) *Hiatus tolerance across the juncture for Type D in M*

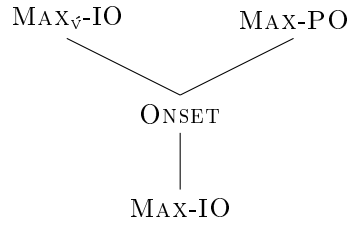
/gú-{u.gú.ya}/	MAX$_{\acute{V}}$-IO	MAX-PO	ONSET	MAX-IO
a. ☞ gù.{u.gú.ya}			*	
b. gù.{gú.ya}		*!		*
c. g{u.gú.ya}	*!			*

(128) *Hiatus tolerance base-internally for Type A in M*

/bí-{bé.i.ha}/	MAX$_{\acute{V}}$-IO	MAX-PO	ONSET	MAX-IO
a. ☞ bì.{bé.i.ha}			*	
b. bì.{bé.ha}		*!		*

The rankings proposed for M are summarized in (129). The only other piece of phonology in M is demotion of the reduplicant's underlying primary stress to secondary stress.

(129) **Ranking summary for M**



5.2.5 An extra node for Type A: Hiatus-resolution in D1′

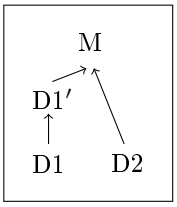
There is one problem remaining with this analysis with respect to Type A, which was suppressed in (128). Given that the output of D1 for Type A is [bé.i], we predict that the unstressed /i/ should delete under hiatus in M, yielding *[bè.{bé.i.ha}] (130b). If deletion of the /i/ were blocked by the fact that this would introduce an initial repetition (a *REPEAT(init) violation), the alternative would be *no deletion* [bè.i.{bé.i.ha}] (130a), not deletion of stressed /é/ (130c,d). And even if something could prefer /é/ deletion, we would still need something else to force the addition of stress to the /i/ in order to overcome (130c).

(130) *Incorrect prediction for Type A in M*

/bé.i-{bé.i.ha}/	(*RPT(init))	MAX$_{\acute{V}}$-IO	MAX-PO	ONSET	MAX-IO	DEP[stress]-IO
a. (💣) bè.i.{bé.i.ha}				*(!)		
b. 💣 bè.{bé.i.ha}	(*!)				*	
c. bi-{bé.i.ha}		*!			*	
d. ☹ bì-{bé.i.ha}		*!			*	*

The simplest solution (following HHK:15–16) is to introduce an additional, semantically vacuous node [D1′] (131) between D1 and M that maps /bé.i/ → [bí] (133).

(131) *MDT derivational structure with D1′*



The way to do this is as follows. ONSET must outrank MAX$_{\acute{V}}$-IO and MAX-IO (132a) to make sure stressed vowel deletion will be tolerated as a means of avoiding hiatus. ONSET must also outrank CONTIGUITY-IO to allow for the discontiguous mapping resulting from V_1-deletion, and outrank DEP[stress]-IO to allow for restressing the /i/ (132a). ANCHOR-R-IO must outrank MAX$_{\acute{V}}$-IO and CONTIGUITY-IO (132b) to prefer keeping the rightmost vowel instead of the stressed one, which is also the one that's adjacent to the stem-initial consonant. Lastly, STRESSL must outrank DEP[stress]-IO (132c) to favor inserting stress onto the newly leftmost /i/, and also outrank NONFIN (132c) (just as in D1) because this means that the final vowel will now be stressed.

(132) **Rankings in D1′ to fix Type A**

a. ONSET ≫ MAX$_{\acute{V}}$-IO, MAX-IO, CONTIGUITY-IO, DEP[stress]-IO [(133d) ≻ (133a)]
b. ANCHOR-R-IO ≫ MAX$_{\acute{V}}$-IO, CONTIGUITY-IO [(133d) ≻ (133b)]
c. STRESSL ≫ DEP[stress]-IO, NONFIN [(133d) ≻ (133c)]

(133) *Fixing Type A in D1′*

/bé.i/	ONSET	ANCHOR-R	STRESSL	MAXV́	CONTIGUITY	DEP[stress]-IO	NONFIN
a. bé.i	*!						
b. bé		*!					*
c. bi			*!	*	*		
d. ☞ bí				*	*	*	*

We'll also need ANCHOR-L-IO to outrank ONSET (134) in order to avoid deleting the stem-initial vowel in Type C (135).

(134) **Ranking for Type C in D1′:** ANCHOR-L-IO ≫ ONSET

(135) *Not messing up Type C in D1′*

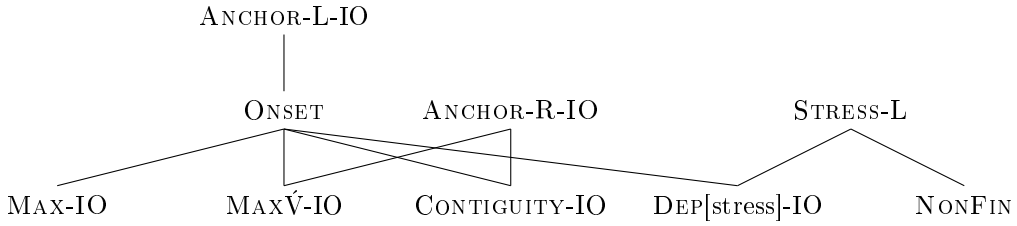
/á.tu/	ANCHOR-L	ONSET	ANCHOR-R	MAX$_{\acute{V}}$-IO	CONTIGUITY
a. ☞ á.tu		*			
b. tú	*!			*	

Once we implement these rankings in D1′, we achieve the desired result for Type A in M (136); in fact, there are no obvious competitor candidates. The rankings proposed for D1′ are summarized in (137).

(136) *Correct result for Type A in M*

/**bí**-{bé.i.ha}/	MAX$_{\acute{V}}$-IO	MAX-PO	ONSET	MAX-IO
a. ☞ bì-{bé.i.ha}				

(137) *Ranking summary for D1′*



5.3 Comparison and local summary

From Tawala, we can make roughly the same argument that we did from Ponapean. BRCT allows for a straightforward, parsimonious analysis of an intricate pattern with all the hallmarks of base-dependence. That is to say, the grammar seems to be actively adjusting the reduplicant in response to the surface conditions it finds itself in in the reduplicant+base string. Despite IZ's claims to the contrary, MDT appears to have the power to generate the whole pattern. It does this by positing highly specific, highly differentiated cophonologies across the different derivational nodes. These grammars include *ad hoc* prosodic constituency,

and are generally opaque. Furthermore, the most complex cophonology must be posited for a semantically vacuous derivational node (D1′), whose sole function is to get the analysis to work right.

It is clear that, in this case, BRCT is to be strongly preferred on the grounds of parsimony. The MDT analysis also raises serious questions about restrictiveness, specifically in regards to the use of vacuous nodes. If one objects to the technology used in the MDT analysis, then, pending an alternative, MDT suffers from undergeneration. And with respect to base-dependence, IZ's claimed predictions again must either be abandoned, or reduce to a tautology, namely, MDT predicts whatever is attested.

6 Discussion

This paper has laid out new analyses of the reduplicant shape alternations in Ponapean and Tawala. It has advocated for the revised BRCT analyses, which have the benefit of being parsimonious and minimalist, and using little in the way of new technology. The analytical take-aways are that stress and phonotactics can strongly dictate reduplicant shape using an a-templatic parallelist approach to reduplication.

This paper has also introduced workable MDT analyses of these same patterns. This is noteworthy because these patterns meet the *prima facie* criteria for being "base-dependent" reduplicant shape alternations, whereby the alternation appears to be driven entirely by surface factors. These analyses showed that, with creative use of prosodic constituency and other technologies, including vacuous derivational nodes, even base-dependent patterns can be generated by MDT. This runs counter to the claims made by Inkelas & Zoll (2005). The complexity of the MDT analyses, especially the one for Tawala, might lead us to view them with skepticism, particularly in light of worries about restrictiveness. Nevertheless, the alternative is undergeneration, which would be an even greater strike against MDT.

As mentioned, Inkelas & Zoll (2005) assert that MDT predicts the absence of base-dependent reduplication patterns, and they argue that such patterns do not exist. This paper has shown that, in the realm reduplicant shape alternations, neither of these claims seem to hold much water. Nevertheless, there are at least two other potential types of base-dependence in reduplication:

(138) *Other claimed types of base-dependence*

a. Prosodic constituent copying (Haugen & Hicks Kennard 2011)
b. Certain opaque reduplication-phonology interactions (Inkelas & Zoll 2005)

What Haugen & Hicks Kennard (2011) mean by prosodic constituent copying is a different kind of reduplicant shape alternation, where reduplicant shape co-varies with the syllabification and/or foot-structure of the base. The two patterns which Haugen & Hicks Kennard focus on are foot copying in Yidin^{y} (Haugen & Hicks Kennard 2011:7; McCarthy & Prince 1986) and syllable copying in Hiaki (Haugen & Hicks Kennard 2011:9; Haugen 2003). Haugen & Hicks Kennard (2011) argue that these patterns are only consistent with MDT if we allow *underlying* prosodic structure, which would require abandoning Richness of the Base (Prince & Smolensky [1993] 2004).

However, from what we have seen with the analyses of Ponapean and Tawala, and also Javanese, there are ways around this. Such cases in principle have at least two possible analyses. First, the reduplicative daughter nodes could be fed by a morphologically complex constituent (or simply a vacuous constituent) which has already built (morpho)prosodic structure. D1 then deletes everything outside of the (morpho)prosodic structure, such that only the target prosodic constituent is inherited by the mother node. Alternatively, D1 could build a morphoprosodic constituent which is constrained to be isomorphic with a syllable or a foot. All material is transmitted to the mother node, but the mother node deletes everything not protected by that morphoprosodic constituent. All of the languages discussed in this paper rely on analyses along these lines. Therefore, whatever arguments hold about reduplicant shape alternations generally also hold of those which can be characterized as prosodic constituent copying.

This leaves only reduplication-phonology interactions (Wilbur 1973) as a grounds on which to evaluate base-dependence. By reduplication-phonology interactions, we mean patterns that have been characterized as over-/under-application opacity, back-copying, re-copying, etc. The existence of these patterns is very much a live question in the field (e.g., McCarthy & Prince 1995, Inkelas & Zoll 2005, Kiparsky 2010, McCarthy, Kimper, & Mullin 2012). Inkelas & Zoll (2005) argue that none of these patterns exist as such. They take on many of the claimed instances of such patterns and provide alternative, frequently more comprehensive,

analyses that do not rely on the base-dependent interpretation proposed in previous literature. Many of these debates actually hinge on *empirical questions* rather than theoretical ones. Therefore, to the extent that base-dependence is a useful concept for distinguishing between theories of reduplication, we should continue to focus our attention on reduplication-phonology interactions, not reduplicant shape alternations.

References

Applegate, Richard B. 1972. Ineseño Chumash Grammar. PhD Dissertation, University of California, Berkeley.

———. 1976. Reduplication in Chumash. In Margaret Langdon & Shirley Silver (eds.), *Hokan Studies*, 271–284. The Hague: Mouton.

Blevins, Juliette & Andrew Garrett. 1992. Ponapean Nasal Substitution: New Evidence for Rhinoglottophilia. *BLS* 18:2–21.

———. 1993. The Evolution of Ponapeic Nasal Substitution. *Oceanic Linguistics* 32(2):199–236.

Booij, Geert. 1985. Coordination Reduction in Complex Words: A Case for Prosodic Phonology. In Harry van der Hulst & Norval Smith (eds.), *Advances in Nonlinear Phonology*, 143–160. Dordrecht: Foris.

Booij, Geert & Rochelle Lieber. 1993. On the Simultaneity of Morphological and Prosodic Structure. In Sharon Hargus & Ellen M. Kaisse (eds.), *Studies in Lexical Phonology* (Phonetics and Phonology 4), 23–44. San Diego: Academic Press.

Cole, Jennifer. 1994. A Prosodic Theory of Reduplication. PhD Dissertation, Stanford University.

Davis, Stuart. 2003. The Interaction of Nasal Substitution and Reduplication in Ponapean. In T.A. Hall & Silke Hamann (eds.), *Papers in Phonology & Phonetics* (ZAS Papers in Linguistics 32), 31–46.

Downing, Laura J. 1998a. On the Prosodic Misalignment of Onsetless Syllables. *Natural Language & Linguistic Theory* 16(1):1–52. doi:10.1023/A:1005968714712.

———. 1998b. Prosodic Misalignment and Reduplication. In Geert Booij & Jaap van Marle (eds.), *Yearbook of Morphology 1997*, 83–120. Kluwer. doi:10.1007/978-94-011-4998-3_4.

Dudas, Karen Marie. 1976. The Phonology and Morphology of Modern Javanese. PhD Dissertation, University of Illinois, Urbana-Champaign.

Elenbaas, Nine & René Kager. 1999. Ternary Rhythm and the Lapse Constraint. *Phonology* 16(3):273–329. doi:10.1017/S0952675799003772.

Ezard, Bryan. 1980. Reduplication in Tawala. *Kivung: Journal of the Linguistic Society of Papua New Guinea* 12(2):145–160.

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

Goodman, Beverley D. 1995. Features in Ponapean Phonology. PhD Dissertation, Cornell University.

Gordon, Matthew. 2002. A Factorial Typology of Quantity-Insensitive Stress. *Natural Language & Linguistic Theory* 20(3):491–552. doi:10.1023/A:1015810531699.

Haugen, Jason D. 2003. Allomorphy in Yaqui Reduplication. In Louis M. Barragan & Jason D. Haugen (eds.), *Studies in Uto-Aztecan* (MIT Working Papers on Endangered and Less Familiar Languages 5), 75–103. Cambridge, MA: MITWPL.

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.

Horne, Elinor Clark. 1961. *Beginning Javanese*. New Haven: Yale University Press.

Hurlbut, Hope M. 1988. *Verb Morphology in Eastern Kadazan*. Canberra: Pacific Linguistics.

Hyde, Brett. 2012. Alignment Constraints. *Natural Language & Linguistic Theory* 30(3):789–836. doi:10.1007/s11049-012-9167-3.

Inkelas, Sharon. 1990. *Prosodic Constituency in the Lexicon*. New York: Garland Publishing.

Inkelas, Sharon, Cemil Orhan Orgun & Cheryl Zoll. 1997. The Implications of Lexical Exceptions for the Nature of Grammar. In Iggy Roca (ed.), *Constraints and Derivations in Phonology*, 542–551. Oxford: Clarendon Press.

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.

Kennedy, Robert. 2002. A Stress-Based Approach to Ponapean Reduplication. In Gina Garding & Mimu Tsujimura (eds.), *Proceedings of the 21st West Coast Conference on Formal Linguistics*, 222–235. Somerville, MA: Cascadilla Press.

———. 2003. Confluence in Phonology: Evidence from Micronesian Reduplication. PhD Dissertation, University of Arizona.

Kiparsky, Paul. 2010. Reduplication in Stratal OT. In Linda Uyechi & Lian Hee Wee (eds.), *Reality Exploration and Discovery: Pattern Interaction in Language & Life*, 125–142. Stanford: CSLI.

Kurisu, Kazutaka. 2013. Nested Derivedness in Ponapean Morphophonology. *Lingua. International review of general linguistics. Revue internationale de linguistique générale* 137:106–127.

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

McCarthy, John J. 2011. Perceptually Grounded Faithfulness in Harmonic Serialism. *Linguistic Inquiry* 42(1):171–183.

McCarthy, John J., Wendell Kimper & Kevin Mullin. 2012. Reduplication in Harmonic Serialism. *Morphology* 22(2):173–232.

McCarthy, John J. & Alan Prince. 1986. Prosodic Morphology. *Linguistics Department Faculty Publication Series* 13 (1996 version).

———. 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).
———. 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.
———. 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.
———. 1999. Faithfulness and Identity in Prosodic Morphology. In René Kager, Harry van der Hulst & Wim Zonneveld (eds.), *The Prosody-Morphology Interface*, 218–309. Cambridge: Cambridge University Press.
Nespor, Marina & Irene Vogel. 1986. *Prosodic Phonology*. Dordrecht: Foris.
Orgun, Cemil Orhan. 1994. Monotonic Cyclicity and Optimality Theory. In Mercè Gonzàlez (ed.), *Proceedings of the North-eastern Linguistic Society 24*, 461–474. Amherst, MA: Graduate Linguistics Student Association.
———. 1996. Sign-Based Morphology and Phonology with Special Attention to Optimality Theory. PhD Dissertation, University of California, Berkeley.
Prince, Alan. 1983. Relating to the Grid. *Linguistic Inquiry* 14(1):19–100.
Prince, Alan & Paul Smolensky. [1993] 2004. *Optimality Theory: Constraint Interaction in Generative Grammar*. Malden, MA: Blackwell Publishing.
Rehg, Kenneth L. 1984. Nasal Substitution Rules in Ponapean. In Bryon W. Bender (ed.), *Studies in Micronesian Linguistics*, 317–337. Canberra: Pacific Linguistics.
———. 1993. Proto-Micronesian Prosody. In Jerold A. Edmondson & Kenneth J. Gregerson (eds.), *Tonality in Austronesian Languages* (Oceanic Linguistics Special Publication 24), 25–46. Honolulu: University of Hawaii Press.
Rehg, Kenneth L. & Damian G. Sohl. 1981. *Ponapean Reference Grammar*. Honolulu: University of Hawaii Press.
Riggle, Jason. 2006. Infixing Reduplication in Pima and its Theoretical Consequences. *Natural Language & Linguistic Theory* 24(3):857–891. doi:10.1007/s11049-006-9003-8.
Schachter, Paul & Fe T. Otanes. 1972. *Tagalog Reference Grammar*. Berkeley: University of California Press.
Selkirk, Elisabeth. 1984. *Phonology and Syntax: The Relation Between Sound and Structure*. Cambridge, MA: MIT Press.
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.
Sproat, Richard. 1986. Malayalam Compounding: A Non-stratum Ordered Account. In Mary Dalrymple, Jeffrey Goldberg & Kristin Hanson (eds.), *Proceedings of the Fifth West Coast Conference on Formal Linguistics*, 268–288. Stanford: Stanford Linguistics Association.
Stanton, Juliet. 2014. A Cyclic Factorial Typology of Pama-Nyungan Stress. In John Kingston, Claire Moore-Cantwell, Joe Pater & Robert Staubs (eds.), *Supplemental Proceedings of the 2013 Annual Meeting on Phonology*, Washington, D.C.: Linguistic Society of America.
———. 2015. Factorial Typology and Accentual Faithfulness. In Ulrike Steindl, Thomas Borer, Huilin Fang, Alfredo García Pardo, Peter Guekguezian, Brian Hsu, Charlie O'Hara & Iris Chuoying Ouyang (eds.), *WCCFL 32: Proceedings of the 32nd West Coast Conference on Formal Linguistics*, 54–63. Somerville, MA: Cascadilla Proceedings Project.
Steriade, Donca. 2001. The Phonology of Perceptibility Effects: The P-Map and its Consequences for Constraint Organization.
Sumukti, Rukmantoro Hadi. 1971. Javanese Morphology and Phonology. PhD Dissertation, Cornell University.
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.
Wash, Barbara. 1995. Productive Reduplication in Barbareño Chumash. MA Thesis, University of California, Santa Barbara.
Wilbur, Ronnie Bring. 1973. The Phonology of Reduplication. PhD Dissertation, University of Illinois, Urbana-Champaign.
Yip, Moira. 1995. Repetition and its Avoidance: The Case of Javanese. Ms., University of California, Irvine.
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.
———. 2017. Indo-European Reduplication: Synchrony, Diachrony, and Theory. PhD Dissertation, MIT. https://www.samzukoff.com/zukoffdiss.
———. 2020a. Reduplicant Shape Alternations in Ponapean. Paper Presented at LSA 2020, New Orleans, LA. January 2–5, 2020.
———. 2020b. Reduplicant Shape Alternations in Ponapean: Evidence Against Morphological Doubling Theory. In Mariam Asatryan, Yixiao Song & Ayana Whitmal (eds.), *NELS 50: Proceedings of the North East Linguistic Society*, vol. 3, 279–288. Amherst, MA: Graduate Linguistics Student Association. https://www.samzukoff.com/nelspaper2020.
———. 2022. The Mirror Alignment Principle: Morpheme Ordering at the Morphosyntax-Phonology Interface. *Natural Language & Linguistic Theory*. doi:10.1007/s11049-022-09537-2.

A Appendix: An alternative MDT analysis of Ponapean: Truncation in D1 + deletion in M

The Ponapean data is repeated in (139). The logic behind this alternative MDT analysis of Ponapean is as follows. First, D1 truncates the input down to the leftmost two moras, placing stress on the initial mora.

Second, D2 applies the regular stress pattern, as described/analyzed in Section 2.1. Lastly, M deletes the reduplicant-final mora if doing so alleviates a moraic lapse, unless doing so would create a sequence of adjacent identical light syllables. Crucially, repair of the base to satisfy *Repeat(light) is blocked because it would inevitably change the stress, which M does not allow.

(139) *Ponapean reduplication: length alternations* (cf. (8))

	Odd	Even	Odd	Even
	1-mora base	**2-mora base**	**3-mora base**	**4-mora base**
1-mora reduplicant		**dù**-duúp **là**-laúd **kè**-keńs		**tò**-toò.roór **lù**-luù.m^{w}uúm^{w} **sò**-soù.pi.sék
2-mora reduplicant	**pàa**-pá **tè.pi**-tép **dòn**-dód	**dùn**-du.né **sì.pi**-si.péd **rèr**-re.ré	**dùu**-dùu.pék **mèe**-mèe.lél **lìl**-lì.ne.nék	**rìi**-ri.àa.lá **lìl**-li.ròo.ró **lì.di**-li.dù.wií

A.1 Truncation in D1

Under this analysis, which is is precisely parallel to the MDT analysis of Tawala in Section 5.2.2, substituting moras for syllables, the target shape for the reduplicant, i.e. the output of D1, is two moras. By requiring a stressed initial mora, via undominated StressL$_\mu$ (140a), and an unstressed final mora, via undominated NonFinality$_\mu$ (140b), we eliminate the possibility of a monomoraic output (143e,f) because these constraints would make incompatible demands of this mora. The initial stress placed by StressL$_\mu$ will be crucial in further truncating down to one mora in the special case (which was the default case in the BRCT analysis) later in the derivation. If these stress constraints outrank *Struc[X] (141a), which itself outranks Max (141b), we generate deletion of everything but the two moras required for stress. The derivation is shown in (143).

(140) *Stress constraints for D1*

a. **StressL$_\mu$**: Assign one violation * if the initial mora is unstressed. (*#$\breve{\mu}$)
b. **NonFinality$_\mu$**: Assign one violation * if the final mora is stressed. (*$\acute{\mu}$#)

(141) *Motivating truncation in D1*

a. ***Struc[X]:** Assign one violation * for each timing slot in the output.
b. **Max:** Assign one violation * for each input segment without a correspondent in the output.

(142) **Ranking:** *Struc[X] ≫ Max

(143) *Truncation to 2 moras in D1:* /dune/ → |dún|

/dune/			StressL$_\mu$	NonFin$_\mu$	*Struc[X]	Max
a.	dú.ne	[10]			****!	
b.	du.né	[01]	*!	*!	****	
c. ☞	dún$_\mu$	[10]			***	*
d.	duń	[01]	*!	*!	***	*
e.	dú	[1]		*!	**	**
f.	du	[0]	*!		**	**
g.	d	[0]	*!		*	***
h.	ú	[1]		*!	*	***
i.	[Ø]	[0]	*!			****

The truncatum is always a contiguous string from the left edge. We can derive this using the following constraints (which have obvious analogs in the BRCT analysis):

(144) *Faithfulness constraints in D1*

a. **Contig:** Assign one violation * if the output corresponds to a discontiguous input string.
b. **Anchor-L:** Assign one violation * if the leftmost output segment doesn't correspond to the leftmost input segment.
c. **Anchor-R:** Assign one violation * if the rightmost output segment doesn't correspond to the rightmost input segment.

(145) **Ranking:** Contig, Anchor-L, *Struc[X] ≫ Anchor-R

(146) *Deriving the correct 2 mora string:* /dune/ → |dún|

/dune/			Contig	Anchor-L	*Struc[X]	Anchor-R	Max
a.	dú.ne	[10]			****!	*	
b. ☞	dún$_\mu$	[10]			***	*	*
c.	dú.e	[10]	*!		***		*
d.	ú.ne	[10]		*!	***		*

D1 will also require final moraic consonants, contra D2 and M. This is accomplished by having Weight-by-Position (147a) outrank the constraints against moraic consonants (147b,c). As long as *Struc[X] also outranks the constraints against moraic consonants (148), we derive the desired result in a case like this, which is to map the post-vocalic consonant to a moraic coda (149).

(147) *Consonant moraicity constraints in D1*

a. **Weight-by-Position [WxP]:** Assign one violation * for each non-moraic coda consonant.
b. ***C$_\mu$#:** Assign one violation * for each final moraic consonant.
c. ***C$_\mu$:** Assign one violation * for each moraic consonant.

(148) **Ranking:** WxP, *Struc[X] ≫ *C$_\mu$#, *C$_\mu$

(149) *Generating moraic codas in D1:* /dune/ → |dún|

/dune/			WxP	*Struc[X]	*C$_\mu$#	*C$_\mu$	Max
a.	dú.ne	[10]		****!	*		
b. ☞	dún$_\mu$	[10]		***	*	*	*

While the D1 outputs sometimes surface faithfully in the ultimate reduplicated word output (as in the preceding example), this is not always the case (e.g. (150,151)). That is to say, under certain conditions, further truncation will occur in M. Nevertheless, the D1 output always follows this grammar.

(150) /laud/ → |láu| (cf. [**là**-laúd])

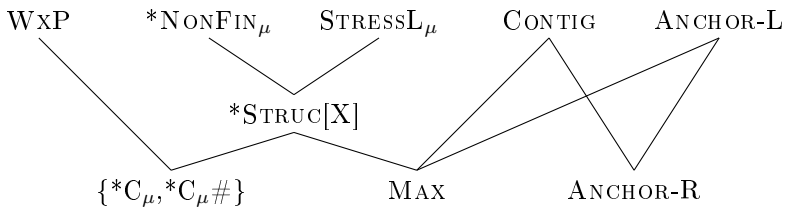
/laud/			NonFin$_\mu$	*Struc[X]	Max
a.	láud$_\mu$	[100]		****!	
b.	láud	[10]		****!	
c. ☞	láu	[10]		***	*
d.	lá	[1]	*!	**	**

(151) /duup/ → |dúu| (cf. [**dù**-duúp])

/duup/			NonFin$_\mu$	*Struc[X]	Max
a.	dúup$_\mu$	[100]		****!	
b.	dúup	[10]		****!	
c. ☞	dúu	[10]		***	*
d.	dú	[1]	*!	**	**

The rankings required for D1 are summarized in (152):

(152) *Ranking summary for D1*



A.2 Lapse-driven "reduplicant" reduction in M

One thing that drives further truncation in M is a desire to optimize stress. Both D2 and M apply the regular stress grammar (see Section 2.1): primary stress on the penultimate mora, alternating secondary stress by mora to the left. (M additionally converts non-final "underlying" primary stresses to secondary stresses; this doesn't violate the Ident[stress] constraint I use below.) The crucial property of M is that *Lapse$_\mu$ dominates Max (154). What this means is that, when the |$\acute{\mu}\mu$| output of D1 happens to be concatenated with an *even mora count base* (output of D2), deletion/shortening will be triggered.

(153) *Stress pattern* (repeated from (4) above)

a. Primary stress on rightmost mora — [StressR$_\mu$ (5a)]
b. R→L alternating secondary stress by mora — [*Clash$_\mu$ (5b), *Lapse$_\mu$ (5c)]
c. Medial coda C's are moraic, final coda C's are not — [*C$_\mu$# (6a) ≫ WxP (6b)]

(154) **Ranking:** *Lapse$_\mu$ ≫ Max

Getting deletion to work right is (relatively) unproblematic. Contig will prefer deleting the "reduplicant"-final moraic segment (155c) over a "base"-internal moraic segment (155d). This requires assuming that adjacency/precedence relations are not present in the M-input for segments belonging to different daughter outputs (a point also made by HHK with respect to Tawala).

(155) *Reduplicant diphthong reduction with even mora bases:* /laud/ → |láu-laúd| → [là-laúd]

|láu|-|laúd|		Ident[stress]	*Lapse$_\mu$	Contig	Max
a.	làu-laúd [20-01]		*!		
b.	laù-laúd [02-01]	*!*			
c. ☞	là-laúd [2-01]				*
d.	làu-lúd [20-1]			*!	*
e.	lù-laúd [2-01]	*!		*!	*

It is less straightforward how to handle the reduplicant vs. base question for cases where an input long vowel needs to get shortened (156).

(156) /duup/ → |dúu-duúp| → [dù-duúp]

|dúu|-|duúp|		Ident[stress]	*Lapse$_\mu$	Ident[long]
a.	dùu-duúp [20-01]		*!	
b.	duù-duúp [02-01]	*!		
c. ☞	dù-duúp [2-01]			*
d. 💣	dùu-dúp [20-1]			*

We might be able to get some traction out of thinking very carefully about moraic parsing. If we assume that the constraints in (157) are ranked as in (158), we can generate the desired outcomes, as shown in (159) below.

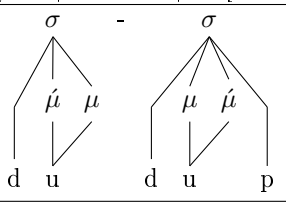
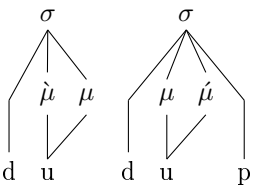
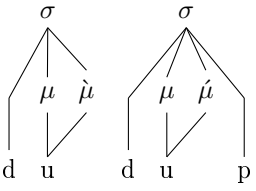
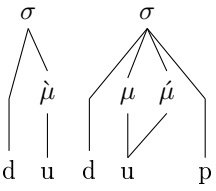
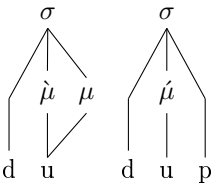
(157) *Moraic faithfulness constraints*

a. **MAX-μ_1:** Assign one violation * for deleting the leftmost underlying mora of a syllable.

b. **IDENT[stress]-μ:** Assign one violation * for each output mora whose corresponding input differs for [±stress] (stress degree irrelevant).

(158) **Ranking:** IDENT[stress], *LAPSE$_\mu$, MAX-μ_1 ≫ MAX-μ

(159) /duup/ → |dúu-duúp| → [dù-duúp]

		IDENT[stress]	*LAPSE$_\mu$	MAX-μ_1	MAX-μ
					
a. 	[20-01]		*!		
b. 	[02-01]	*!*			
c. ☞ 	[2-01]				*
d. 	[20-1]			*!	*

*LAPSE$_\mu$ penalizes the faithful candidate (159a), because there is a lapse between the reduplicant-initial mora and the base-peninitial mora. Candidate (159b), which switches stress from the first to the second in order to avoid the lapse fatally violates IDENT[stress]. The remaining options both delete a mora: candidate (159c) from the reduplicant, candidate (159d) from the base. Since the reduplicant's underlyingly unstressed mora is the second in its syllable, but the base's is the first, deletion IDENT[stress]-respecting deletion will violate MAX-μ_1 only in the base. Therefore, MAX-μ_1 correctly selects reduplicant mora deletion (159c).

A.3 *REPEAT(light) blocks lapse-driven deletion

The preceding analysis predicts that all lapses will be repaired by deleting the rightmost mora of the reduplicant. However, as mentioned earlier, this does not happen if it would result in repeated identical light syllables across the juncture:

(160) /dune/ → |dún-du.né| → [dùn-du.né] (with medial lapse)

|dún|-|du.né|		IDENT[stress]	*LAPSE$_\mu$	MAX-μ_1	MAX-μ
a. ☹	dùn$_\mu$-du.né [20-01]		*!		
b.	duǹ$_\mu$-du.né [02-01]	*!*			
c. 💣	dù-du.né [2-01]				*
d.	dùn$_\mu$-né [20-1]			*!	*

If *REPEAT(light) outranks *LAPSE$_\mu$, this will rule out problematic candidate (160c). We also need MAX-μ_1 to outrank *LAPSE$_\mu$ (which is consistent with (159)), so that (160a) ≻ (160d).

(161) **REPEAT(light) blocks lapse avoidance:* /dune/ → |dún-du.né| → [dùn-du.né] (with medial lapse)

|dún|-|du.né|		IDENT[stress]	*REPEAT(light)	MAX-μ_1	*LAPSE$_\mu$	MAX-μ
a. ☞	dùn$_\mu$-du.né [20-01]				*	
b.	duǹ$_\mu$-du.né [02-01]	*!*				
c.	dù-du.né [2-01]		*!			*
d.	dùn$_\mu$-né [20-1]			*!		*

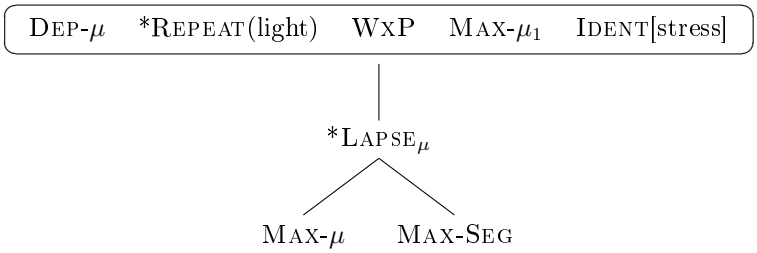
The lapse cannot be avoided by simply deleting the underlying mora attached to reduplicant-final |n| without deleting the segment (162b) because word-internal codas must be moraic (WEIGHT-BY-POSITION ≫ *LAPSE$_\mu$). Also, we can rule out having the lapse be avoided by inserting a (stressed) mora in the base (162d) by having DEP-μ outrank *LAPSE$_\mu$.

(162) /dune/ → |dún-du.né| → [dùn-du.né] (with medial lapse)

|dún|-|du.né|		DEP-μ	*REPEAT(light)	WXP	*LAPSE$_\mu$	MAX-μ
a. ☞	dùn$_\mu$-du.né [20-01]				*	
b.	dùn-du.né [2-01]			*!		*
c.	dù-du.né [2-01]		*!			*
d.	dùn$_\mu$-d**ù**u.né [20-201]	*!				

The rankings required thus far for M (excluding those needed only for the stress pattern) are summarized in (163).

(163) *Ranking summary for M*



A.4 *REPEAT(light) in the base

In this analysis, *REPEAT(light) *blocks* deletion, rather than triggering it. Nevertheless, its high position in the ranking — above MAX-μ and MAX-Segment (via transitivity) — could in theory be enough to trigger repairs inside bases containing repetitions, contrary to fact. But it turns out that the faithfulness constraints that we need in order to derive the correct outcomes across the juncture independently rule out changes to the base. As long as DEP-μ, IDENT[stress], and MAX-μ_1 all outrank *REPEAT(light) (consistent with (162)), and also ONSET outranks *REPEAT(light), all possible repairs are ruled out:

(164) /rere/ → |rér-re.ré| → [rér-re.ré] (with medial lapse)

|rér|-|re.ré|			Dep-μ	Id[stress]	Max-μ_1	Ons	*Repeat	*Lapse$_\mu$	Max-μ
a. ☞	rèr.-re.ré	[20-01]					*	*	
b.	reȅr.-re.ré	[02-01]		*!*			*		
c.	rè.-re.ré	[2-01]					**!		*
d.	rèr.-ré	[20-1]			*!				*
e.	rèr.-rér	[20-1]		*!	*!				*
f.	rè.r-e.ré	[2-01]					**!		*
g.	rèr.-e.ré	[20-01]				*!		*	
h.	rèr.-re.é	[20-01]				*!		*	
i.	rèr.-r**è**e.ré	[20-201]	*!						

We also must have all relevant featural Ident constraints, e.g. Ident[lateral], outrank *Repeat(light), so that the violation(s) cannot be avoid by changing features:

(165) /rere/ → |rér-re.ré| → [rèr-re.ré] (with medial lapse)

|rér|-|re.ré|			Ident[lateral]	*Repeat(light)	*Lapse$_\mu$	Max-μ
a. ☞	rèr.-re.ré	[20-01]		*	*	
b.	rèr.-re.**l**é	[20-01]	*!		*	
c.	rè.-re.ré	[2-01]		**!		*
d.	rè.-**l**e.ré	[2-01]	*!			*

However, this poses a serious problem: in order to account for the process of "nasal substitution", which repairs impermissible medial clusters arising across the reduplicant-base juncture, certain Ident[feature] constraints must be violated. Take, for example, the 3-mora base [lì.ne.nék]: the reduplicant final |n| changes to [l] because [nl] sequences are disallowed by the language's version of CodaCondition. Deleting to avoid the CodaCond violation is ruled out because it would create a clash (violating undominated *Clash$_\mu$).

(166) /linenek/ → |lín-lì.ne.nék| → [lì**l**-lì.ne.nek] (feature change across juncture)

|lín-lì.ne.nék|			*Clash$_\mu$	CodaCond	Id[lat]	*Repeat	*Lapse$_\mu$	Max-μ
a.	lìn.-lì.ne.nék	[20-201]		*!				
b. ☞	lìl.-lì.ne.nék	[20-201]			*			
c.	lì.-lì.ne.nék	[2-201]	*!			*		*

On the other hand, the 4-mora base [li.ròo.ró] can't rely on *Clash$_\mu$ to prevent deletion as a repair for CodaCond. The ranking we needed in order to avoid a base repair for /rere/ (Ident[lateral] ≫ *Repeat(light)) now predicts reduplicant-final deletion to satisfy CodaCond (167c).

(167) /lirooro/ → |lír-li.ròo.ró| → [lí**l**-li.ròo.ró] (feature change across juncture)

|lír-li.ròo.ró|			CodaCond	Max-μ_1	Id[lat]	*Repeat	*Lapse$_\mu$	Max-μ
a.	lìr.-li.ròo.ró	[20-0201]	*!				*	
b. ☹	lìl.-li.ròo.ró	[20-0201]			*!		*	
c. 💣	lì.-li.ròo.ró	[2-0201]				*		*
d.	lì.-ròo.ró	[2-01]		*!				*

The most straightforward solution is to specify that the high-ranked Ident[feature] constraints are contextually restricted, such that they only regulate consonants that are *pre-vocalic in the input* (cf. Steriade 2001, McCarthy 2011). This prevents the repeated consonants in the base from feature change, because they will necessarily be pre-vocalic in the input. The unrestricted versions rank low (at least below *Repeat(light)), such that reduplicant-final consonants (which are not pre-vocalic in the input, i.e. the output of D1) can change freely to satisfy CodaCond.

(168) /lirooro/ → |lír-li.ròo.ró| → [líl-li.ròo.ró] (feature change across juncture)

|lír-li.ròo.ró|	CodaCond	Id[lat]/_V	*Rpt	*Lapse$_\mu$	Max-μ	Id[lat]
a. lìr.-li.ròo.ró [20-0201]	*!			*		
b. ☞ lìl.-li.ròo.ró [20-0201]				*		*
c. lì.-li.ròo.ró [2-0201]			*!		*	

(169) /rere/ → |rér-re.ré| → [rèr-re.ré] (with medial lapse)

|rér|-|re.ré|	CodaCond	Id[lat]/_V	*Rpt	*Lapse$_\mu$	Max-μ	Id[lat]
a. ☞ rèr.-re.ré [20-01]			*	*		
b. rèr.-re.lé [20-01]		*!		*		*
c. rè.-re.ré [2-01]			**!		*	
d. rè.-le.ré [2-01]		*!			*	*

An alternative, slightly more drastic, solution would be to posit an additional node (M′) *above* the Mother Node where nasal substitution takes place. CodaCond would be inactive in M, such that M selects candidates like (167a) with a surface-impermissible medial cluster. This cluster would become subject to an active CodaCond at M′, where Max can be undominated and thus block all deletion repairs. While we saw above that this kind of vacuous additional node was necessary for Tawala (see also Haugen & Hicks Kennard 2011), it would be nothing more than an *ad hoc* stipulation in this case.

A.5 Local summary

This analysis seems to work, and it uses many of the same constraints involved in the BRCT analysis. One new piece of this analysis is the Max-μ_1 constraint, which is doing a lot of work. We might worry about the typological ramifications of such a constraint.

One potentially objectionable aspect of the analysis is that it posits a stress pattern for D1 that runs completely counter to the rest of the language: D1 creates a (single, left-aligned) "trochee" ([$\acute{\mu}\mu$]) while all the other nodes create (iterative, right-aligned) "iambs" ([$\mu\acute{\mu}$]). D1 requires final consonants (even obstruents) to be moraic (WxP $\gg$ *C$_\mu$#), while the other nodes require final consonants to be non-moraic (*C$_\mu$# $\gg$ WxP). What this is probably reflecting is that Ponapean underwent a historical apocope process that lopped off final unstressed moras. D1 could then be viewed as attesting to the earlier stress pattern, because the reduplicant was protected from apocope because it was non-final.