Reduplication in Ponapean (and Tawala): Re-evaluating "Base-Dependence"

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

AFLA 28
McGill University & National University of Singapore, May 26–28, 2021

1 Introduction

- **Ponapean** (Pohnpeian; Micronesian, Rehg & Sohl 1981) and **Tawala** (Western Oceanic; Ezard 1997) both display intricate phonologically-driven *reduplicant-shape alternations*.[*]
  - On Ponapean, see McCarthy & Prince (1986), Kennedy (2002, 2003), Zukoff (2020), *a.o.*
  - On Tawala, see Hicks Kennard (2004), Inkelas & Zoll (2005:95), Haugen & Hicks Kennard (2011).

- 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, e.g. Base-Reduplicant Correspondence Theory (BRCT; McCarthy & Prince 1995, 1999) are equipped/designed to deal with base-dependent reduplication patterns.

⋆ On the other hand, Morphological Doubling Theory (MDT; Inkelas & Zoll 2005) (claims to) predict the *absence* of base-dependence in reduplication (ibid.:92).

- Haugen & Hicks Kennard (2011) have claimed that Tawala's reduplication-shape alternations must indeed be viewed as base-dependent, and thus resistant to MDT analysis, but compatible with BRCT.
  - In Zukoff (2020), I made the same argument for Ponapean.

→ Today I will show that, while Ponapean's reduplicant-shape alternations *do* constitute base-dependence, they in fact *are* amenable to MDT analysis. (See Appendix A for the same argument in Tawala.)

- By invoking independently argued-for technology relating to prosodic constituency, the BRCT analysis can be imported into MDT in a way that preserves its base-dependent character.

⋆ The take-away is that MDT's claim that it cannot capture base-dependent reduplicant-shape alternations must be abandoned.

→ Therefore, these types of patterns actually will *not* help us distinguish between MDT and other theories of reduplication after all.

[*] Thanks to Jason Haugen, Sharon Inkelas, Nik Rolle, Donca Steriade, Eva Zimmermann, the audiences at AMP 2019, NELS 50, and LSA 2020, and audiences at MIT and Leipzig, for helpful feedback on various portions of this talk. I would especially like to thank Jason Haugen for pointing me towards base-dependence. All mistakes, omissions, and infelicities are my own doing.

Roadmap

Section 2 I propose a slightly revised version of Zukoff's (2020) BRCT analysis of Ponapean.

Section 3 I sketch the architecture of MDT, and show more precisely how Ponapean instantiates base-dependence.

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.

Section 5 I conclude that base-dependence, in terms of reduplicant-shape alternations at least, may not be a suitable grounds for distinguishing between BRCT and MDT.

Appendix A I lay out BRCT and MDT analyses of Tawala (refining Hicks Kennard 2004, Haugen & Hicks Kennard 2011), showing that this "base-dependent" pattern is also amenable to MDT analysis.

Appendix B I lay out an alternative, more traditional MDT analysis of Ponapean.

---

2 Ponapean: a BRCT analysis

- In Ponapean (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):

(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*

∗ The two-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). Since this is a huge topic on its own, I will collapse over these alternations for today's talk.

∗ I also do not consider vowel-initial or syllabic-consonant-initial forms today (consult, e.g., Kennedy 2003:93–100). It is not clear whether these patterns follow completely from any of the analyses I propose below, but I do believe that they are all *consistent* with the constraints I propose.

- I argue that the **length** alternation can be derived through a relatively simple interaction between *stress and phonotactics* (building on Kennedy 2002, 2003), using constraints whose **domain of evaluation spans 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** 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 (based on Zukoff 2020), which is lightly couched in BRCT, but probably compatible with other constraint-based theories that allows for base-dependence.

2.1 A phonological preliminary: Stress and accent in Ponapean

- Rehg (1993) describes the Ponapean stress and accent system as follows:

(4) "High pitch occurs on the penultimate mora, while primary stress is on the final mora; secondary stress occurs on alternate preceding morae" (Rehg 1993:29).

∗ I will assume that we can scale up from his 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 (5), and analyze it with the (foot-free) constraints in (6).

(5) *Stress pattern*
  a. Primary stress on rightmost mora [STRESSR$_\mu$ (6a)]
  b. R→L alternating secondary stress by mora [*CLASH$_\mu$ (6b), *LAPSE$_\mu$ (6c)]

(6) *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. (*$\breve{\mu}\breve{\mu}$)

- Final consonants are non-moraic, while medial coda consonants are moraic (*C$_\mu$# ≫ WEIGHT-BY-POSITION).

⋆ 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:

(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)

→ This difference will be crucial in explaining the distribution of reduplicant length.

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/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 in (8) (chart adapted from Kennedy 2002:225, Kennedy 2003:80):

(8) *Ponapean reduplication: length alternations* (reduplicants bolded)

	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

- In Zukoff (2020), I assumed that the 2-mora reduplicants to even mora count bases had stress on the second mora of the reduplicant, as per Rehg's (1993) stress algorithm.
  - ∗ Today, 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 I propose below.
  - → But note that my BRCT analysis is consistent with second-mora stress; the MDT analyses, on the other hand, would have more trouble with that. This is an empirical question that deserves investigation.

- A clear generalization emerges when looking at the mora count of the base:

(9) a. **Odd** mora count bases always have **2-mora** reduplicants.
b. **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.
  - Even mora count bases have peninitial-mora stress.

→ Therefore, this generalization about mora count can actually be reduced to stress:

(10) 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 (11)), there is a consistent difference that determines which reduplicant length occurs (12).

(11) *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-**keń s**	tò-**toò**.roór lù-**luù**.mʷuúmʷ 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	

(12) a. If it has an initial **light** syllable (i.e. CV̆), it always has a **2-mora** reduplicant.
b. If it has an initial (super)**heavy** syllable [better: *complex rhyme*], it always has a **1-mora** reduplicant.

- 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):

(13) All reduplicants bear a stress on their leftmost mora.

→ This is true of all forms in (8)/(11), 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:

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

2.3.1 Preference for shorter reduplicants (14a)

- I motivate this with the "size restrictor" constraint ALIGN-ROOT-L (15), following Hendricks's (1999) "compression model" of reduplicant shape.[1]

(15) **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)

∗ Note that I define the unit of alignment (cf. Hyde 2012) as the "timing slot" rather than the segment.
  ◦ This is because we will need long vowels, diphthongs, and V̆C sequences to count the same.

- To effectuate the preference for shorter reduplicants, ALIGN-ROOT-L must outrank MAX-BR:

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

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

2.3.2 Reduplicant stress requirement (14b)

- I enforce this via constraint:

(18) **STRESSL-RED:** Assign one violation * for each reduplicant whose leftmost mora is not stressed.

- This is the only context where (moraic) stress lapses can occur (cf. (8)). Therefore, we need this constraint to outrank *LAPSE$_\mu$ (6c):

(19) **Ranking:** STRESSL-RED ≫ *LAPSE$_\mu$

2.3.3 Ban on moraic clash (14c)

- This follows from the same *CLASH$_\mu$ constraint (6b) (repeated here) needed for the basic stress pattern.

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

→ This constraint (+ STRESSL-RED) motivates reduplicant extension for bases with initial stress.

2.3.4 Ban on adjacent identical light syllables (14d)

- This is encoded with the constraint *REPEAT(light) (discussed further in Section 2.4.3):

(21) ***REPEAT(light):** Assign one violation * for each sequence of two adjacent identical light syllables (i.e. *$[C_\alpha \breve{V}_\beta]_\sigma [C_\alpha \breve{V}_\beta]_\sigma$). (cf. Yip 1995, Hicks Kennard 2004)

→ This will motivate reduplicant extension for bases with initial light syllables.

2.3.5 Analysis preview

⋆ These last three constraints all outrank the size restrictor constraint (and *LAPSE$_\mu$):

(22) **Ranking:** STRESSL-RED, *CLASH$_\mu$, *REPEAT(light) ≫ ALIGN-ROOT-L, *LAPSE$_\mu$

Take-away:
→ The combined effect of these constraints can *override* the preference for short reduplicants, yielding otherwise dispreferred 2-mora reduplicants for particular base shapes.

[1] 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.

2.4 Reduplicant shape alternation: Analysis

- This analysis divides up the data into three distinct cases:

(23) 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]

- I will also focus on a sub-type of (23c) where the base itself contains a repetition. [Section 2.4.4]

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 two conditions are met simultaneously:

(24) a. **The base has an even number of moras**, such that the leftmost base stress falls on the base's peninitial mora; and
b. **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 (25):

(25) *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 (26) with a monosyllabic base with a long vowel: [**dù**-duúp].

∗ This tableau shows four candidate outputs that fully cross two variables:

⋆ Most of the tableaux in the rest of this section follow the same format.

	INITIAL STRESS?	
RED. LENGTH	**No**	**Yes**
1-mora	(26a)	(26b)
2-mora	(26c)	(26d)

(26) *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!

- None of the candidates violate *CLASH$_\mu$, because the base-initial mora is unstressed.
- None of the candidates violate *REPEAT(light), because the base-initial syllable is heavy.
- Candidates (26a) and (26c) don't stress the reduplicant-initial mora, and thus fatally violate STRESSL-RED.

→ This leaves (26b) and (26d), pulled out in (27), as the only candidates that satisfy all the top-ranked constraints:

(27) *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 (27d) does worse on both of the lower-ranked constraints the shorter (1-mora) reduplicant in (27b).

- ○ Candidate (27d) violates *LAPSE$_\mu$ because of the sequence of reduplicant-final unstressed mora + base-initial unstressed mora.
- ○ Candidate (27d) incurs an extra violation of ALIGN-ROOT-L because of its extra mora.

→ In this case, both of these constraints help select the 1-mora reduplicant in (27b).

2.4.2 Odd mora count bases → 2-mora reduplicants: STRESSL-RED + *CLASH$_\mu$

- When we move to odd mora count bases (28), the stress considerations flip:
  → The strictly alternating rhythm places a stress on the first mora of the base (*primary* stress in monomoraic bases; *secondary* stress in longer odd mora count bases).

(28) *Odd mora count bases*

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

⋆ Crucially, this now causes the stressed 1-mora reduplicant candidate (29b) 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 (29a), but this would violate STRESSL-RED.

(29) *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

- 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 (29d) 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.

- 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$_\mu$.)

1. 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 (30b) retracts the final stress to the penult, allowing for a stressed 1-mora reduplicant with alternating rhythm.
   - ○ Candidate (30c) eschews base-initial stress, allowing for a stressed 1-mora reduplicant without a clash, but at the expense of a medial lapse.

→ Since the 2-mora reduplicant is optimal, we know that STRESSR$_\mu$ and *LAPSE$_\mu$ outrank ALIGN-ROOT-L.

(30) *Stress is mora 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

2. Extending the reduplicant out beyond 2 moras is never helpful.
   - Candidates (31b) and (31c) show that extending the reduplicant to 3 moras worsens the stress pattern, in addition to increasing violation of ALIGN-ROOT-L.
   - → Candidate (31d) reveals that ALIGN-ROOT-L is actually doing work in the analysis: it prefers the 2-mora reduplicant (31a) to the 4-mora reduplicant (31d), which have equally good stress patterns.

(31) *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!

3. The comparison between winning candidate (31a) and losing candidate (31d) 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 (31a).
   - This would be enough to select (31d), 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.[2]

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 the 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., $[C\breve{V}]_\sigma$), as exemplified in (32).

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

	EVEN	
	2-mora base	**4-mora base**
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

⋆ This effect can be captured with the constraint *REPEAT(light) (repeated in (33)), inspired by Yip's (1995) more general *REPEAT constraint, via Hicks Kennard's (2004) analysis of Tawala (see Appendix A).

(33) ***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 (21) above]

[2] This explains why McCarthy & Prince's (1986) analysis based on "quantitative complementarity" could not account for the entire range of reduplicant-shape alternations.

→ *Repeat(light) will motivate extension to a 2-mora reduplicant in case the base begins in a light syllable:

(34) *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.	duǹ-du.né	[02-01]	*!			3
d. ☞	dùn-du.né	[20-01]			*	3

⋆ Crucially, in this case, *Repeat(light) penalizes the stressed 1-mora reduplicant candidate (34b), 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. (29)).
- ∘ Again, the preferred alternative is the 2-mora reduplicant with initial stress (34d), even though this time it also incurs a *Lapse$_\mu$ violation.

→ 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 *Repeat(light).[3]

- Several additional points can be made with respect to the form [dùn-du.né] and/or *Repeat(light):

1. Stress must not factor into the relevant calculation of "identical" for *Repeat(light), as the two syllables in losing candidate (34b) 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.

2. Align-Root-L helps distinguish between different ways of achieve a 2-mora reduplicant.
   - ∘ Winning candidate (35a) acquires its second mora by copying the nasal consonant from the second onset of the base, and assigning it a mora.
   - ∘ Losing candidate (35b) additionally copies the second base vowel, syllabifying the [n] as a (non-moraic) onset.

→ Since (35a) contains one fewer segment (i.e. timing slot) in the reduplicant than does (35b), Align-Root-L correctly selects it as the winner.

(35) *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!

3. *Repeat(light) is equally applicable in light-syllable-initial *odd* mora count bases, for example:

(36) 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.

(37) *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

[3] The fact that (34d) is preferred to *[dùn-dú.ne] shows that *StressR$_\mu$ ≫ *Lapse$_\mu$.

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.
  - Take, for example, the root in (38), which is composed of two identical light syllables.

(38) 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 (39).
  - Faith-IO rules out various ways of altering the root that could eliminate all consecutive identical light syllables, including deleting the final root vowel (39c) or lengthening the first root vowel (39d).
  - However, since, in BRCT at least, the reduplicant is not subject to IO-faithfulness, it can expand or contract freely in order to avoid a *Repeat(light) violation, thus preferring (39b) over (39a).

(39) *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 (sensitivity only in reduplication) instantiates *the emergence of the unmarked* (TETU; McCarthy & Prince 1994, 1995).

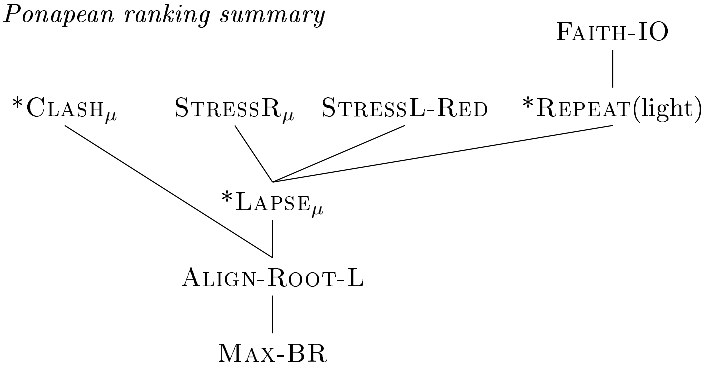
⋆ This is worth dwelling on because Morphological Doubling Theory treats TETU in a very different way.

- 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 (40):

(40) *Ponapean ranking summary*



- This analysis can be understood in a very straightforward way:

(41) a. Reduplicants prefer to be 1-mora long.
b. 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 does it not rule out analyses in other frameworks.

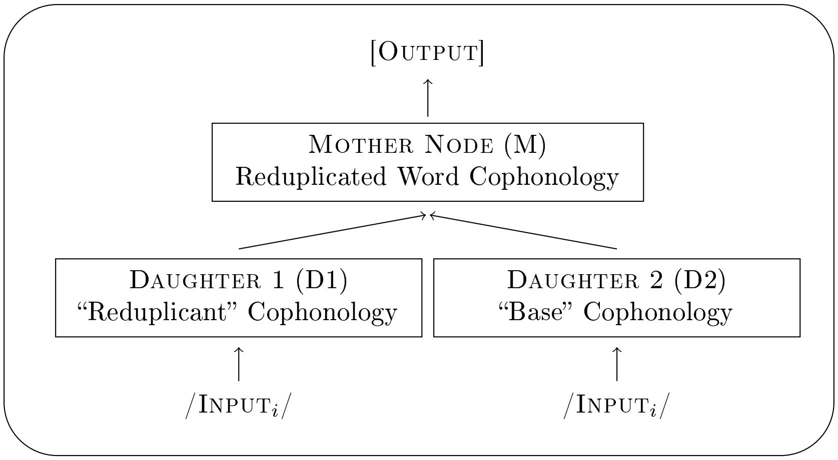
⋆ In the following sections, I will show that **a version of this analysis can in fact be implemented in Morphological Doubling Theory**, despite its apparent "base-dependence".

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 (42):

(42) *Reduplication in MDT*



- The derivation begins with two identical(/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 *cophonologies* (i.e. phonological grammars).

- 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.

3.2 The predicted lack of base-dependence in MDT

⋆ MDT's predicted lack of "base-dependence" follows from this architecture. We can see this when we consider how it allows phonology to apply to its different morphological constituents:

(43) **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.	

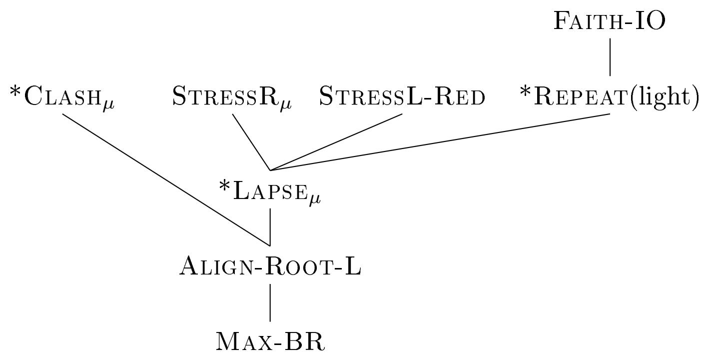
- If we observe a pattern that truly constitutes this last type of interaction, this should be a problem for MDT.

→ 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.

3.3 Base-dependence in the analysis of Ponapean reduplication

- The 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:

(44) *Ponapean BRCT-analysis ranking summary* (repeated from (40) 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:

(45) 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*, 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:

(46) 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 tells us something about the architecture of the reduplicative grammar:

(47) *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), Serial Template Satisfaction in Harmonic Serialism (McCarthy, Kimper, & Mullin 2012).

(48) *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 you that **this is not the case**.

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:** 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 B, 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; cf. Cole 1994, Downing 1998a,b, *a.o.*).

(49) *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 (50):

(50) *Javanese reduplicated causatives* (IZ:139)
   ROOT *uni* 'sound' → CAUSATIVE *ŋ-une-ʔake* → REDUPLICATED *<u>une**ʔ**</u>-une-ʔake* (*<u>une</u>-une-ʔake*)

- Their analysis:

(51) a. The stem-node that 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.
   b. **D1 deletes all non-PRoot segments.**

- While IZ don't spell out exactly how the step in (51b) works, it must be that there are special MAX constraints for segments belonging to PRoots (or any other P-constituents):

(52) **MAX$_{\text{PROOT}}$-IO:** Assign one violation * for each segment contained within a PRoot in the input which lacks an output correspondent.

- In D1, this special MAX constraint must outrank a constraint motivating truncation, e.g. *STRUC (53).

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

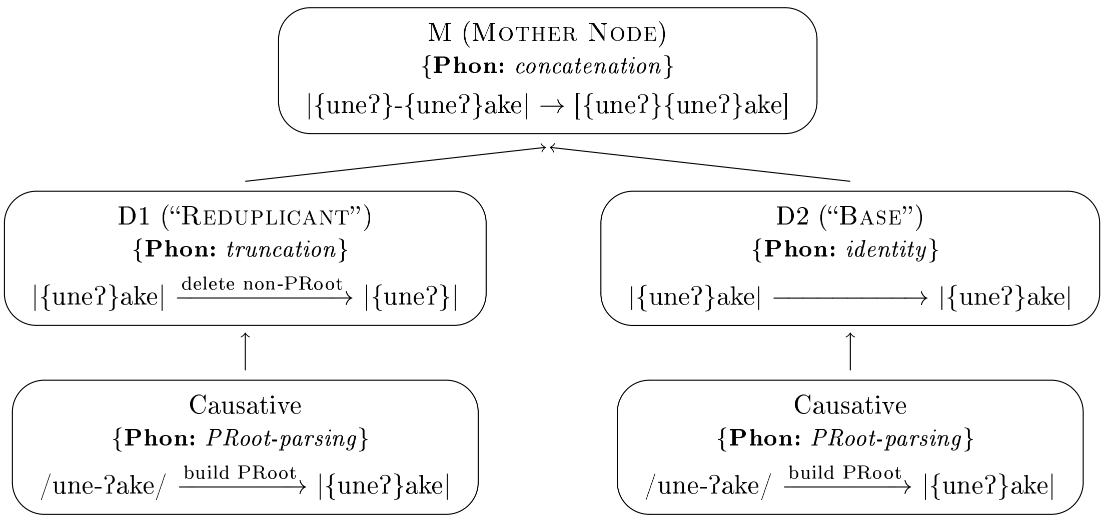
- This constraint in turn outranks the general MAX constraint, motivating deletion of everything outside the PRoot ((54b) ≻ (54a)) but nothing inside the PRoot ((54b) ≻ (54c)).

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

|{uneʔ}ake|		MAX$_{\text{PROOT}}$-IO	*STRUC	MAX-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 realization of the stem, then we derive the desired result:

(55) *Javanese PRoot-based 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:

(56) *Properties of the daughter nodes in the MDT analysis of Ponapean*
  a. D2 builds a PRoot over all of its input material. (It doesn't matter whether it assigns stress.)
  b. 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:

(57) a. Faithfulness to PRoot segments, via MAX$_{\text{PRoot}}$-IO
  b. Faithfulness to stressed segments, via MAX$_{\acute{\mu}}$-IO (58)

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

- While we could use *STRUC to trigger deletion like in Javanese, we could also use an alignment constraint:

(59) **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.

→ This has the exact same "size-restrictor" effect as ALIGN-ROOT-L did in the BRCT analysis:

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

|láud-{laud}|			MAX$_{\text{PRT}}$	MAX$_{\acute{\mu}}$	*CLASH$_{\mu}$	*LAPSE$_{\mu}$	ALN-PRT-L	MAX
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

∗ Demotion of D1's primary stress to a second stress can be accounted for by requiring the rightmost stress to be primary, and having the constraint enforcing this outrank IDENT[stress degree].

- Deleting segments from the PRoot (60d,e,g) isn't allowed because it violates undominated MAX$_{\text{PRoot}}$-IO.
  ∗ Deleting PRoot segments confers no benefit with the constraints used here, but it would increase satisfaction of *STRUC.
- Deleting stressed "reduplicant" segments (60e,f,g) isn't allowed because it violates undominated MAX$_{\acute{\mu}}$-IO.
  ∗ Because all PRoot segments are protected by MAX$_{\text{PRoot}}$-IO, it doesn't matter whether the output of D2 contains stress or not, and thus whether deleting them ever violates MAX$_{\acute{\mu}}$-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 (60a,b) results in increased violation of ALIGN-PROOT-L (and, in this case, *LAPSE$_{\mu}$), and therefore the candidate that deletes these segments (60c) 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$_{\mu}$ (61) or *REPEAT(light) (62).

- 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 (61c).
  → As long as *CLASH$_{\mu}$ $\gg$ ALIGN-PROOT-L, it will be preferable *not* to delete the next mora (61b) so as to avoid the clash.

(61) *Odd mora count bases block reduction*

|dúupek-{duupek}|			Max$_{\acute{\mu}}$	*Clash$_\mu$	*Lapse$_\mu$	Aln-PRt-L	Max
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 in there being identical light syllables across the juncture (62c).
  → As long as *Repeat(light) ≫ Align-PRoot-L (and *Lapse$_\mu$), it will be preferable *not* to delete the next mora so as to avoid the light repetition.

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

|dúne-{dune}|			Max$_{\acute{\mu}}$	*Repeat(light)	*Lapse$_\mu$	Aln-PRt-L	Max
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 (63) and that rightmost stress is more important than avoiding a lapse (64).

(63) *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

(64) *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$_{\text{PRoot}}$ generates TETU

- We've just seen that Max$_{\text{PRoot}}$-IO prevents deletion of "base" material in the general case.

⋆ It will also block deletion that could repair base-internal and/or base-reduplicant-junctural *Repeat(light) violations, given the ranking Max$_{\text{PRoot}}$-IO ≫ *Repeat(light):

(65) Max$_{\text{PRoot}}$-IO *prevents base reduction for* *Repeat(light)

|rére-{rere}|			Max$_{\text{PRt}}$	*Repeat(light)	*Lapse$_\mu$	Aln-PRt-L	Max
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$_{\text{PRoot}}$ blocks deletion in the base that could have avoided a *Repeat(light) violation (65d), just like Max-IO did in the BRCT analysis.

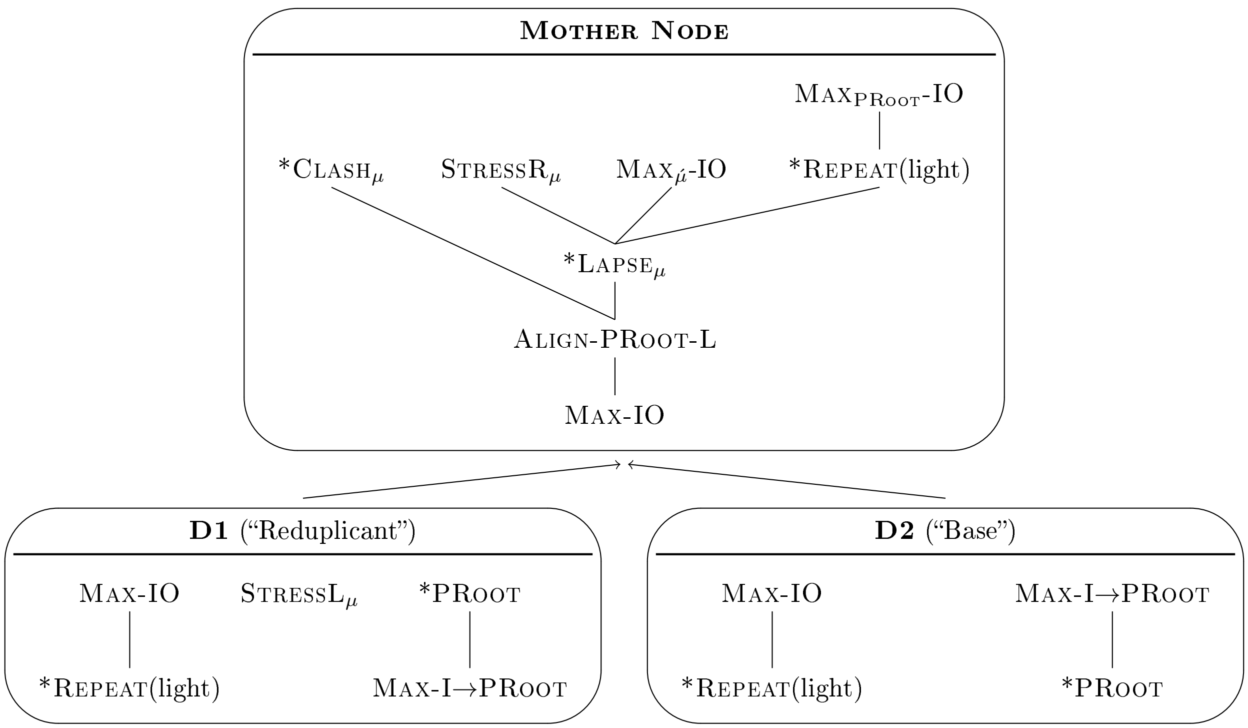
- To prevent other repairs in the base which may be tolerated in the reduplicant (i.e. feature change, lengthening, or epenthesis), we can index other faithfulness constraints to the PRoot as well.

> → 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

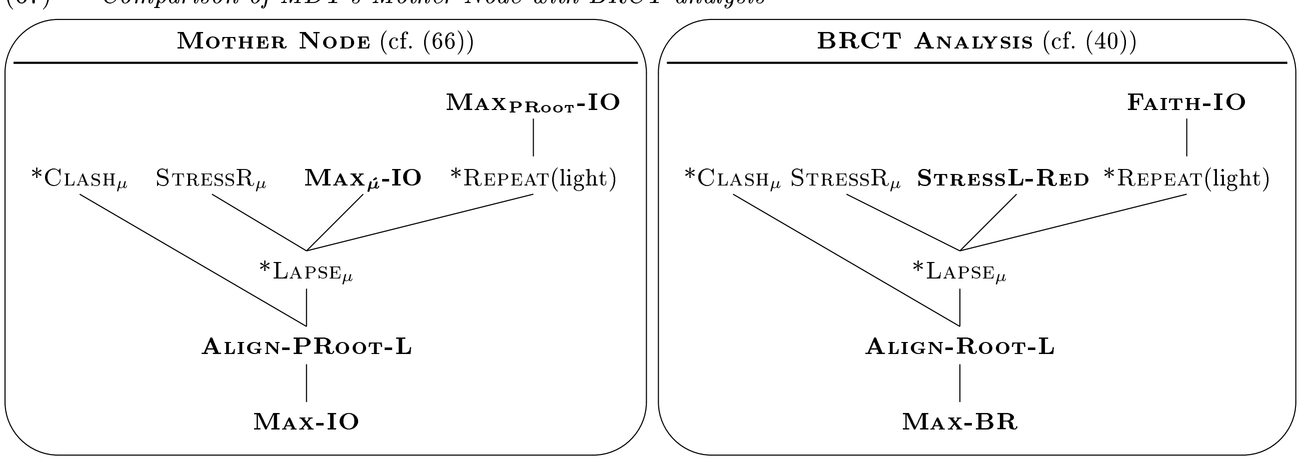
- The rankings motivated by the MDT Mother Node analysis (coupled with a sketch of the constraints needed for the daughter nodes) are summarized in (66):

(66) *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:

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



5 Discussion

5.1 Summary

- This demonstrates that MDT can actually directly recapitulate the BRCT analysis of Ponapean reduplication, by placing most of the action in the Mother Node.
- This analysis only uses technologies that Inkelas & Zoll (2005) use themselves, most notably morphologically-related prosodic constituents, and faithfulness to that constituency.

→ Since this analysis takes base-dependence at face value, i.e. the ultimate shape of the reduplicant still requires knowledge of the concatenated output string, Inkelas & Zoll's (2005) claim that MDT cannot generate base-dependent reduplicant-shape alternations cannot be upheld.
  - ⋆ This argument holds regardless of whether this is the *correct* analysis of Ponapean, because the tools employed are all appropriate to MDT, and thus we predict languages that do work this way.
  - ∘ See Appendix B for an alternative, more traditional MDT analysis of the Ponapean facts which distributes reduplicant deletion across D1 and M.

∗ Similar arguments can be made from Tawala, as laid out in Appendix A.

- Furthermore, it shows that TETU effects can in fact by analyzed *as such* in MDT, where *prosodic* asymmetries stand in for *morphological* asymmetries in other theories.

5.2 Conclusion

- This leaves us with three options for how to think about base-dependent reduplicant-shape alternations and the status of MDT:

(68) *The range of logical conclusions*

a. Ponapean *counts* as a base-dependent reduplicant-shape alternation, and MDT can account for it.
↪ MDT must abandon its claimed prediction of base-*in*dependence (for reduplicant-shape alternations, at least)

b. Ponapean *doesn't count* as a base-dependent reduplicant-shape alternation, *because* MDT can account for it.
↪ MDT's prediction of base-*in*dependence for reduplicant-shape alternations is vacuous/ tautological.

c. We reject the MDT analysis based on the way it uses prosodic constituents, and reserve judgment about whether Ponapean constitutes base-dependence.
↪ MDT must abandon a number of its central arguments against BRCT, which are based on analyses that use prosodic constituents.

⋆ I see no reason not to accept (68a), and embrace the idea that base-dependent reduplicant-shape alternations ultimately are *not* a fruitful ground for distinguishing between MDT and other theories of reduplication.

5.3 Outlook

- Note, however, that this kind of reduplicant-shape alternation is not the only claimed type of base-dependence:

(69) *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.

(70) *Examples of prosodic constituent copying*

a. Foot copying in Yidin^y (Haugen & Hicks Kennard 2011:7; McCarthy & Prince 1986)
b. Syllable copying in Hiaki (Haugen & Hicks Kennard 2011:9; Haugen 2003)

- Haugen & Hicks Kennard (2011) assert 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 analysis of Ponapean (and also Javanese), I don't think that this is a correct interpretation. Such cases actually have at least two possible analyses:

(71) *Possible MDT analyses of prosodic constituent copying*

a. The reduplicative daughter node (D1) is fed by a constituent which builds syllable/foot structure, and D1 deletes everything not contained within that syllable/foot.
b. The syllable/foot-sized constituent is actually a prosodic root/stem (constrained to be of the relevant size), and all additional material transmitted from D1 is deleted in the mother node.

- On the other hand, the question of whether base-dependence exists in reduplication-phonology interactions (Wilbur 1973) — things like over-/under-application opacity, back-copying, re-copying, etc. — is very much a live issue in the literature (e.g., McCarthy & Prince 1995, Inkelas & Zoll 2005, Kiparsky 2010, McCarthy, Kimper, & Mullin 2012).
  → Many of these debates actually hinge on *empirical questions* rather than theoretical ones, and thus may have a better chance of getting resolved and yielding results.

Ultimate take-away:
→ Reduplicant-shape alternations do not substantially differentiate MDT from BRCT, so we should keep focusing on reduplication-phonology interactions.

References

Applegate, Richard B. 1972. Ineseño Chumash Grammar. PhD Dissertation, University of California, Berkeley. https://escholarship.org/content/qt2hp7h2xv/qt2hp7h2xv.pdf.

———. 1976. Reduplication in Chumash. In Margaret Langdon & Shirley Silver (eds.), *Hokan Studies*, 271–284. The Hague: Mouton. https://doi.org/10.1515/9783110819113-017.

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.

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.

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

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

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

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. https://openresearch-repository.anu.edu.au/bitstream/1885/145419/1/PL-B97.pdf.

Hyde, Brett. 2012. Alignment Constraints. *Natural Language & Linguistic Theory* 30(3):789–836.

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. http://www.linguistics.ucsb.edu/faculty/rkennedy/papers/kennedy.wccfl.21.pdf.

———. 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. http://www.stanford.edu/~kiparsky/Papers/reduplication.pdf.

Kurisu, Kazutaka. 2013. Nested Derivedness in Ponapean Morphophonology. *Lingua* 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). http://scholarworks.umass.edu/linguist_faculty_pubs/13.

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

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

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

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

———. 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. http://works.bepress.com/john_j_mccarthy/77.

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.

Schachter, Paul & Fe T. Otanes. 1972. *Tagalog Reference Grammar*. Berkeley: University of California 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.

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. http://arizona.openrepository.com/arizona/handle/10150/227280.

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.

———. 2020. 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.

A Appendix A: Tawala reduplication

A.1 Tawala: a BRCT analysis

- Hicks Kennard (2004) [henceforth HK] develops an atemplatic analysis of reduplication in Tawala (Papua New Guinea, Western Oceanic, Oceanic; Ezard 1997).
- 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 revision of the analysis based on 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.

A.1.1 The data

- The Tawala durative exhibits four distinct reduplicant shapes, whose distribution is phonologically predictable (see HK, based on Ezard 1980, 1997):

(72) *Reduplicant-shape alternations in Tawala*

Type A: $C_1V_1.V_2$-initial bases reduplicate as 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 vowel

∗ 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].

- These patterns are schematized and exemplified in (73):

(73) *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* → *bi-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* → *hu.ne-hu.ne.ya*	'praise/be praising'	[HK:307]
C.	V_1C_1X	→ $V_1.C_1$-	e.g. *a.tu.na* → *a.t-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.gu.ya*	'preach/be preaching'	[HK:305]

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

A.1.2 Consonant-initial roots: Type A & Type B reduplication

HK's analysis

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

- The first constraint is *REPEAT (74), which bans adjacent identical syllables.
  → I am going to use a more specific version of this constraint, *REPEAT(initial) (75), for reasons which will become clear below.

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

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

- The other constraint is ALIGN-ROOT-L (76), which prefers minimizing the length of the reduplicant.

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

> ∗ 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 (77), the constraint requiring contiguous copying, in order to generate Type A's discontiguous copying (78d).

(77) **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 $r_1 ... r_n$ standing in correspondence with a base string $b_1 ... b_n$, assign one violation * for each segment between b_1 and b_n which lacks a correspondent in $r_1 ... r_n$.

(78) *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 (79d), which was not considered by HK (nor by Haugen & Hicks Kennard 2011).

(79) *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	*(*)

> ∗ HK includes a constraint requiring a foot at the left edge of the word, and implies that this has something to do with generating disyllabic copying in Type B, but I don't think this actually follows from her analysis.
> • There does need to be constraint requiring the left-edge of prefixes to bear a stress (HK:306–307) (≈ STRESSL-RED for Ponapean), but this does not explain the copying facts.

My proposed fix

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

(80) a. Type A skips only vowels (base V_1): *bi-b[e].i.ha*
b. For Type B, the problematic discontiguous candidate (79d) 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 (81) and vowels (82):

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

(82) **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 (83), we derive the right results (84, 85).

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

(84) *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	*

(85) *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	*

A.1.3 Vowel-initial roots

- In order to account for VC-copying to vowel-initial roots, the only additional piece we need is a requirement to copy from the left edge of the base (Anchor-L-BR (86)) which outranks Align-Root-L.

(86) **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)

(87) *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 (88) turns out to be unnecessary, even though we might have expected it to be responsible for eliminating (87a) and (87c), as it is in HK's analysis.

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

- The same ranking explains why mis-anchoring is not tolerated in Type B to achieve a CV reduplicant:

(89) *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

A.1.4 Type D reduplication, *REPEAT, and TETU

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

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

- HK (followed by Haugen & Hicks Kennard 2011 [henceforth HHK]) also uses it to help analyze Type D:

(91) *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'

*REPEAT outside of reduplication

- However, *REPEAT (both the specific and the more general version) is freely violated outside of reduplication (HHK:24–26), including within roots, across compound boundaries, and at other base-affix junctures.
  - This is illustrated for the root /totogo/ → [to.to.go] 'be sick' (Ezard 1997:33, HK:305) in (92):

(92) *REPEAT(init) *violations permitted outside of reduplication*

/RED, totogo/	MAX-IO	DEP-IO	IDENT-IO	*REPEAT(init)
a. ☞ to.to.go				*
b. to.t**i**.go			*!	
c. to.*pa*.to.go		*!*		
d. to.go	*!*			

⋆ This means that the avoidance of repeated identical (initial) syllables in reduplication in Tawala is again an instance of TETU, as argued by HK & HHK.

Type D reduplication

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

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

/RED, $g_1u_2g_3u_4y_5a_6$/	FAITH-IO	*REPEAT (init)	ANCH-L-BR	ALN-RED-L	ALN-RT-L	*REPEAT
a. $\underline{g_1u_2.g_3i_4}.\text{-}g_1u_2.g_3i_4.y_5a_6$	*!				4	
b. $\underline{g_1u_2.g_3u_4}.\text{-}g_1u_2.g_3u_4.y_5a_6$		*!			4	***
c. $\underline{g_1u_2}.\text{-}g_1u_2.g_3u_4.y_5a_6$		*!			2	**
d. $\underline{g_1u_4}.\text{-}g_1u_2.g_3u_4.y_5a_6$		*!			2	**
e. $\underline{y_5a_6}.\text{-}g_1u_2.g_3u_4.y_5a_6$			*!		2	*
f. ☞ $g_1\text{-}\underline{u_2}\text{-}.u_2.g_3u_4.y_5a_6$				1		
g. $g_1\text{-}\underline{u_4.y_5}\text{-}u_2.g_3u_4.y_5a_6$			*!	1		

- The crucial rankings that makes this interaction possible are:

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

- ∗ In order to prefer desired candidate (93f) over candidate (93g), 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 (93g), by lower-ranked ONSET and/or MAX-BR.

Why not general *REPEAT?

- 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)

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

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

- There is at least one relevant base which can disambiguate between these two predictions:

(96) *kilolo* 'urinate' → *$\underline{kilo}$-kilolo* 'urinating' (**kil-$\underline{o}$-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.

(97) *Type B reduplication for* /kilolo/ *with* *REPEAT(init)

/RED, $k_1i_2l_3o_4l_5o_6$/	*REPEAT-(init)	CONTIGC-BR	ALIGN-RED-L	ALIGN-ROOT-L	*REPEAT
a. ☞ $\underline{k_1i_2.l_3o_4}.\text{-}k_1i_2.l_3o_4.l_5o_6$				4	*
b. $\underline{k_1i_2}.\text{-}k_1i_2.l_3o_4.l_5o_6$	*!			2	**
c. $\underline{k_1o_4}.\text{-}k_1i_2.l_3o_4.l_5o_6$		*!		2	*
d. $k_1i_2.l_3\text{-}\underline{o_4}.\text{-}o_4.l_5o_6$			3!		

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

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

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

- In (99), we see that general *REPEAT must be ranked *below* ALIGN-ROOT-L, or else candidate (99b), which additionally copies V_2 to avoid the medial repetition, would be preferred to desired candidate (99a).

(99) *Type C reduplication for* /otowi/ *with* *REPEAT(init)

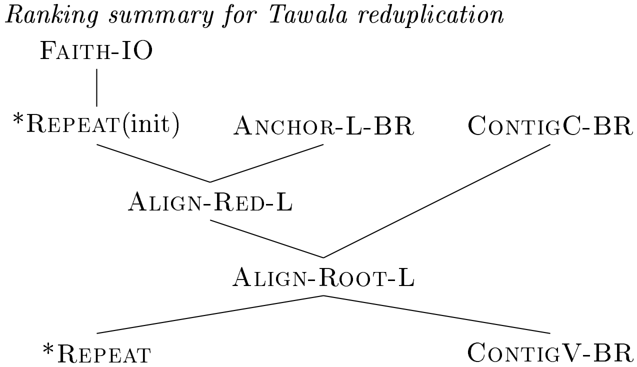
/RED, otowi/	*REPT(init)	ANCH-L-BR	ALIGN-RED-L	ALIGN-ROOT-L	*REPT
a. o.to.-o.to.wi				3!	
b. ☞ o.t-o.to.wi				2	*
c. o.-o.to.wi	*!			1	*
d. t-o.to.wi	*!	*!		1	*
e. w-o.to.wi		*!		1	
f. o.t-o-o.wi			2!		

⋆ 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.

A.1.5 Local summary

(100) *Ranking summary for Tawala reduplication*



A.2 Tawala: an MDT analysis

- Because the BRCT analysis of Tawala leans heavily on BR-faithfulness constraints, there's no easy way to import it into MDT like we did for Ponapean.
- Nevertheless, using the PRoot technology and distributing deletion across the different nodes, we can establish a consistent analysis.

A.2.1 Truncation to two syllables in D1

The general case: Types A, B, and C

- Inkelas & Zoll (2005:95) [henceforth IZ] assert (in line with HK's analysis) that the target reduplicant shape is a single foot (i.e. two syllables), and that this is the output of the reduplicative daughter cophonology (i.e. D1 induces truncation). (They don't consider Type D.)

(101) **Outputs of D1: truncation (left-anchored, two syllables)**

	INPUT			OUTPUT OF D1	
Type A:	*beiha*	(/$C_1V_1.V_2X$/)	→	*be.i*	(|$C_1V_1.V_2$|)
Type B:	*huneya*	(/$C_1V_1.C_2V_2X$/)	→	*hu.ne*	(|$C_1V_1.C_2V_2$|)
Type C:	*atuna*	(/$V_1C_1V_2X$/)	→	*a.tu*	(|$V_1.C_1V_2$|)

- I will derive the two-syllable shape using foot-free stress constraints:

(102) *Stress constraints for D1*

a. **STRESSL:** Assign one violation * if the initial mora is unstressed. (*#σ̆)

b. **NONFINALITY:** Assign one violation * if the final mora is stressed. (*σ́#)

∗ I use the same constraints for the same purpose in the alternative MDT analysis of Ponapean in Appendix B.

- These constraints outrank *STRUC, which in turn outranks MAX:

(103) *Truncation to two syllables in D1* (Type B)

/huneya/				STRESSL	NONFIN	*STRUC	MAX
a.		hú.ne.ya	[100]			6!	
b.	☞	hú.ne	[10]			4	2
c.		hú	[1]		*!	2	4
d.		hu	[0]	*!		2	4

- The two-syllable string is necessarily left-anchored and contiguous, even if that leads to hiatus:

(104) *Selecting the correct two syllables* (Type A)

/beiha/				ANCHOR-L	CONTIG	*VV
a.	☞	bé.i	[10]			*
b.		bé.ha	[10]		*!	
c.		bí.ha	[10]		*!	
d.		í.ha	[10]	*!		

  ◦ STRESSL and NONFIN must dominate *VV, so that |bé.i| ≻ *|bé| and *|be|.

***REPEAT(init) and Type D**

- The key to getting Type D to work out is generating further truncation in D1.
  ⋆ The motivation is clear: *REPEAT(init).

→ As long as ANCHOR-L, CONTIG, STRESSL, and *REPEAT(init) all dominate NONFIN, we derive truncation down to the initial syllable (with stress):

(105) *Truncation to **one** syllable in D1 motivated by* *REPEAT(init) (Type D)

/gu_1gu_2ya/				ANCHOR-L	CONTIG	STRESSL	*REPEAT(init)	NONFIN
a.		$gú_1.gu_2.ya$	[100]				*!	
b.		$gú_1.gu_2$	[10]				*!	
c.		$gú_1.u_2$	[10]		*!			
d.	☞	$gú_1$	[1]					*
e.		gu_1	[0]			*!		
f.		$gu_1.ya$	[10]		*!			
g.		$gu_2.ya$	[10]	*!				

- We now (nearly) have the truncation outputs we'll need in order to derive the right patterns in M:

(106) **Outputs of D1: truncation (one or two syllables)**

	Input		Output of D1
Type A:	*beiha* (/$C_1V_1.V_2X$/)	→	*bé.i* (|$C_1\acute{V}_1.V_2$|)
Type B:	*huneya* (/$C_1V_1.C_2V_2X$/)	→	*hú.ne* (|$C_1\acute{V}_1.C_2V_2$|)
Type C:	*atuna* (/$V_1C_1V_2X$/)	→	*á.tu* (|$\acute{V}_1.C_1V_2$|)
Type D:	*guguya* (/$C_1V_1C_1V_1X$/)	→	*gú* (|$C_1\acute{V}_1$|)

→ There is a problem with Type A that I will return to below.

A.2.2 PRoots, stress, and *Repeat(init) in D2

- D2 must have three properties:

(107) a. All of its output segments are contained within a PRoot (or PStem).
b. It applies the language's regular stress pattern.
c. It deletes its initial consonant if that can repair a *Repeat(init) violation.

- The PRoot will help us generate the asymmetry between deletion in the reduplicant and non-deletion in the base, in the same way it did in Ponapean.
- We need to assign regular stress (≈ right-to-left trochees) in this node, because reduplicated forms exceptionally allow clashes at the reduplicant-base juncture.

∗ I omit the constraints that derive these facts because they don't interact with anything else.

- The *Repeat(init)-driven deletion will be enough to get us to the unusual "V-doubling" pattern in Type D:
  - As long as *Repeat(init) and Contig outrank Anchor-L and Max, this will trigger minimal contiguous deletion (i.e. the first C) from the left edge in order to repair an initial repetition.

(108) *Repeat(init)-*driven deletion in D2* (Type D)

/gu_1gu_2ya/	Contig	*Repeat(init)	Anchor-L	Max	*Struc
a. {$gu_1.g\acute{u}_2.ya$}		*!			6
b. ☞ {$u_1.g\acute{u}_2.ya$}			*	*	5
c. {$g\acute{u}_2.ya$}			*	**!	4
d. {$g\acute{u}_1.ya$}	*!				4

- With this in hand, we have the following outputs of D1 and D2, which form the input to M:

(109) **Inputs to M**

	Input to M
Type A:	|bé.i|-|{be.í.ha}|
Type B:	|hú.ne|-|{hu.né.ya}|
Type C:	|á.tu|-|{a.tú.na}|
Type D:	|gú|-|{u.gú.ya}|

A.2.3 Vowel elision outside of the PRoot in the Mother Node

- The only process which needs to take place in the Mother Node is deletion to repair hiatus.
- This deletion process is constrained by the same two things as it was in Ponapean:

(110) a. Input affiliation with a PRoot
b. Input specification for stress

Type D

- First consider Type D. In this case, there is hiatus across the juncture, but it cannot be resolved because V_1 is stressed and V_2 is part of a PRoot:

(111) *Vowel deletion in M blocked by faithfulness for Type D*

|gú|-|{u.gú.ya}|	Max$_{\acute{V}}$	Max$_{PRt}$	*VV	Max
a. ☞ gù.{u.gú.ya}			*	
b. gù.{gú.ya}		*!		*
c. g{u.gú.ya}	*!			*

Type C

- In Type C, the reduplicant-final vowel is unstressed, so it can delete in response to *VV:

(112) *Reduplicant-final vowel deletion in M for Type C*

|á.tu|-|{a.tú.na}|	Max$_{\acute{V}}$	Max$_{PRt}$	*VV	Max
a. à.tu.{a.tú.na}			*!	
b. à.tu.{tú.na}		*!		*
c. ☞ à.t{a.tú.na}				*

- This accounts equally for [o.to.to.wi], where the D1-final and D2-initial vowels are identical. It is distinguished from the other case of identical vowels (Type D) by the absence of stress.

(113) *Reduplicant-final vowel deletion in M for Type C:* [o.to.to.wi]

|ó.to|-|{o.tó.wi}|	Max$_{\acute{V}}$	Max$_{PRt}$	*VV	Max
a. ò.to.{o.tó.wi}			*!	
b. ò.to.{tó.wi}		*!		*
c. ☞ ò.t{o.tó.wi}				*

Type B

- Since deletion in M is only triggered by *VV, and not a general truncation-inducing/size-restrictor constraint, when hiatus is not at stake, e.g. Type B, no deletion is motivated:

(114) *Faithfulness in Type B*

|hú.ne|-|{hu.né.ya}|	Max$_{\acute{V}}$	Max$_{PRt}$	*VV	Max
a. ☞ hù.ne.{hu.né.ya}				
b. hù.ne.{né.ya}		*!*		**
c. hù.{hu.né.ya}				*!*

The problem with Type A

- We encounter a problem here with Type A.
  - What we want is to delete D1's stressed V_1 (\|é\|) and keep D1's unstressed V_2 (\|i\|).
  - But we currently predict the reverse:

(115) *Incorrectly predicted V_2-deletion for Type A*

|bé.i|-|{be.í.ha}|	Max$_{\acute{V}}$	Max$_{PRt}$	*VV	Max
a. bè.i.{be.í.ha}			**!	
b. bè.i.{bí.ha}		*!	*	*
c. bi.{bí.ha}		*!		**
d. 💣 bè.{be.í.ha}			*	*
e. ☹ bi.{be.í.ha}	*!		*	*

- We can notice that the problematic candidate (115d) has an initial repetition. However, ranking *Repeat(init) above $\text{Max}_{\acute{V}}$ results not in our desired candidate (116e), but in the non-deletion candidate (116a).

(116) *Adding* *Repeat(init) *incorrectly predicts no deletion for Type A*

|bé.i|-|{be.í.ha}|	Max_{PRt}	*Repeat(init)	$\text{Max}_{\acute{V}}$	*VV	Max
a. 💣 bè.i.{be.í.ha}				**	
b. bè.i.{bí.ha}	*!			*	*
c. bi.{bí.ha}	*!	*			**
d. bè.{be.í.ha}		*!		*	*
e. ☹ bi.{be.í.ha}			*!	*	*

A.2.4 A brute-force fix: adding a node

- Following a suggestion by Haugen & Hicks Kennard (2011:15–16), we can solve this problem by introducing an additional (semantically vacuous) node (D1′) between D1 and M.
  → In D1′, we can enforce the change we need: |bé.i| → |bí|

(117) *Vowel deletion in D1′ for Type A*

|bé.i|	*VV	Anchor-R	StressL	Max	Contig
a. bé.i	*!				
b. bé		*!		*	
c. bi			*!	*	*
d. ☞ bí				*	*

- This does not effect any of the other types, because none of them have hiatus in the output of D1.

- If |bí| is what is transmitted to M, then we avoid the problem with Type A:

(118) *Faithful realization of D1′ in M for Type A*

|bí|-|{be.í.ha}|	$\text{Max}_{\acute{V}}$	Max_{PRt}	*VV	Max
a. ☞ bì.{be.í.ha}			*	
b. bì.{bí.ha}		*!		*

- This solution makes the analysis more complex, opaque, and *ad hoc*, but it seems to work.

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

- The logic behind this alternative MDT analysis of Ponapean is as follows:

(119) a. D1 truncates the input down to the leftmost two moras, placing stress on the initial mora.
b. D2 applies the regular stress pattern, as described/analyzed in Section 2.1.
c. 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.

- The data is repeated in (120):

(120) *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

B.1 Truncation in D1

- Under this analysis, the target shape for the reduplicant, i.e. the output of D1, is 2 moras.
- Because *LAPSE$_\mu$ will be implicated in further truncating down to 1 mora in the special case (which was the default case in the BRCT analysis), it is crucial that D1 place stress on the first mora.
- Usefully, stress constraints can do the job:

(121) *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. (*$\acute{\mu}$#)

- The stress constraints in (121) will be undominated in D1. They crucially dominate

(122) *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 w/o a correspondent in the output.
c. **Ranking:** *$\mu \gg$ MAX

- If the stress constraints in (121) outrank *μ, then each input will be truncated down to exactly two moras:
  - One to satisfy STRESSL$_\mu$, one to satisfy NONFIN$_\mu$.

(123) /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):

(124) *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.

(125) **Ranking:** CONTIG, ANCHOR-L, *STRUC[X] ≫ ANCHOR-R

(126) /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:

(127) *Consonant moraicity constraints in D1*

a. **WEIGHT-BY-POSITION [WBP]:** 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.
d. **Ranking:** (WBP,) *STRUC[X] ≫ *C$_\mu$#, *C$_\mu$

(128) /dune/ → |dún|

/dune/			WBP	*STRUC[X]	*C$_\mu$#	*C$_\mu$	MAX
a.	dú.ne	[10]		****!	*		
b. ☞	dún$_\mu$	[10]		***	*	*	*

- While the D1 outputs while sometimes surface faithfully in the ultimate reduplicated word output, there also sometimes it won't. Nevertheless, the D1 output always follows this grammar. For example:

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

/laud/			NONFIN$_\mu$	*STRUC[X]	MAX
a.	láud$_\mu$	[100]		****!	
b.	láud	[10]		****!	
c. ☞	láu	[10]		***	*
d.	lá	[1]	*!	**	**

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

/duup/			NONFIN$_\mu$	*STRUC[X]	MAX
a.	dúup$_\mu$	[100]		****!	
b.	dúup	[10]		****!	
c. ☞	dúu	[10]		***	*
d.	dú	[1]	*!	**	**

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

- M has essentially the same stress grammar as D2, i.e. the stress grammar from the BRCT analysis.
  - It also converts non-final "underlying" primary stresses to second stresses; this doesn't violate the IDENT[stress] constraint I use below.
- What is interesting about stress in M is that *LAPSE$_\mu$ dominates MAX.
- 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.

- Getting deletion to work right is (relatively) unproblematic. CONTIG will prefer deleting the "reduplicant"-final moraic segment (131c) over a "base"-internal moraic segment (131d).
  - This requires assuming that adjacency/precedence relations are not present in M-input for segments belonging to different daughter outputs (a point also made by HHK w.r.t. Tawala).

(131) /laud/ → |láu-laúd| → [là-laúd]

|láu|-|laúp|			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 (132).

(132) /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. Let's assume the following constraints

(133) *Moraic faithfulness constraints*

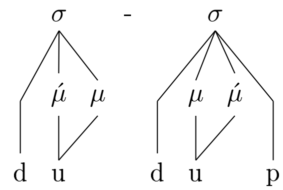
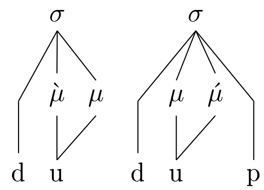
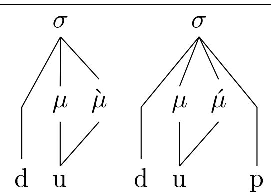
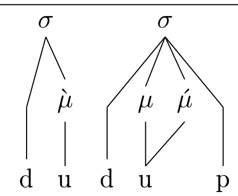
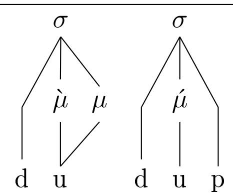
a. **MAX-μ_1:** Assign one violation * for deleting the leftmost underlying mora of a syllable.

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

- If we assume the ranking in (134), we can generate the desired outcomes, as shown in (135) below.

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

(135) /duup/ → |dúu-duúp| → [dù-duúp]

		Ident[stress]	*Lapse$_\mu$	Max-μ	Max-μ_1
					
a. 	[20-01]		*!		
b. 	[02-01]	*!*			
c. ☞ 	[2-01]			*	
d. 	[20-1]			*	*!

B.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:

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

|dún|-|du.né|		Ident[stress]	*Lapse$_\mu$	Max-μ	Max-μ_1
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 (136c).
  - We also need Max-μ_1 to outrank *Lapse$_\mu$ (which is consistent with (135)), so that (136a) ≻ (136d).

(137) /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 (138b) 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 (138d) by having Dep-μ outrank *Lapse$_\mu$.

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

|dún|-|du.né|		Dep-μ	*Repeat(light)	WBP	*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]	*!				

B.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 (138)), and also Onset outranks *Repeat(light), all possible repairs are ruled out:

(139) /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. reȑ.-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:

(140) /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 repair 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 languages version of CodaCondition.
  - Deleting to avoid the CodaCond violation is ruled out because it would create a clash (violating undominated *Clash$_\mu$).

(141) /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 (142c).

(142) /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.

(143) /lirooro/ → |lír-li.ròo.ró| → [lí**l**-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]			*!		*	

(144) /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.**l**é	[20-01]		*!		*		*
c. rè.-re.ré	[2-01]			**!		*	
d. rè.-**l**e.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 (142a) 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 these sorts of vacuous additional nodes are sometimes necessary in MDT (see Haugen & Hicks Kennard 2011 on Tawala), it would be nothing more than an *ad hoc* stipulation in this case.

B.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 (WBP ≫ *C_μ#), while the other nodes require final consonants to be non-moraic (*C_μ# ≫ WBP).

→ Though 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.