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

Sam Zukoff, UCLA

samzukoff@gmail.com · www.samzukoff.com

July 7, 2023

Abstract

Most theories of reduplication, including Base-Reduplicant Correspondence Theory (McCarthy & Prince 1995), predict the existence of "base-dependent" reduplication-phonology interactions (Inkelas & Zoll 2005; Haugen & Hicks Kennard 2011), wherein phonological properties of the reduplicant or base crucially depend on information present only in the other constituent. Inkelas & Zoll assert that Morphological Doubling Theory (MDT) predicts the absence of such patterns, and indeed that, when borderline cases are properly analyzed, no such cases exist.

This paper seeks to clarify the state of affairs regarding base-dependent *reduplicant shape alternations* via the analysis of partial reduplication in Tawala (Ezard 1997) — based on refinements and expansions of the analyses presented in Hicks Kennard (2004) and Haugen & Hicks Kennard (2011) — which has been identified as a potential case of base-dependence. This paper advances two conclusions. First, Tawala's reduplication pattern *does* constitute base-dependence, countering Inkelas & Zoll's claim to the contrary and undermining a prediction of MDT. Second, *contra* Haugen & Hicks Kennard, MDT *is* capable of analyzing this pattern; however, doing so requires a highly *ad hoc* analysis involving powerful machinery, raising further concerns about the restrictiveness of MDT.

1 Introduction

Theories of reduplication, just as theories of phonology and morphophonology generally, differ on the extent to which they rely on parallel (all at once) derivation vs. serial or cyclic (step-by-step) derivation. The dominant parallelist theory of reduplication since the advent of constraint-based phonology in Optimality Theory (OT; Prince & Smolensky [1993] 2004) is Base-Reduplicant Correspondence Theory (BRCT; McCarthy & Prince 1993b, 1995, 1999, et seq.). The original "Basic Model" of BRCT (McCarthy & Prince 1995) holds that the contents of the reduplicant are determined all at once via the interaction of (i) Base-Reduplicant (BR) faithfulness constraints, which demand identity between the segments in the reduplicant and their correspondents in the base, and (ii) markedness constraints, which penalize marked structures anywhere in the word, including the reduplicant. In the years since BRCT was first introduced, numerous other constraint-based theories of reduplication have been proposed, seeking to remedy various predictions of BRCT which were deemed incorrect. Most of these theories have advocated at least some degree of serial/cyclic derivation (e.g. Kiparsky 2010, Saba Kirchner 2010, McCarthy, Kimper, & Mullin 2012). One such theory is Morphological Doubling Theory (MDT), as proposed by Inkelas & Zoll (2005) [henceforth IZ]. In MDT, base and reduplicant are first computed separately according to individuated phonological grammars — called "cophonologies" — applying to identical/equivalent inputs, and then concatenated according to another individuated phonological grammar. This theory is embedded in Cophonology Theory (Orgun 1996, Inkelas, Orgun, & Zoll 1997, Anttila 2002, 2009, Inkelas & Zoll 2007), which is a highly cyclic derivative of OT.

One of the grounds on which IZ argue in favor of MDT over BRCT is with respect to the concept of "base-dependence", as introduced by IZ (pp. 92–95), and discussed extensively by Haugen & Hicks Kennard (2011) [henceforth HHK]. Base-dependence refers to a variety of putative patterns wherein phonological properties of the reduplicant or base crucially depend on information present only in the other constituent, as opposed to information that can be ascertained from the phonological input on its own. There are three general categories of (putative) base-dependent patterns which have been identified in the literature:

(1) *Putative types of base-dependence*
   a. Certain opaque reduplication-phonology interactions (IZ)
   b. Certain reduplicant shape alternations conditioned at the base-reduplicant juncture (IZ)
   c. Reduplicant shape alternations involving prosodic constituent copying (HHK)

IZ assert that MDT is unable to generate base-dependent reduplication patterns of any kind. This is a result of its separationist architecture, where base and reduplicant are computed without access to the other. Similar predictions are likely to follow from other serial/cyclic theories as well, e.g. reduplication in Stratal OT (Kiparsky 2010). On the other hand, BRCT, and generally any theory admitting BR correspondence and/or parallel evaluation, predicts the existence of all of these sorts of patterns. Therefore, a proper understanding of the empirical landscape regarding putative base-dependent patterns is critical in adjudicating between different theories of reduplication. Crucially, IZ claim that, when all the data is properly understood in its full context, no true cases of base-dependent reduplication patterns are attested. This paper will argue, largely following HHK, that that claim is incorrect.

This argument will be based around the Austronesian language Tawala (Papua New Guinea, Oceanic, Malayo-Polynesian; Ezard 1980, 1997), which displays an intricate phonologically-driven reduplicant shape alternation. As discussed by IZ (pp. 94–96, incl. n. 18), and argued by HHK, certain aspects of Tawala's reduplication pattern look as though they are conditioned across the base-reduplicant juncture in a way that would be challenging to MDT. IZ dismiss this possibility; however, they do not consider all of the data. HHK consider additional portions of the data (namely, the pattern which is identified as "Type D" below), and conclude that Tawala does rightly meet the requirements for base-dependence, and that MDT indeed cannot analyze this portion of the data.

This paper will seek to complicate this picture by demonstrating that there in fact is a feasible MDT analysis of the entirety of the data. Nevertheless, the take-away will be much the same as that advocated by HHK, because this analysis requires the use of a substantial amount of powerful machinery, which on its face saps MDT of much of its claimed restrictiveness. This will contrast with the BRCT analysis provided below — which is based heavily upon the analysis developed by Hicks Kennard (2004) [henceforth HK], but fixes several small deficiencies and expands on several underdeveloped points — which uses a relatively small set of well-motivated, surface-oriented constraints to fully describe the data. In other words, the argument in favor of BRCT over MDT will be transferred from an undergeneration argument (i.e., MDT can't analyze Tawala, as argued for by HHK) to an argument by parsimony and overgeneration (i.e., MDT can analyze Tawala but only using many complex, powerful devices which would seem to severely undercut its restrictiveness).

In the end, the MDT analysis presented below will remove base-dependent juncture-conditioned reduplicant shape alternations from the list of analytical properties that distinguish between BRCT and MDT. HHK also argue that base-dependent reduplicant shape alternations involving prosodic constituent copying exist, which I accept, and that MDT cannot derive these patterns without giving up on Richness of the Base (Prince & Smolensky [1993] 2004). I will briefly show that the sorts of technology that would be required to analyze Tawala in MDT would equally well be able to generate this kind of prosodic constituent copying. Therefore, I will conclude that the only type of putative base-dependent reduplication pattern that may still cleanly distinguish between BRCT and MDT is opaque reduplication-phonology interactions.

This paper is structured as follows. After introducing the Tawala data in Section 2, Section 3 will work through the BRCT analysis of Tawala's reduplication pattern. This analysis is based heavily on the one developed by HK (and followed by HHK), but with a number of refinements and expansions, namely, clarifying the definition of CONTIGUITY-BR and restricting the domain of the anti-repetition constraint *REPEAT in order to rule out several unnoticed problematic candidate outputs. This section will also include an explicit analysis of the various exceptional patterns found among CVV roots, and a formalization of the stress pattern and how it interacts with reduplication, which will be relevant for the subsequent MDT analysis. Section 4 will introduce the basic architecture and machinery of MDT, including how morphoprosodic structure can be manipulated to various effects. This will be illustrated with a fleshing out of IZ's analysis of certain aspects of reduplication in Javanese (Austronesian). Section 5 will introduce the novel MDT analysis of Tawala reduplication, inspired by that proposed in HHK, but diverging from it substantially. This will demonstrate that a convergent MDT analysis is possible (*contra* HHK), but only given substantial appeal to morphoprosodic structure *and* the introduction of an *ad hoc* derivational node with a highly complex cophonology. In this light, it becomes clear that these devices, which are employed in more limited respects by IZ, severely undercut much of the claimed restrictiveness of MDT's separationist architecture. Section 6 will summarize these arguments by comparing the BRCT and MDT analyses, and conclude with a brief treatment of prosodic constituent copying in MDT, showing that opaque reduplication-phonology interactions are in fact the only reliable grounds for distinguishing between BRCT and MDT on the basis of "base-dependence".

2 Tawala reduplication: The data

The Tawala durative exhibits four distinct reduplicant shapes (2), whose distribution is phonologically predictable (see HK, based on Ezard 1980, 1997; see also HHK).[1] These patterns are schematized and exemplified in brief in (3); they are exemplified exhaustively immediately below.[2] I omit stress marking for the time being, but will return to this apropos of the BRCT analysis in Section 3.2, and in the context of the MDT analysis in Section 5, where stress plays a more central role.

(2) *Reduplicant-shape alternations in Tawala*

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

(3) *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]

2.1 Consonant-initial roots

Type B is the most common root shape, and attests many examples, as given in (4). Many of these have morphologically complex bases, but this does not appear to have any impact on reduplication. From this data, we can see that roots beginning in CVCV reduplicate by copying that entire string. Type D forms also begin in CVCV, but they will be distinguished by another property, namely, having identical first two syllables.

[1] Type labels are given for convenience of reference, but have no formal status. Ezard (1980, 1997) includes other reduplicated forms in the language which do not belong to the durative category. These largely but not entirely fall into the same patterns as exhibited by the durative. I do not analyze these because it is not clear that they belong to any productive category.

[2] 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]. Following HK, I treat it as the latter.

(4) *Examples of Type B reduplication* (Ezard 1980:147, Ezard 1997:41)

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

There are many fewer CVV-initial roots. The primary pattern is Type A reduplication (5), which is characterized by copying of the root-initial consonant and the second root vowel. However, it must be admitted that there are just as many "exceptions" (6), that is, CVV-initial roots with other reduplicant shapes. Namely, the roots in (6.i) copy the first root vowel not the second, the roots in (6.ii) copy both root vowels, and the single example in (6.iii) displays a reduplicant vowel [i] which is distinct from both root vowels. Following the previous literature, I take the Type A forms to represent the productive treatment of CVV-initial roots. The exceptional patterns will be derived through lexically-indexed constraints (see Section 3.3 below).

(5) *Examples of Type A reduplication* (Ezard 1980:147, 1997:43)

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

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

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

2.2 Vowel-initial roots

The Type C pattern, where vowel-initial roots copy their initial VC- string, is exemplified further in (7). There does not appear to be any variation within this category.

(7) *Examples of Type C reduplication* (Ezard 1980:147, 1997:42)

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

2.3 Type D roots

The last type of reduplication pattern, Type D, is further exemplified in (8). Like Type B, this class consists of roots that begin in CVCV. However, unlike those roots exhibiting Type B reduplication, in these roots, the first two syllables are always identical. That identity will be crucial in generating their distinct behavior, namely, doubling the first root vowel in situ, as opposed to the otherwise attested types of leftward partial copying.

(8) *Examples of Type D reduplication* (Ezard 1997:44, HK:305)

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

3 Analyzing Tawala in Base-Reduplicant Correspondence Theory

HK develops an atemplatic BRCT analysis of reduplication in Tawala. In this section, I revise this analysis to account for two problems. First, in Section 3.1, I show that an unconsidered candidate for the CVCV pattern (Type B) requires a more articulated version of CONTIGUITY-BR (McCarthy & Prince 1995), which individuates violations for each locus of segment skipping, and further relativizes violations to consonants and vowels. Second, in Section 3.5, I show that a more holistic treatment of repeated identical syllables requires restricting the relevant constraint, *REPEAT, to word-initial position. This section also expands on HK's analysis in several ways, including formalizing the analysis of stress and its relationship to reduplication (Section 3.2), and integrating the exceptional copying patterns for CVV-initial roots into the larger analysis using lexically-indexed constraints (Section 3.3). In total, these changes have the effect of re-characterizing the system as targeting minimal reduplication (cf. Spaelti 1997, Hendricks 1999, *a.o.*), rather than a foot-sized reduplicant.

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

The examples of Type B and Type A reduplication are repeated here in (9) and (10). (I return to the "exceptional" CVV patterns in Section 3.3 below.)

(9) *Examples of Type B reduplication* (repeated from (4) above; Ezard 1980:147, Ezard 1997:41)

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

(10) *Examples of Type A reduplication* (repeated from (5) above; Ezard 1980:147, 1997:43)

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

HK derives Type A primarily through the operation of two constraints. The first constraint is *Repeat (11), an OCP-type constraint which bans adjacent identical syllables (HK:310; citing Yip 1995, 1998; cf. Leben 1973, Goldsmith 1976, Menn & MacWhinney 1984, McCarthy 1986, Suzuki 1998, *a.o.*). I am going to use a more specific version of this constraint, *Repeat(initial) (12), for reasons which will be motivated in Section 3.5 below. The other constraint is Align-Root-L (13) (HK:309; cf. McCarthy & Prince 1993a, Hendricks 1999), which has the effect of minimizing the length of the reduplicant by preferring the root's left edge to be as close to beginning of the word as possible.

(11) ***Repeat:** Assign one violation * for each pair of adjacent identical syllables.

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

(13) **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.

These constraints must outrank Contiguity-BR (HK:308; cf. McCarthy & Prince 1995) — the constraint requiring contiguous copying, preliminarily defined in the traditional manner in (14) — in order to permit Type A's discontiguous copying (15d). Contig-BR penalizes winning candidate (15d), because [bi] is not a contiguous substring of the base. In order for this candidate to be selected as optimal, this constraint must rank below both Align-Root-L, which prefers the shorter reduplicants in (15c,d) over the longer ones in (15a,b), and *Repeat(init), which penalizes (15c) for its initial repetition ($\#[\mathrm{be}]_\sigma[\mathrm{be}]_\sigma$).[3]

(14) **Contiguity-BR** ("*Don't skip–BR*"): Assign one violation * if the reduplicant doesn't correspond to a contiguous substring of the base. [*definition to be revised*]

[3] As identified by HK (p. 310), *Repeat(init) must only care about segmental content, and specifically not care suprasegmental properties like stress. This is because there are certain candidates it is responsible for eliminating, such as *[hù-hu.né.ya] — candidate (16b) with stress marks added (see Section 3.2) — where the identical syllables differ with respect to stress. See Zukoff (2020) for equivalent evidence from Ponapean.

(15) *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 (16d), which was not considered by HK (nor by HHK). With the current constraints, there should be no difference in the constraint interaction; we should continue to select the C_1V_2-copying candidate (16d) (à la Type A). But do note that we have a potential difference with respect to Contiguity, in that the two patterns are distinguished by what would be getting skipped in their discontiguous C_1V_2-copying candidate.

(16) *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	*(*)

Type A skips only vowels (base V_1): *bi-b*[*e*]*.i.ha*. On the other hand, for Type B, the problematic discontiguous candidate (16d) also skips a consonant (base C_2) in addition to a vowel (base V_1): **he-h*[*u.n*]*e.ya*. This distinction is preliminarily represented in the violations profiles of the Contiguity constraint in (15) vs. (16), where I have put a second violation in parentheses for candidate (16d), indicating that there could be more violation of the constraint by that candidate than by (15d). However, even with this potential extra violation, Contiguity's strict ranking below Align-Root-L will render this distinction moot.

Nevertheless, we can take advantage of this distinction if we make our definition of Contiguity more precise. The definitions immediately below assess violations for each individual locus of skipping, rather than for each discontinuity as a whole. This type of definition allows us to distinguish between skipped consonants (17) and skipped vowels (18). (It is evident that HK had something like this mode of violation assessment in mind in her use of Contig-BR, though this is never made explicit; see (63) below for further discussion.)

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

(18) **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 individuated Contig constraints as shown in (19), with ContigC-BR ranked highest, we derive the right results (20, 21). Splitting up Contig has no effect on Type A (20), because there's no medial consonant to skip, so the evaluation looks exactly the same as before. But for Type B (21), the new high-ranked ContigC-BR can rule out the discontiguous copying candidate (21d) because its discontiguity includes a consonant (unlike Type A). *Repeat(init) and ContigC-BR now respectively rule out both minimal copying candidates (21c,d), and so the next shortest possible (C-contiguous) reduplicant (21b) is selected. Tableau (21) adds an additional candidate, (21e), which skips only the consonant. Like losing candidate (21d), this candidate is also ruled out by ContigC-BR.

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

(20) *Type A reduplication with individuated* 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	*

(21) *Type B reduplication with individuated* Contiguity

/red, huneya/			*Repeat(init)	ContigC-BR	Aln-Rt-L	ContigV-BR
a.		hu.ne.ya-hu.ne.ya			6!	
b.	☞	hu.ne-hu.ne.ya			4	
c.		hu-hu.ne.ya	*!		2	
d.		he-hu.ne.ya		*!	2	*
e.		hu.e-hu.ne.ya		*!	3	

One other type of alternative candidate which could yield a short reduplicant without running afoul of *Repeat(init) and/or ContigC-BR would be one whose reduplicant does not include the initial syllable in the first place, as in (23b,c). We can rule these candidates out by ranking Anchor-L-BR (22) (HK:307; cf. McCarthy & Prince 1995, Shaw 2005), which requires copying from the left edge, above Align-Root-L, as shown in (23).

(22) **Anchor-L-BR:** Assign one violation * if the leftmost segment of the reduplicant does not correspond to the leftmost segment of the base.

(23) *Type B reduplication and* Anchor-L-BR

/red, huneya/			Anchor-L-BR	Align-Root-L
a.	☞	hu.ne-hu.ne.ya		4
b.		ne-hu.ne.ya	*!	2
c.		u-hu.ne.ya	*!	1

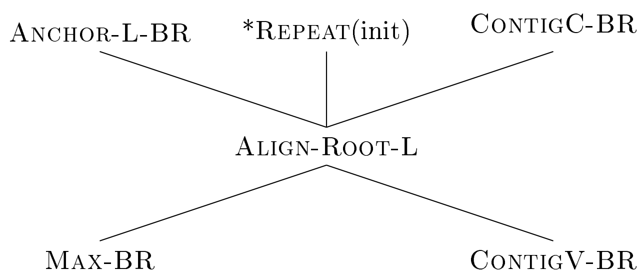
The Hasse diagram in (26) summarizes the rankings developed thus far. It is supplemented with the constraint Max-BR (24) (McCarthy & Prince 1995; HK:308), which prefers maximal copying from the base into the reduplicant. This constraint essentially trades off with the size-restrictor constraint Align-Root-L, because the shorter the reduplicant, the fewer segments that have been copied. Since Align-Root-L in fact limits the size of the reduplicant, we know that Align-Root-L $\gg$ Max-BR, as demonstrated in (25). Max-BR plays no active role in the analysis, so it will be omitted from here on out.

(24) **Max-BR:** Assign one violation * for each segment in the base which does not have a correspondent in the reduplicant.

(25) *Type A reduplication and* Max-BR

/red, beiha/			Align-Root-L	ContigV-BR	Max-BR
a.		be.i.ha-be.i.ha	5!		0
b.		be.i-be.i.ha	3!		2
c.	☞	bi-be.i.ha	2	*	3

(26) *Interim ranking summary for Tawala reduplication*



3.2 A note on feet and stress

All of the constraints employed in the present analysis have direct analogs in HK's analysis, albeit with some revisions as just detailed. One constraint which HK uses and takes to be active in the analysis is "Anchor-L(Wd,Ft)" (HK:308):

(27) **Anchor-L(Wd,Ft):** Anchor the left edge of the prosodic word to the left edge of a foot.

This constraint would effectively require a foot to coincide with the beginning of the reduplicant, since the reduplicant surfaces in word-initial position. She asserts (p. 308) that this constraint not only accounts for certain facts about stress in reduplicated forms (see immediately below), but that it is implicated in the calculation of reduplicant shape; namely, that it helps generate disyllabic copying in Type B, preferring a disyllabic copying candidate like [(hù.ne).hu.(né.ya)] (28a) to a trisyllabic copying candidate like *[hu.(nè.ya).hu.(né.ya)] (28b). (Parentheses indicate foot boundaries.) However, it is clear from the subsequent portions of HK's analysis that this is not necessary. HK (p. 309) later uses Align-Root-L to prefer disyllabic copying to an alternative trisyllabic copying candidate, *[(hù.ne).(yà.hu).(né.ya)] (28c), which has a left-aligned foot, and indeed perfect iterative trochaic footing. This same constraint will equally well rule out *[hu.(nè.ya).hu.(né.ya)] (28b). Note that she is explicit that this constraint is not Red = Foot, so the mismatches between foot boundaries and reduplicant boundaries do not effect violation assessment.

(28) *HK's foot-based analysis*

/red, huneya/	Align-Root-L	Anchor-L(Wd,Ft)
a. ☞ (hù.ne).hu.(né.ya)	4	
b. hu.(nè.ya).hu.(né.ya)	6!	*
c. (hù.ne).(yà.hu).(né.ya)	6!	

There does not appear to be any other portion of HK's analysis of reduplicant shape which relies on Anchor-L(Wd,Ft). Therefore, this has demonstrated, *contra* HK, that this constraint is not involved in generating reduplicant shape alternations in Tawala. This is what is assumed in the analysis developed in this paper. Nevertheless, as at least implied by HK, a constraint of this sort could reasonably be invoked to account for some divergent stress facts in reduplicated words.

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

(29) *Tawala stress*

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

While alternating rhythm of this sort has traditionally been understood in terms of iterative foot assignment — this would be right-to-left trochees — it can alternatively be understood as the result of a desire to avoid adjacent syllables that are both stressed (a clash) or both unstressed (a lapse). The one exception to the generalization about alternating rhythm in Tawala in (29b) is found in reduplication: there is a requirement that the initial syllable of the reduplicant be stressed, even if this leads to a lapse (30a–c) or a clash (30d). (Note that the prefixes to the left of the reduplicant appear to be outside the stress domain.)

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

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

Following HK, we could account for this with feet, by saying that a foot is constructed beginning on the first syllable of the reduplicant and terminating before the first syllable of the base, driven by Anchor-L(Wd,Ft). This would be a binary foot in cases like (30a–c) but a unary foot in cases like (30d). Therefore, foot binarity would not seem to be playing any role in determining reduplicant shape.[4]

But we can also do this straightforwardly using the foot-free stress constraints below:[5]

(31) *Stress constraints*

a. **StressL-Red:** Assign one violation * if the reduplicant-initial syllable is unstressed.
b. ***Clash:** Assign one violation * if for each pair of adjacent stressed syllables. (*$\acute{\sigma}\acute{\sigma}$)
c. ***Lapse:** Assign one violation * if for each pair of adjacent unstressed syllables. (*$\sigma\sigma$)

If StressL-Red outranks the constraints on alternating rhythm, we will generate clashes and lapses just in reduplication (32, 33). But note from losing candidates (32c) and (33c), where the reduplicant is extended in order to avoid the stress problems, that the stress constraints have no impact on what gets copied (cf. Zukoff 2016, 2020). This reiterates that the language is *not* treating the (binary) foot as the target shape in reduplication. (The numerals next to the candidates indicate the schematic representation of stress in the candidate: "1" for primary stress, "2" for secondary stress, and "0" for unstressed.)

(32) *Clashes in reduplication*

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

(33) *Lapses in reduplication*

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

In the lapse cases, monosyllabic reduplicants could also solve the stress problems, and with better satisfaction of Align-Root-L. However, the constraints prohibiting CV-copying for CVCV-initial roots developed above prevent this from occurring. Tableau (34) demonstrates that this follows from the rankings motivated above. Interestingly, there is no direct interaction between these constraints and the placement of stress, and these candidate comparisons do not reveal any new rankings.

[4] It is also worth worrying about the role of Anchor-L(Wd,Ft) outside of reduplication. As noted in (29), we observe alternating rhythm (from the right) outside of reduplication, which results in unstressed initial syllables in words with an odd number of syllables. Since we observe clashes in reduplication (30d), which would be triggered by Anchor-L(Wd,Ft), it is likely that the ranking would predict unary feet and clashes in this position outside of reduplication as well, contrary to fact.

[5] On foot-free constraint-based approaches to stress, see, e.g., Elenbaas & Kager (1999), Gordon (2002), Stanton (2014, 2015), Zukoff (2016); following Prince (1983), Selkirk (1984), *a.o.*

(34) *Interaction between stress constraints and reduplication-shape alternation constraints*

/RED, dewaya/		*REPEAT(init)	CNTGC-BR	ALIGN-ROOT-L	*LAPSE
a. ☞ dè.wa-de.wá.ya	[20-010]			4	*
b. dè-de.wá.ya	[2-010]	*!		2	
c. dà-de.wá.ya	[2-010]		*!	2	

We now see that feet (or indeed stress in general) are not required in order to account for reduplicant shape alternations in Tawala, nor are they necessary to account for aberrant stress facts in reduplication. Therefore, it is reasonable to jettison them from the analysis. Taking this approach clarifies the nature of the reduplicant shape alternations in this language: they are driven by a desire to attain the smallest possible reduplicant shape (via ALIGN-ROOT-L), subject to interaction with higher-ranked constraints demanding a longer reduplicant in particular contexts. This is the ethos of the "atemplatic" approach to reduplication (e.g. Spaelti 1997, Hendricks 1999, Riggle 2006, Zukoff 2016, 2017a), which HK is seeking to implement (p. 309). This allows us to eliminate prosodic templates, either in the form of underlying representations (McCarthy & Prince 1986) or violable constraints (McCarthy & Prince 1993b, 1995), from the analysis of Tawala, and indeed clarifies that even general prosodic constraints are not actively involved in the analysis of these particular reduplicant shape alternations.

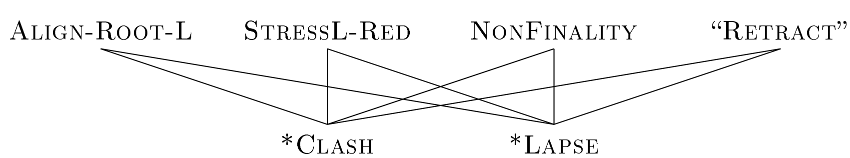
Before moving on, it is worth pausing briefly on the position of primary stress. As mentioned above (29), primary stress is normally penultimate. We can derive this through the interaction between *LAPSE and NONFINALITY (*"Don't stress the final syllable"*): one of the final two syllables must bear a stress, but it can't be the final, so it must be the penult. The one exception to penultimate primary stress (Ezard 1997:45) is when the antepenult has a lower vowel than the penult and the penult is onsetless or has an onset [l]. Whatever the right formulation of the markedness constraint(s) driving this retraction — which I will shorthand as "RETRACT" — that constraint outranks *LAPSE, allowing for stress retraction to the antepenult, even though it creates a lapse at the right edge. Because of this retraction, we get a clash also in certain Type A reduplicated forms like [bì-bé.i.ha]. It must therefore also be the case that "RETRACT" outranks *CLASH, as shown in (35).

(35) *Retraction and clashes*

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

The rankings used in this subsection are summarized in (36). The only point of contact they have with the rankings for the remainder of the analysis, which can be found in (62) below, is the constraint ALIGN-ROOT-L.

(36) *Ranking summary for Tawala stress*



3.3 Exceptional CVV patterns

As mentioned above, Type A C_1V_2-copying is not the only attested reduplication pattern for CVV-initial roots. The table in (37) shows these alternative patterns.

(37) *Other reduplication patterns for CVV roots* (repeated from (6) above; Ezard 1980:147, 1997:43)

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

The tableau in (38) shows the analysis developed above for Type A reduplication, which we are treating as the default pattern for CVV-roots.[6] This tableau includes the three viable candidates for CVV-roots: $C_1V_1V_2$-copying (the (a) candidate), C_1V_1-copying (the (b) candidate), and C_1V_2-copying (the (c) candidate). These are the three copying patterns for CVV-roots that have multiple attestations. (I will put aside for now the single C_1i-reduplication case, /peu/ → [pi-pe.u] 'to fall'.) What we can observe by looking at (38) is that different permutations of the ranking of *Repeat(init), Align-Root-L, and ContigV-BR would select different winners, because each of the candidates exhibits a minimal constraint conflict with the others.

(38) *Analysis of Type A reduplication for CVV-roots* (adapted from (20))

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

Because of the nature of this constraint conflict, we can use lexically-indexed constraints to derive the exceptional patterns (Kraska-Szlenk 1997, Pater 2000, 2007, 2009, Becker 2009, Coetzee 2009, Nazarov 2019, *a.o.*). Indexed constraints are constraints which apply only to derivations containing morphemes that share a special index or diacritic with that constraint. Pater (2007, 2008, 2009) and Becker (2009) show that a process of "constraint cloning" embedded in a learning procedure like Recursive Constraint Demotion (Tesar & Smolensky 1998, 2000) can restrictively generate indexed constraints that resolve cases such as these where different patterns seem to require inconsistent rankings. (See also Zukoff 2017a,b for an application of constraint cloning in a similar case of reduplication in Ancient Greek.) The way to use indexed constraints in this case is to posit special versions of the lower-ranked constraints, which are annotated with indices shared by the exceptional roots in (37) and then ranked above *Repeat(init). The definitions of these cloned constraints are given in (39) and (40). Note that, in this case, each constraint is assigned a distinct index.

(39) **Align-Root-L$_i$:** (cf. (13))
Assign one violation * for each segment which intervenes between the left edge of the root and the left edge of the word **if the input contains a morpheme with index** i.

(40) **ContiguityV-B(→)R$_j$:** (cf. (18))
For a reduplicant string $r_1 ... r_n$ standing in correspondence with a base string $b_1 ... b_n$, assign one violation * for each **vowel** between b_1 and b_n which lacks a correspondent in $r_1 ... r_n$ **if the input contains a morpheme with index** j.

[6] Tableau (38) omits the full-copying candidate (20a), to make for clearer comparison with the exceptional cases analyzed below. Additionally, this tableau and those that follow omit ContigC-BR, because this constraint plays no role with CVV-roots.

Tableaux demonstrating the indexed constraint analysis are given in (41–43). To derive the C_1V_1-copying cases, as shown in (41), we need to assume that such roots are indexed to *both* constraints, i.e. indexed with i and j. Since these roots have the index i, Align-Root-L_i will rule out $C_1V_1V_2$-copying (41a), because this candidate would have a longer reduplicant than the others. Since these roots *also* have the index j, ContigV-BR_j will rule out C_1V_2-copying (41c), which is the productive Type A reduplication pattern, because it skips a vowel. Therefore, for CVV-roots indexed with both i and j, the optimal output is C_1V_1-copying (41b), even though this violates the normally decisive *Repeat(init).

(41) *Analysis of exceptional C_1V_1-copying for CVV-roots*

/red, nei$_{ij}$/	Aln-Rt-L_i	CntgV-BR_j	*Repeat(init)	Aln-Rt-L	CntgV-BR
a. ne.i-ne.i	3!			3	
b. ☞ ne-ne.i	2		*	2	
c. ni-ne.i	2	*!		2	*

When a CVV-root is indexed only with j, Align-Root-L_i is inoperative. This means that the $C_1V_1V_2$-copying candidate (42a) only fares worse than its competitors on the regular version of Align-Root-L, which continues to be low-ranked. Since these roots *are* indexed to j, ContigV-BR_j again rules out Type A C_1V_2-copying (42c). However, now that Align-Root-L_i is not invoked, *Repeat(init) is once again high-enough ranked to eliminate the C_1V_1-copying candidate (42b). This allows the $C_1V_1V_2$-copying candidate (42a) to emerge as the winner.

(42) *Analysis of exceptional $C_1V_1V_2$-copying for CVV-roots*

/red, woe$_j$/	Aln-Rt-L_i	CntgV-BR_j	*Repeat(init)	Aln-Rt-L	CntgV-BR
a. ☞ wo.e-wo.e				3	
b. wo-wo.e	*n/a*		*!	2	
c. we-wo.e		*!		2	*

Tableau (43) is a sanity check, demonstrating that this analysis does not have any deleterious effects for the productive pattern. For roots which lack any index, and will thus behave purely according to the productive constraint ranking, both Align-Root-L_i and ContigV-BR_j have no effect. This allows for the two highest-ranked un-indexed constraints, *Repeat(init) and Align-Root-L, to respectively rule out $C_1V_1V_2$-copying (43a) and C_1V_1-copying (43b), in favor of C_1V_2-copying (43c), the regular Type A pattern.

(43) *Analysis of Type A reduplication for CVV-roots*

/red, beiha/	Aln-Rt-L_i	CntgV-BR_j	*Repeat(init)	Aln-Rt-L	CntgV-BR
a. be.i-be.i.ha				3!	
b. be-be.i.ha	*n/a*	*n/a*	*!	2	
c. ☞ bi-be.i.ha				2	*

Ezard (1980:148) identifies two or three forms which might be treated as exceptional patterns to CVCV-roots, which normally undergo Type B CVCV-copying. These are listed in (44).[7] I will return to the form *wi-waya* below. The other pattern (44.i) amounts to another case of C_1V_1-copying, exactly equivalent to what was just observed in CVV-roots, and can be handled in exactly the same manner, via constraint indexation.

[7] For some reason, Ezard (1997) does not include any of the forms in (44) in the apparently exhaustive list of reduplicated forms in pages 41–44. Yet, *ha-hale* (Ezard 1997:92, ex. 7(12)) and *ni-niye-ya* (Ezard 1997:245, ex. 16(53)) were both found in examples in later sections of the book, identified using automated search.

(44) *Exceptional reduplication patterns for CVCV roots* (Ezard 1980:148)

Simplex		**Reduplicated (durative)**	
i. C_1V_1-reduplication			
hale	→	*ha-hale*	'to throw'
niye-ya	→	*ni-niye-ya*	'to bring (tr.)'
ii. C_1i-reduplication(?)			
waya	→	*wi-waya*	'to take it'
waiya	→	*wi-waiya*	'to take it' (dialectal)

There exist two workable analyses, which are slight variants of one another. The reason for this underdetermination is because of the operation of the un-indexed constraint CONTIGC-BR, included once again in the following tableaux, with respect to CVCV roots. The first option, demonstrated in (45), is one which is precisely parallel to the treatment of C_1V_1-copying for CVV-roots in (41), namely, indexation of these roots to both indexes i and j. Under this assumption, ALIGN-ROOT-L_i rules out the four- and three-segment long reduplicant candidates (45a,b), and CONTIGV-BR_j rules out the skipping C_1V_2-copying candidate (45d). This analysis thus correctly selects C_1V_1-copying (candidate (45c)) for this set of roots.

(45) *Analysis of exceptional C_1V_1-copying for CVV-roots: option 1 — double indexation*

/RED, hale$_{ij}$/	ALN-RT-L_i	CNTGV-BR_j	CNTGC-BR	*RPT(init)	ALN-RT-L	CNTGV-BR
a. hale-hale	4!				4	
b. ha.e-hale	3!		*		3	
c. ☞ ha-hale	2			*	2	
d. he-hale	2	*!	*		2	*

However, since the C_1V_2-copying candidate for a CVCV-root would also violate the un-indexed constraint CONTIGC-BR (45d/46d), it would also be sufficient to assume indexation just to i, activating ALIGN-ROOT-L_i, and allow CONTIGC-BR to be what rules out C_1V_2-copying (46d). This requires fixing the ranking of CONTIGC-BR above *REPEAT(init), but this is fully consistent with all the other data. ALIGN-ROOT-L_i remains crucial, because it is the only thing which can prefer desired candidate (46d) to the CVCV-copying candidate (46a). It appears as though these two analyses are fully equivalent, and so I will not endeavor to choose between them.

(46) *Analysis of exceptional C_1V_1-copying for CVV-roots: option 2 — single indexation +* CONTIGC-BR

/RED, hale$_i$/	ALN-RT-L_i	CNTGV-BR_j	CNTGC-BR	*RPT(init)	ALN-RT-L	CNTGV-BR
a. hale-hale	4!	n/a			4	
b. ha.e-hale	3!		*!		3	
c. ☞ ha-hale	2			*	2	
d. he-hale	2		*!		2	*

This indexed-constraint analysis thus successfully generates the three main types of copying for CVV-roots, as well as the minority C_1V_1-copying pattern for CVCV-roots. The analysis thus far has nothing to say about /peu/ → [pi-pe.u] 'to fall'. Given that this is an isolated example, it may be reasonable to assume this is a listed, frozen form of some sort. If we were to want to ascribe to it a phonological analysis, we could again use indexed constraints to analyze it as a case of phonological fixed segmentism (Alderete et al. 1999), which is essentially a type of emergence of the unmarked (McCarthy & Prince 1994) interaction where featural contrasts are neutralized in the reduplicant. I refrain from exploring the details of such an analysis here for reasons of space.

This analysis could, though, be relevant also to the irregular pattern observed in *wi-waya*, since it looks to have an unwarranted [i] in the reduplicant. There do, however, seem to be alternative explanations here. First, Ezard (1980:148) identifies /waya/ → [wi-waya] as existing beside /waiya/ → [wi-waiya] from an alternative dialect. This latter form would represent the regular treatment of a CVV-root (Type A), where the reduplicant [i] corresponds to root-V_2. In any event, this would certainly seem to provide a diachronic explanation, coupled with some putative change of *aiya to *aya*. This could be ported into a synchronic analysis by appealing to some degree of abstractness and/or opacity. The second option is that this form is simply erroneous, as it appears to be completely absent from Ezard (1997).

Before moving on, it should be reiterated that the number of CVV-roots belonging to each reduplication pattern is essentially equivalent: there are 5 Type A C_1V_2-copying roots roots ((5)/(10)) vs. 3 $C_1V_1V_2$-copying roots vs. 3 C_1V_1-copying roots ((6)/(37)). Ezard (1980, 1997) is concerned more with description than analysis, and effectively does not take a stance on whether any of the three patterns could be rightly characterized as the "productive" one. In this paper, I have followed HK in assuming Type A to be the productive pattern, in large part because its analysis is most consistent with what is needed for the patterns of other root shapes. Nevertheless, we are lacking the sorts of theory-external evidence that could help disambiguate, such as "wug" testing (Berko 1958) or other sociolinguistic/dialectal information. Pending such evidence, the treatment of the different CVV types with respect to productivity advanced in this paper must be viewed as tentative.

3.4 Vowel-initial roots

The attested examples of the Type C pattern, where vowel-initial roots copy their initial VC- string, are repeated in (47). This pattern follows completely from existing rankings. Among the shortest possible reduplicants, *Repeat(init) rules out copying just the initial vowel (48c), and Anchor-L-BR rules out copying just the base-second consonant (48d). Lower-ranked Align-Root-L selects the next shortest properly-anchored reduplicant, which is the desired VC-copying candidate (48b).

(47) *Examples of Type C reduplication* (repeated from (7) above; Ezard 1980:147, 1997:42)

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

(48) *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 (49) (HK:306; cf. Itô 1989, Prince & Smolensky [1993] 2004) turns out to be unnecessary, even though we might have expected it to be responsible for eliminating (48a) and (48c), as it is in HK's analysis.

(49) **Onset:** Assign one violation * for each onsetless syllable.

3.5 Type D reduplication, *Repeat, and TETU

We've now used *Repeat (or the more specific *Repeat(init)) to account for both the lack of C_1V_1-reduplication in consonant-initial roots (Types A & B), and the lack of V_1-reduplication for vowel-initial roots (Type C). HK (followed by HHK) also uses it to help analyze Type D reduplication, of which the attested examples are repeated in (50) below.

(50) *Examples of Type D reduplication* (repeated from (8) above; Ezard 1997:44, HK:305)

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

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 stem-affix junctures. This is illustrated for the root /totogo/ → [to.to.go] ‘be sick’ (Ezard 1997:33, HK:305) in (51). None of the conceivable means of avoiding the repetition are employed, and the violation is tolerated. This means that the avoidance of repeated identical (initial) syllables in reduplication in Tawala is an instance of the emergence of the unmarked (TETU; McCarthy & Prince 1994), as argued by HK and HHK.

(51) *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	*!*			

HK and HHK analyze the Type D vowel-doubling pattern as an extreme instantiation of TETU: the reduplicant surfaces as an infixed copy of base-V_1 in order to break up the root’s repeated syllables. Infixal reduplication provides a unique way to satisfy *Repeat that is not available in non-reduplicative constructions, i.e., infixation via violation of Align-Red-L (52) (HK:307; cf. McCarthy & Prince 1993a, Hendricks 1999).[8] Since reduplication is generally prefixal not infixal, we know that Align-Red-L must dominate Align-Root-L (HK:307–309; cf. Zukoff 2023).

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

There are numerous types of viable candidates here, which I will consider step-by-step. (Superscripts indicate correspondence.) Following HK and HHK, the candidates which are of primary interest here as those with initial repetitions. Candidate (53c) is the otherwise expected CVCV-copying candidate (Type B). This candidate obviously incurs a *Repeat(init) violation, as well as three *Repeat violations for its four identical syllables in a row. Copying only the first syllable (53b) reduces the number of *Repeat violations from four to three, but does not improve on *Repeat(init). A surface identical candidate *[$\underline{g^1u^4}$.-g^1u^2.g^3u^4.y^5a^6], which copies the consonant from the first syllable but the vowel from the second (akin to Type A) does no better because $V_1 = V_2$, and additionally violates both Contiguity constraints. Since local vowel doubling is evidently preferred, it must be the case that Align-Red-L must rank below at least one of the *Repeat constraints. I will demonstrate below that it must be *Repeat(init).

(53) *Type D reduplication: V-doubling > initial repetitions*

/red, $g^1u^2g^3u^4y^5a^6$/	*Rpt (init)	Anchor-L-BR	Align-Red-L	Align-Root-L	*Rpt
a. ☞ g^1-$\underline{u^2}$-.u^2.g^3u^4.y^5a^6			1		
b. $\underline{g^1u^2}$.-g^1u^2.g^3u^4.y^5a^6	*!			2	**
c. $\underline{g^1u^2.g^3u^4}$.-$g^1u^2$.$g^3u^4$.$y^5a^6$	*!			4	***

[8] Infixal canidates will also violate Contiguity-IO. Therefore, all ranking arguments pertaining to Align-Red-L in the following tableaux apply also to Contiguity-IO, at least the version pertaining to segments. I will not consider this constraint further.

The main other alternative set of candidates are those which avoid an initial repetition via mis-anchoring. Candidate (54b) avoids the initial repetition by copying only the vowel from the first syllable of the base. Candidate (54c) copies the final syllable of the base, which is distinct from the identical base-initial syllables. This sort of mis-anchored / non-local copying can avoid the *Repeat(init) violation, but only at the expense of violating Anchor-L-BR. Therefore, as long as Anchor-L-BR outranks Align-Red-L, we can correctly prefer the infixal doubling candidate.

(54) *Type D reduplication: V-doubling > mis-anchoring*

/red, $g^1u^2g^3u^4y^5a^6$/	*Rpt (init)	Anchor-L-BR	Align-Red-L	Align-Root-L	*Rpt
a. ☞ g^1-$\underline{u^2}$-.u^2.g^3u^4.y^5a^6			1		
b. $\underline{u^2}$.-g^1u^2.g^3u^4.y^5a^6		*!		1	*
c. $\underline{y^5a^6}$.-g^1u^2.g^3u^4.y^5a^6		*!		2	*

Note that, in order to prefer desired candidate (54a) over the non-infixal candidates in (54), it must be the case that (54a) does *not* violate Anchor-L-BR. This is not obvious. If the base comprised the entire non-reduplicative string (Lunden 2004), as assumed by HK (p. 317) and HHK (p. 19), or some other constituent (Shaw 2005, Haugen 2009) that includes the leftmost segment of the root, all three candidates would have a single violation of Anchor-L-BR. Desired candidate (54a)'s violation of Align-Red-L would then be fatal, and (54b) would be incorrectly selected as winner.[9] If, however, we assume that the base of reduplication initiates with the segment immediately following the reduplicant — as per, for example, McCarthy & Prince (1993b) and Urbanczyk (1996) — then (54a) should be considered properly-anchored, thus satisfying Anchor-L-BR. To the extent that this analysis is correct, this provides evidence in favor of this view of the base of reduplication.

The major crucial rankings required and motivated by this portion of the analysis are as follows:

(55) **Ranking:** *Repeat(init), Anchor-L-BR ≫ Align-Red-L

It is also worth considering other possible infixal candidates. Candidate (56d) copies the VC sequence straddling the identical first and second syllables of the base, which re-introduces the initial repetition, and thus violates *Repeat(init). Candidate (56c) copies the VC sequence straddling the second and third syllables of the base, and places it after the root-initial consonant. This succeeds in avoiding the initial repetition, but again only at the expense of Anchor-L-BR. Candidate (56b), which is surface-identical to (56c), copies the vowel of the first syllable of the base and the consonant of the third syllable. This avoids the Anchor-L-BR violation, because the reduplicant-initial segment is now in correspondence with the base-initial segment, following the discussion above. However, this comes at the expense of having skipped the consonant of the base's second syllable, fatally violating ContigC-BR, which is entered into this tableau. All of these candidates are harmonically bounded by the winner given this constraint set. The constraints Max-BR and Onset would both favor the losers in this tableau, but it has already been shown that these are not high-ranked enough to have any effect.

(56) *Type D reduplication: V-doubling > VC-infixation*

/red, $g^1u^2g^3u^4y^5a^6$/	*Rpt (init)	Anchor-L-BR	ContigC-BR	Align-Red-L	Align-Root-L	*Rpt
a. ☞ g^1-$\underline{u^2}$-.u^2.g^3u^4.y^5a^6				1		
b. g^1-$\underline{u^2.y^5}$-u^2.g^3u^4.y^5a^6			*!	1		
c. g^1-$\underline{u^4.y^5}$-u^2.g^3u^4.y^5a^6		*!		1		
d. g^1-$\underline{u^2.g^3}$-u^2.g^3u^4.y^5a^6	*!			1		**

[9] *Repeat would favor the desired candidate, and thus select it if ranked above Align-Red-L. However, it will be shown below in (60) that the context-free *Repeat constraint must rank below Align-Red-L to derive other facts about the system.

Lastly, we can consider candidates which resolve the repeated syllables through Input-Output faithfulness violations. For example, candidate (57b) makes a featural change in the first syllable of the base, while the reduplicant consists of a faithful realization of the underlying root-initial CV- string. This results in a short reduplicant and no sequences of identical syllables. However, as demonstrated in (51) above, this sort of faithfulness violation in service of *Repeat(init) is not tolerated. This candidate also incurs violations of BR-identity constraints (not recorded in the tableau), for the mismatch in features between the reduplicant and the first syllable of the base. Candidate (57c), which is surface-identical to (54b) above, deletes the root-initial consonant, and reduplicates as if it is VC-initial (as in Type C). This Input-Output deletion is also not tolerated. Since the doubling candidate (57) is evidently preferred, we know that the Input-Output faithfulness constraints outrank Align-Red-L. This ranking also follows from transitivity through other rankings already established.

(57) *Type D reduplication: V-doubling > IO-faithfulness violation*

/red, $g^1u^2g^3u^4y^5a^6$/			Faith-IO	*Rpt (init)	Anchor-L-BR	Align-Red-L	Align-Root-L	*Rpt
a	☞	g^1-$\underline{u^2}$-.u^2.g^3u^4.y^5a^6				1		
b.		$\underline{g^1u^2}$.-g^1i^2.g^3u^4.y^5a^6	*!				2	
c.		$\underline{u^2.g^3}$-u^2.g^3u^4.y^5a^6	*!				2	*

As can be verified from the tableaux in (53–57), 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:

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

	Underlying initial repetition	Underlying non-initial repetition
a.	***Repeat(init) prediction:** V-doubling infixation to avoid *a word-initial* repetition	
	✓ /$\#C_1V_1C_2V_2(X)$/ → [$\#C_1$-$\underline{V_1}$-$V_1C_2V_2(X)$]	☒ /$XC_1V_1C_2V_2(X)$/ → [XC_1-$\underline{V_1}$-$V_1C_2V_2(X)$]
b.	***Repeat prediction:** V-doubling infixation to avoid *any* repetition	
	✓ /$\#C_1V_1C_2V_2(X)$/ → [$\#C_1$-$\underline{V_1}$-$V_1C_2V_2(X)$]	☑ /$XC_1V_1C_2V_2(X)$/ → [XC_1-$\underline{V_1}$-$V_1C_2V_2(X)$]

There is at least one relevant base which can disambiguate between these two predictions: *kilolo* 'urinate' → *kilo-kilolo* 'urinating' (**kil-o-olo*) (Ezard 1997:61, HK:307). Since it shows prefixation (Type B reduplication) rather than infixation (Type D reduplication), we know that we must be dealing with *Repeat(init), not general *Repeat. We can see the argument most clearly if we try getting rid of *Repeat(init), as in (59). Since the reduplicant is by default a prefix, we know that Align-Red-L ≫ Align-Root-L (cf., e.g., Zukoff 2023). From the Type D cases, we know that some version of *Repeat must dominate Align-Red-L. Implementing these rankings, we incorrectly derive infixation into the second syllable of the root (59c):

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

/red, $k^1i^2l^3o^4l^5o^6$/			*Repeat	Align-Red-L	Align-Root-L
a.	☹	$\underline{k^1i^2.l^3o^4}$.-$k^1i^2$.$l^3o^4$.$l^5o^6$	*!		4
b.		$\underline{k^1i^2}$.-k^1i^2.l^3o^4.l^5o^6	*!*		2
c.	💣	k^1i^2.l^3-$\underline{o^4}$.-o^4.l^5o^6		3	

This shows that the general *Repeat constraint must rank below Align-Red-L. But if this were the only active *Repeat constraint, we would no longer be able to generate Type D infixation at all. Allowing *Repeat(init) to rank high while general *Repeat is ranked low (or non-existent), we derive the correct results for /kilolo/, as shown in (60). *Repeat(init) correctly rules out C_1V_1-reduplication (60b), but does not penalize retaining the underlying non-initial repetition in desired candidate (60a). This allows Align-Red-L to eliminate the infixal candidate (60c).

(60) /kilolo/ → [<u>kilo</u>-kilolo] *(Type B reduplication) with* *REPEAT(init)

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

Using *REPEAT(init) also comports with the one attested vowel-initial root with identical V_1 and V_2, which attests Type C reduplication that creates medial identical syllables: *o.to.wi* 'make an appointment' → *<u>o.t</u>-o.to.wi* (Ezard 1980:147; IZ:95, HHK:26).[10] In (61), we see that general *REPEAT must be ranked *below* ALIGN-ROOT-L, or else candidate (61a), which additionally copies V_2 to avoid the medial repetition, would be preferred to desired candidate (61b).

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

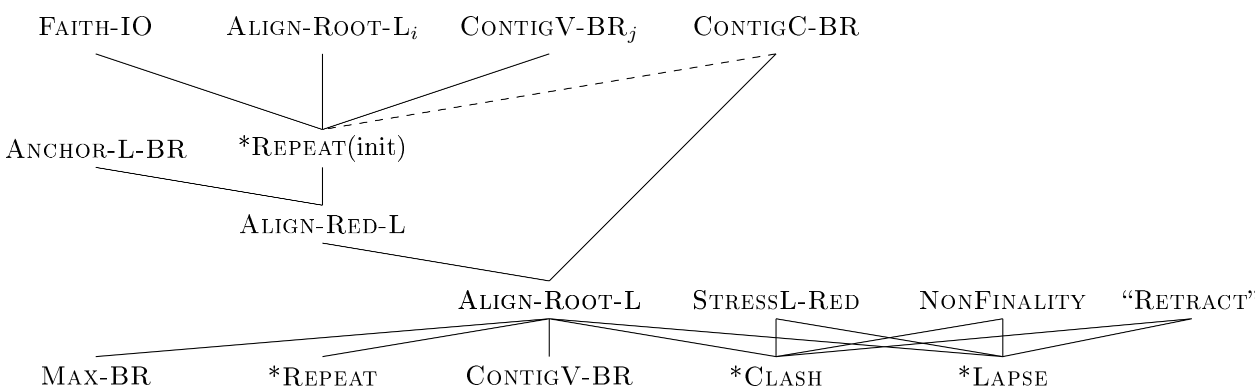
/RED, otowi/	*REPEAT(init)	ANCH-L-BR	ALN-RED-L	ALN-RT-L	*REPEAT
a. <u>o.to</u>.-o.to.wi				3!	
b. ☞ <u>o.t</u>-o.to.wi				2	*
c. <u>o</u>.-o.to.wi	*!			1	*
d. <u>t</u>-o.to.wi	*!	*!		1	*
e. <u>w</u>-o.to.wi		*!		1	
f. o.t-<u>o</u>-o.wi			2!		

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.

3.6 Local summary

The rankings motivated above are summarized in (62):

(62) *Ranking summary for Tawala reduplication*



This analysis revises and expands that of HK in a number of ways. First, it splits CONTIGUITY-BR into separate constraints, one banning consonant skipping, the other banning vowel skipping. This fixes HK's unrecognized prediction of Type A C_1V_2-reduplication for $C_1V_1C_2V_2$-initial bases. Second, it restricts the active *REPEAT constraint to *initial position*. This fixes HK's unrecognized prediction of Type D reduplicative infixation to repair non-base-initial repetitions, as well as $V_1C_2V_2$-reduplication for vowel-initial roots where $V_1 = V_2$ (which was recognized by HHK (pp. 26–27)). Third, through its use of indexed constraints,

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

it provides a convergent analysis of a small but, relatively speaking, sizable set of exceptions involving consonant-initial roots, which have not been formally analyzed in previous work. Lastly, it provides a more complete formal analysis of the stress system and its interaction with reduplication, which was only discussed in generalities by HK.

These changes allow us to understand the system as preferring the shortest possible reduplicant, subject to the needs of higher-ranked constraints. This is consistent with the atemplatic approach to reduplication (e.g. Spaelti 1997, Hendricks 1999, Riggle 2006, Zukoff 2016, 2017a), which HK seeks to implement. This is also consistent with the observed stress facts, such that reduplication-specific stress (or footing) requirements can induce exceptional stress while not having an effect on reduplicant shape.

Before moving on to our discussion of Morphological Doubling Theory, there is one other point to be made regarding CONTIGUITY. The individuated CONTIGUITY approach proposed here not only solves the Type B copying problem, it clarifies HK's "gradient" evaluation of CONTIGUITY. For example, in Type A forms, HK (p. 314) uses CONTIGUITY-BR to prefer copying V_2 [bi-beiha] (63b) rather than V_3 *[ba-beiha] (63c), under the assumption that the latter incurs greater CONTIGUITY violation.

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

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

However, if we adopted the traditional definition, repeated in (64) immediately below (cf. McCarthy & Prince 1995:123, HK:308), which treats contiguity violation as an all or nothing proposition, we actually shouldn't be able to distinguish between the two. On the other hand, the new definition proposed in this paper (cf. (17, 18)), which can be generalized over different units as in (65), spells out a method for categorical violation assignment over multiple loci.

(64) *Traditional definition of* (INPUT) CONTIGUITY: ((14) above)
Assign one violation * if the reduplicant doesn't correspond to a contiguous substring of the base.

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

With this in hand, for a Type A base, CONTIGC-BR will rule out $V_{n>2}$-copying (66c) because it skips any/all subsequent consonant(s). Even if we had an example with no subsequent consonants, e.g. a hypothetical root /beia/ (67), CONTIGV-BR would assign a fatal violation for skipping V_2 and any subsequent vowels (67c), because it counts up each locus of violation. This shows that this revised approach addresses multiple analytical questions within Tawala, as well as giving us traction on our theoretical understanding of CONTIGUITY.

(66) *Selecting the reduplicative vowel with individuated* CONTIGUITY

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

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

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

4 Overview of Morphological Doubling Theory

HHK argue that Tawala's reduplicant shape alternations constitute "base-dependence", in that the shape of the reduplicant cannot be determined based on its underlying representation alone, but rather only by knowing the (surface) phonological properties of the base and the reduplicant's position relative to it. IZ assert that their proposed theory of reduplication, Morphological Doubling Theory (MDT), could not generate such patterns. They argue that no base-dependent patterns, of this sort or any other, truly exist, and therefore that MDT is to be preferred over theories which predict such patterns, especially BRCT. HHK put forward an effort to analyze Tawala in MDT, and come to the conclusion that it cannot do so, as would be expected given IZ's assertions.

In the remainder of this paper, I show that there are certain additional technologies proposed by IZ in the context of MDT, not considered or explored fully by HHK, which may be able to generate a convergent MDT analysis of Tawala. The way to interpret the theoretical consquences of the analysis to be developed here is as follows. If one deems any of the technologies or analytical moves inappropriate, then HHK's conclusion is to be upheld: MDT does not have the power to generate Tawala's reduplicant shape alternations, because of their base-dependent nature. If one does not object to the analysis, then MDT *can* generate a pattern with all the hallmarks of base-dependence, voiding much of the content of IZ's claimed predictions in this regard. Furthermore, it lays bare the powerful mechanisms MDT has at its disposal, which should cast serious doubt on any claims about the restrictiveness of the theory.

This section provides an overview of MDT as a theory of reduplication, including its use of Sign-Base Morphology as its theory of morphology. This section also introduces one of the key additional technologies which will be brought to bear in the MDT analysis of Tawala, faithfulness-constraint-driven morphoprosodic parsing, by fleshing out IZ's analysis of reduplication in Javanese. Then, in Section 5, I lay out my analysis of Tawala. This analysis is based on the one developed by HHK, but diverges from it in a number of ways, mostly regarding the use of stress and morphoprosodic structure to govern what can and cannot be deleted at various stages.

4.1 The basic mechanics of Morphological Doubling Theory

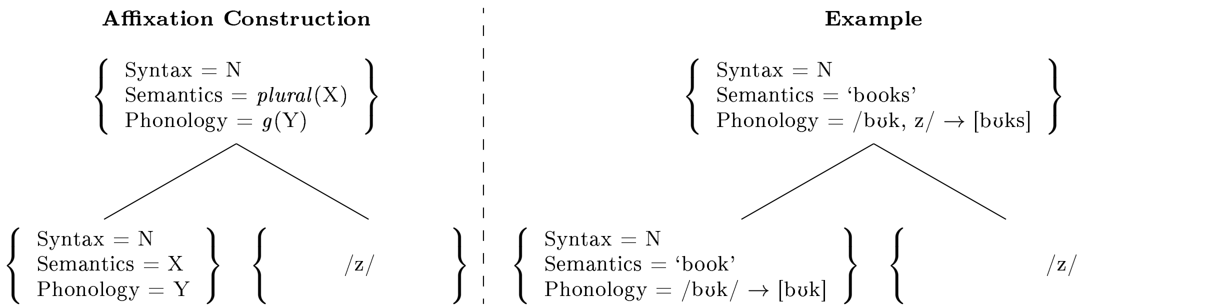
MDT is situated within the framework of Sign-Based Morphology (Orgun 1996, 1998, 1999, Orgun & Inkelas 2002),[11] a particular implementation of Construction Grammar. Words, morphemes, and other type of morphological constituents are instances of "constructions":

> "*In SBM constructions (and meta-constructions) are grammatical primitives, elaborated versions of phrase-structure rules which encode the semantic, syntactic, and phonological mappings between daughters and mothers*" (IZ:12).

The diagrams in (68) provide the constructional representation of English regular plural affixation. The construction is centered on the "mother node", the information in the upper set of curly braces in (68), which specifies the output syntax and semantics of the construction, plus the phonological grammar which will be applied to its inputs, indicated as the function *g(Y)* in the schematic representation on the left.

[11] IZ (p. 12) assert that the choice of SBM as the morphological framework is not crucial. Certain more recent work in cophonology theory, "Cophonologies by Phase" theory (Sande & Jenks 2018, Sande, Jenks, & Inkelas 2020), has been situated within Distributed Morphology (Halle & Marantz 1993). I consider only the SBM-based version of MDT developed by IZ, as, to my knowledge, no other alternative implementations of MDT have been proposed.

(68) *SBM representation of plural in English* (IZ:13)



The construction is also specified for its daughter nodes, represented as the respective sets of information in the lower curly braces in (68), attached to the mother node via branches in the mode of a syntactic or morphological tree. Put another way, the mother node of the construction subcategorizes for daughters with particular properties. In an affixation construction such as this, one of the daughter nodes is specified only for its phonological content — this is the "affix", which is /z/ in this example. The other node is specified for its own syntax and semantics, which essentially restricts the set of inputs to those which are morphosyntactically appropriate — in this case, a noun. It seems most appropriate to interpret the "Phonology" component of all the daughter nodes not just as underlying representations, but rather, like in the mother node, functions mapping inputs to outputs, i.e., a phonological grammar, as indicated in the example on the right in (68). While not terribly important in an affixation construction such as this, this property is crucial to the understanding of reduplication in MDT, as will be discussed immediately below. The outputs of the daughter nodes are fed upwards to the mother node as its input. The mother node then concatenates the two pieces and applies its phonology, generating the output of the construction. This construction might itself becomes an input to a higher-level construction, such that the mother node is recruited to serve as a daughter in the other construction.[12]

According to IZ, a "reduplicative" construction is one where the mother node subcategorizes for daughters that each have the same semantic specification. In this view, reduplication is essentially a compounding process where both daughters are specified as **{Semantics = 'Sem$_{\mathbf{x}}$'}**. They argue that tying reduplication to semantic identity rather than, say, phonological identity allows various similar processes, such as synonym compounding processes and (quasi-)reduplicative processes involving distinct allomorphs of the same morpheme (IZ:47ff.), to be brought under the same umbrella as reduplication. More importantly, using semantic rather than phonological identity allows them to ascribe *distinct phonological grammars* to each daughter, even while the input to the phonology is required to be identical.

These (possibly) distinct phonological grammars specific to particular nodes are referred to as *cophonologies* (Orgun 1996, Inkelas, Orgun, & Zoll 1997, Anttila 2002, 2009, Inkelas & Zoll 2007). In this Cophonology Theory approach to SBM, every single node has the capacity to have a unique phonological grammar, and thus a distinct input-output mapping for any given phonological input. The diagram in (69) provides a schematic reduplication construction with three distinct cophonologies, one for each node. The two daughter nodes, D_1 and D_2, each have the same semantic (and syntactic) properties, and they each select for the same phonological input, INPUT$_\alpha$. However, they have distinct cophonologies, Φ_i for D_1 and Φ_j for D_2. If the result of applying Φ_i to INPUT$_\alpha$ is distinct from the result of applying Φ_j to INPUT$_\alpha$, then the two components of the input to the mother node will be phonologically distinct. This is the way that MDT generates distinctions between the base and reduplicant: specifying relevantly distinct cophonologies for the two daughters of a "reduplicative" construction.

[12] As a consequence of IZ's notation, SBM/MDT derivations proceed from the bottom up, akin to a syntactic derivation, rather than from the top down, as in the traditional representation of a serial phonological derivation.

(69) *Reduplicative construction with distinct cophonologies*

Reduplicative Construction

$$\left\{\begin{array}{l}\text{Syntax} = \text{A}\\ \text{Semantics} = \text{B}\\ \textbf{Phonology} = \Phi_k(\Phi_i(\text{INPUT}_\alpha),\ \Phi_j(\text{INPUT}_\alpha))\end{array}\right\}_{\text{M}}$$

$$\left\{\begin{array}{l}\text{Syntax} = \text{X}\\ \text{Semantics} = \text{Y}\\ \textbf{Phonology} = \boxed{\Phi_i(\text{INPUT}_\alpha)}\end{array}\right\}_{\text{D}_1} \qquad \left\{\begin{array}{l}\text{Syntax} = \text{X}\\ \text{Semantics} = \text{Y}\\ \textbf{Phonology} = \boxed{\Phi_j(\text{INPUT}_\alpha)}\end{array}\right\}_{\text{D}_2}$$

Crucially, the phonologies of the respective daughter nodes operate in complete isolation from one another. All that they can see is their own phonological input. It is not until the outputs of the daughter cophonologies are entered into the input of the mother node cophonology — which itself can, of course, be completely distinct from that of the daughters' — that there can be any interaction between the two. And this interaction can only be governed by the mother node cophonology (Φ_k in (69)), which must apply equally to the two portions of the input. In other words, no special phonology can apply to either the "base" or the "reduplicant" (these terms have no formal status in MDT, but remain useful descriptors) once the two are placed in their output context, i.e., adjacent to their counterpart.

This is what IZ mean by "base-*in*dependence" (p. 92ff.; see also HHK): base and reduplicant cannot affect each other in any way not relating to the phonology of the mother node.[13] The non-MDT mechanism that most directly generates base-dependence is Base-Reduplicant correspondence, as proposed by McCarthy & Prince (1995) for BRCT. However, this is not the only means of generating base-dependence. Base-dependent reduplicant shape alternations, of the sort purportedly found in Tawala, can arise in BRCT and other similar frameworks because base and reduplicant are determined simultaneously in full view of the other. This is the crux of the analysis of Tawala presented in Section 3 above, as well as that of HK, which it is based on.

This makes it clear that accurately understanding the typological status of base-dependence in reduplication is essential for comparing between BRCT and MDT as theories of reduplication. If the BRCT analysis of Tawala is valid, then it must be stated that base-dependent reduplicant shape alternations are attested, contradicting IZ's claims. If, though, MDT can find a workable analysis of Tawala, then the assertion that MDT cannot derive base-dependent reduplicant shape alternations is falsified. This would therefore remove reduplicant shape alternations as a domain for comparing the two theories with respect to the concept of base-dependence. This latter state of affairs is the conclusion that will be advocated in this paper.

4.2 TETU in MDT

HHK determine that there is no convergent MDT analysis of the Tawala reduplicant shape alternations. The main grounds on which they make this determination is that MDT is incapable of capturing the emergence of the unmarked (TETU) in reduplication (McCarthy & Prince 1994, 1995, Alderete et al. 1999). This claim primarily regards HK's and HHK's treatment of Type D reduplication as infixation of the reduplicant to break up a root-initial sequence of identical syllables (see Section 3.5 above), which they correctly understand to be a TETU effect with respect to the constraint *Repeat (or, in my analysis, *Repeat(init)).

In McCarthy & Prince's "Basic Model" of reduplication, reduplicative TETU effects arise via the ranking schema Faith-IO ≫ Markedness ≫ Faith-BR (McCarthy & Prince 1995:81), under the crucial assumption that the reduplicant is not subject to Input-Output faithfulness. The ranking fragment Faith-IO ≫ Markedness means that the marked structure will be tolerated outside of the reduplicant, while the ranking fragment Markedness ≫ Faith-BR means that it will not be allowed to surface in the reduplicant. In other words, BRCT generates reduplicative TETU by setting up a faithfulness asymmetry between the

[13] Note that this may or not be equivalent to the "regular phonology" of the language, since the mother node of the reduplicative construction may itself have morpheme-specific phonology.

base and the reduplicant, in favor of the base (see also Beckman 1998 regarding positional faithfulness, and Stanton & Zukoff 2018, 2021 for TETU in copy epenthesis).

Since MDT rejects BR-correspondence, it is self-evident that this precise approach to TETU will not be available in MDT. However, this does not mean that TETU is not generable. For one, if the TETU effect does not depend on context contributed by the base, then it can be generated simply by placing the ranking MARKEDNESS ≫ FAITH-IO in the cophonology of the reduplicative daughter node (i.e. D_1 in (69)). This is not sufficient for the Tawala case, because the context is crucial, as the choice of reduplicant shape and position is conditioned entirely by the phonological properties of the base. But there is at least one other way of replicating the faithfulness asymmetry of BRCT in MDT such that we can generate TETU effects in the mother node cophonology in a manner exactly parallel to the IO vs. BR distinction: faithfulness to (morpho)prosodic constituents.

4.3 Prosodic constituents and truncation in MDT

One of the tools that IZ make extensive use of in their various analyses is the notion of *prosodic constituents*. These are phonological representations that are associated with, but distinct from, equivalent morphological constituents (IZ:140; following Booij 1985, Nespor & Vogel 1986, Sproat 1986, Inkelas 1990, Booij & Lieber 1993; see Cole 1994, Downing 1998a,b, *a.o.*, on reduplication). I will henceforth refer to them as "morpho-prosodic" constituents, so as to distinguish them from purely prosodic representations like syllables and feet. IZ's set of morphoprosodic constituents are given in (70):

(70) *Morphoprosodic constituents*

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

IZ (Ch. 5.1, esp. pp. 140–141) introduce the formalism in their analysis of Javanese (Java, Malayo-Polynesian, Austronesian; 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 (71):

(71) *Javanese reduplicated causatives* (IZ:139)[14]
ROOT *uni* 'sound' → CAUSATIVE *ŋ-une-ʔake* → REDUPLICATED *une**ʔ**-une-ʔake* (**une-une-ʔake*)

Their analysis is as follows.[15] The reduplicative construction categorizes not for roots but for stems in its daughter nodes. These stems can, and, in cases like (71), must be causative stems, which are themselves morphologically complex. Because of the semantic identity condition on reduplicative constructions, both daughters must be causative stems, and thus they both must contain the causative suffix in the input to their phonology. (See (78) below for a visual representation of this structure.)

The phonology of the causative stem construction, which feeds both daughter nodes, builds a PRoot (indicated by {...}) that includes all root segments *and all adjacent non-root segments that don't add a syllable* (i.e. adjacent consonants). IZ spell out their method for PRoot-parsing (p. 141). They employ the faithfulness constraints in (72), which introduce Roots (R) and PRoots (P) into the set of constituents which can be related by faithfulness, ranked as in (73).[16] The constraint MAX_{SEG}-IP (72e) makes clear that the Input as a whole can also be related to the output's PRoot constituent via faithfulness. Segments which are not parsed into PRoots are nonetheless retained in the output of this node. This will be crucial for deriving the correct ultimate surface form of the reduplicated word.

[14] IZ suggest that the [ʔ] of the suffix is an independent morpheme, but this does not factor into their analysis.
[15] IZ's full analysis covers a great deal more of the complexity involved in this case than is being presented here, omitted for reasons of space. What follows is a faithful presentation of this portion of IZ's analysis, using their chosen example, which will suffice to illustrate their conception of faithfulness to morphoprosodic constituents.
[16] Note that their use of syllable-oriented faithfulness constraints presupposes that the root is parsed into syllables prior to this mapping. This is highly relevant for assessing HHK's claims regarding prosodic constituent-copying reduplication patterns; see Section 6.

(72) a. **Max$_{\text{seg}}$-RP:** All Root segments should have correspondents in the PRoot.
b. **Max$_{\text{syl}}$-RP:** All Root syllables should have correspondents in the PRoot.
c. **Dep$_{\text{seg}}$-RP:** All PRoot segments should have correspondents in the Root.
d. **Dep$_{\text{syl}}$-RP:** All PRoot syllables should have correspondents in the Root.
e. **Max$_{\text{seg}}$-IP:** All Input segments should have correspondents in the PRoot.

(73) *PRoot parsing at the stem node*[17]

/uni-ʔake/	Dep$_{\text{syl}}$-RP	Max$_{\text{seg}}$-IP	Dep$_{\text{seg}}$-RP
a. {une}ʔake		4!	
b. {uneʔake}	2!		4
c. ☞ {uneʔ}ake		3	1
d. {u}neʔake		6!	

The output of the causative stem, [{uneʔ}ake], is entered into the input of both D1 and D2. According to IZ's analysis, D1 then deletes all segments which are not contained in the PRoot, while D2 simply preserves all segments faithfully. IZ are less explicit about how to do the PRoot-conditioned deletion in D1, but we can extrapolate that there are similar Max/Dep constraints for segments referencing the PRoot-to-Output relation:

(74) **Max$_{\text{seg}}$-PO:** Assign one violation * for each segment contained within a PRoot in the input which lacks an output correspondent.

In D1, this Max constraint must outrank a constraint motivating truncation, e.g. *Struc[ture] (75). This constraint in turn outranks the general Max$_{\text{seg}}$-IO constraint, motivating deletion of everything outside the PRoot ((76b) ≻ (76a)) but nothing inside the PRoot ((76b) ≻ (76c)).

(75) ***Struc**: Assign one violation * for each segment in the output.

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

/{uneʔ}ake/	Max$_{\text{seg}}$-PO	*Struc	Max$_{\text{seg}}$-IO
a. {uneʔ}ake		7!	
b. ☞ {uneʔ}		4	***
c. {une}	*!	3	****

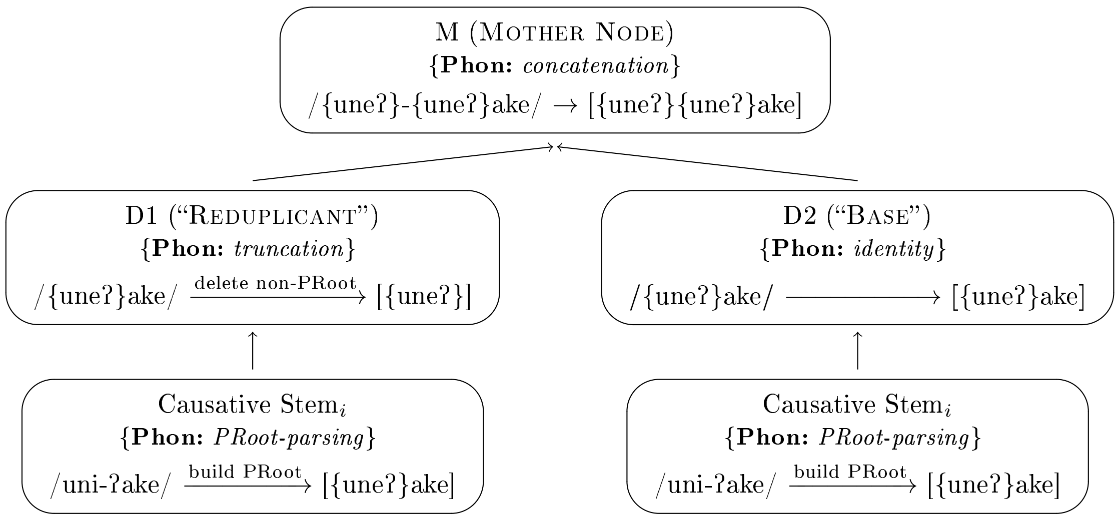
The interaction in (76) is a sort of TETU, or at the very least recapitulates its mechanics. Within a single representation, there is a faithfulness constraint, Max-PO, that privileges segments belonging to a particular constituent, the PRoot. This faithful constraint outranks a markedness constraint, *Struc, such that the markedness constraint has no effect on the segments belonging to that constituent. Yet, the markedness constraint outranks the equivalent faithfulness constraint protecting all other segments, Max-IO, triggering markedness reduction effects on those segments, i.e., deletion in this case. In (77), we see that this ranking schema perfectly mirrors the ranking involving BR-faithfulness that derives reduplicative TETU in BRCT. This demonstrates that, by including morphoprosodic constituents in the analysis, along with faithfulness constraints that can refer to them, MDT allows for TETU to occur within a single mapping, not just across distinct mappings.

(77) *TETU rankings*
a. Faith-IO ≫ Markedness ≫ Faith-BR (reduplicative TETU in BRCT)
b. Faith-PO ≫ Markedness ≫ Faith-IO (morphoprosodic TETU in MDT)

[17] The easiest way to interpret the Max/Dep$_{\text{syl}}$ constraints is as referring to [+syllabic] segments, i.e. vowels. IZ do not indicate syllable boundaries in these derivations, so I am not sure whether they intend for the [ʔ] in winning candidate (73c) to be parsed as a coda, thus having syllable and PRoot boundaries coinciding, or as an onset, leading to a mismatch. If it is syllabified as a coda, then the reflex flowing through D2 will (presumably) need to be resyllabified as an onset later in the derivation. This same question will arise again in the analysis of Tawala in Section 5.4.

To complete the analysis of Javanese, as long as (76b) is the output of D1, and the output of D2 is a faithful realization of the stem (i.e. (73c)), then we derive the desired result. The full analysis is summarized in (78):

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



IZ use these sorts of morphoprosodic 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. Faithfulness to morphoprosodic constituency will bring us closer to a convergent analysis for Tawala, solving some of the issues raised by HHK.

5 Analyzing Tawala in Morphological Doubling Theory

Having introduced the core architecture of MDT, we can now proceed to the MDT analysis of Tawala's reduplicant shape alternations. This analysis takes as its starting point the one developed by HHK (pp. 14–17, 21–26). They argue that their analysis, and indeed any possible MDT analysis, cannot derive Type D infixation in the context of the other, more canonical copying patterns. Since reduplicant shape cannot be determined in a dynamic, base-dependent manner in MDT, partial reduplicants must be derived through deletion. The problem with Type D is that deletion all the way down to the observed size seems impossible to motivate. In this section, I will show that faithfulness to carefully-assigned stress and morphoprosodic structure, with just the right kind of forethought, can generate exactly the necessary patterns of deletion. What turns out to be a potentially more serious problem is the treatment of vowel-vowel sequences in Type A forms. Following a suggestion by HHK (p. 16), I show that we can solve this by positing an additional, *ad hoc* constructional node, which will allow us to opaquely repair the problem. However, this analytical move may be inconsistent with the understanding of allowable constructions in SBM.

If the technology and analytical strategies employed all pass muster, then we will be forced to conclude that MDT *can* analyze Tawala's putatively base-dependent reduplicant shape alternations, *contra* HHK. If, on the other hand, they do not pass muster, then this reaffirms HHK's conclusion that MDT cannot. The contours of this question will be explored more thoroughly in Section 6.

5.1 Data review and analysis preview

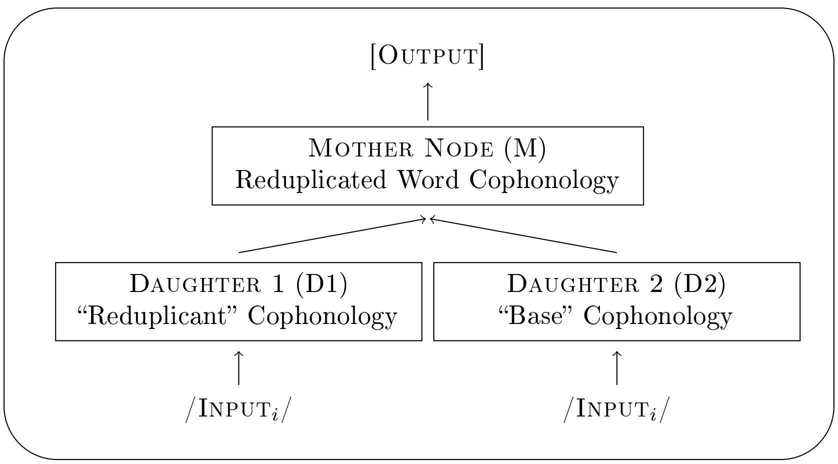
The table in (79) repeats the basic data patterns introduced above, now including stress markings, which will be significant for the MDT analysis. As discussed in Section 3.2, primary stress generally falls on the penult (79b,c,d). The exception is when the penultimate vowel is in hiatus with the antepenultimate vowel, and the antepenultimate vowel is lower in height. In that case (79a), primary stress falls on the antepenult. The first syllable of the reduplicant always bears a secondary stress, even if this results in a clash, as it does in (79a).

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

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

To review, the diagram in (80) presents a trimmed down schematic derivation of reduplication in MDT, showing just the relevant pieces of phonology and morphology. It shows what this would look like for a "prefixal" partial reduplication pattern, i.e. one where truncation takes place in the lefthand daughter (which we will continue to call D_1), such as the one in Tawala. This architecture will serve as the basis for the analysis in this section.

(80) *The morphophonological derivation of reduplication in MDT*

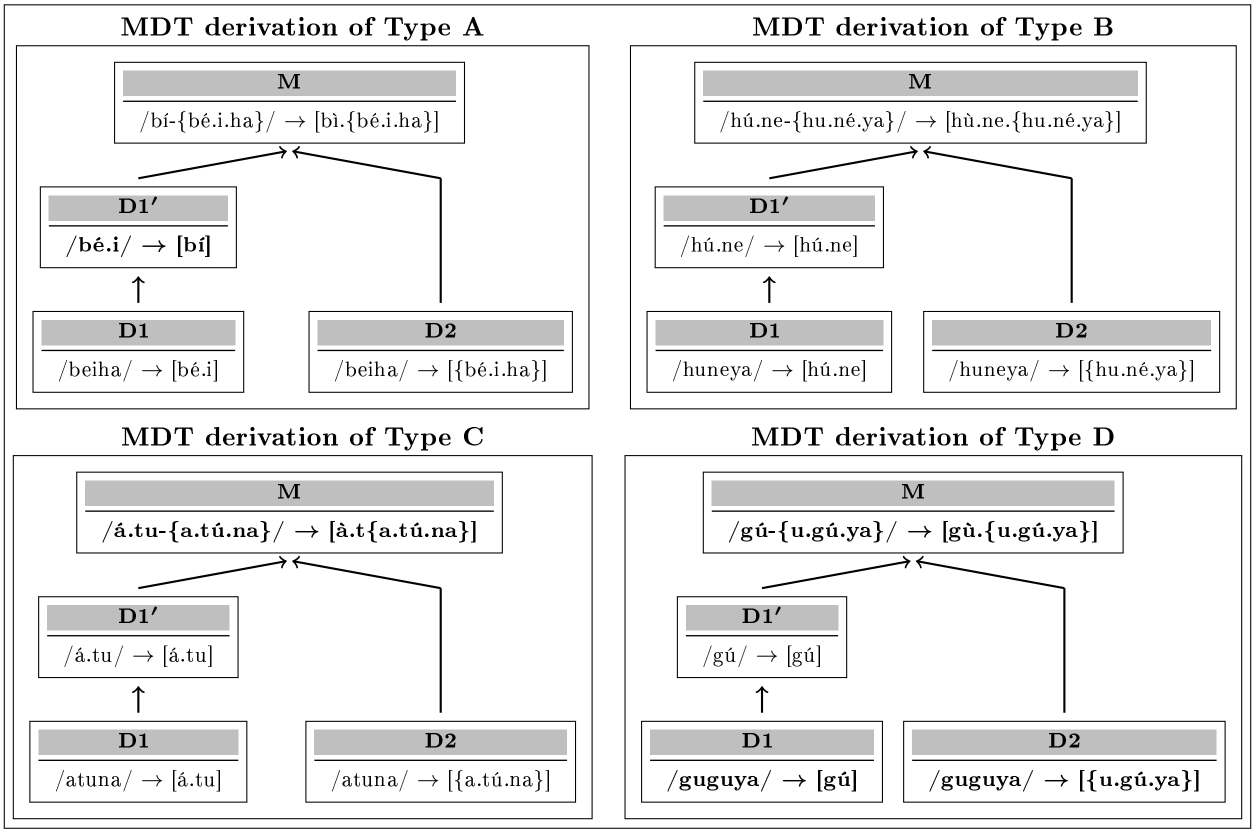


The analysis will work as follows. The "reduplicant" cophonology in D1 preferentially truncates the input down to two syllables. This is driven primarily by stress considerations, which involve placing stress on the leftmost vowel. Extra deletion occurs for Type D — truncation down to a single syllable, e.g. /guguya/ → [gú] — driven by its underlying repeated identical root-initial syllables. These interactions are detailed in Section 5.2. The "base" cophonology in D2 faithfully retains all underlying material in the general case, and parses that material into a PRoot. The exception again comes from Type D forms, where the identical syllables are resolved by a different type of deletion, namely, root-initial-consonant deletion: e.g. /guguya/ → [{u.gú.ya}] (where curly braces indicated PRoot boundaries). This is shown in Section 5.3.

As demonstrated in Section 5.4, these particular deletion patterns for Type D roots in D1 and D2 lead us directly to the attested Type D pattern: /gú/ + /{u.gú.ya}/ → [gù.{u.gú.ya}] (79d). The reduplicated word cophonology in M concatenates these outputs as such, successfully deriving the Type D pattern which HHK asserted could not be derived in MDT. Cophonology M also resolves hiatus through V_1-deletion, correctly deriving Type C forms. V_1-deletion is blocked in Type D forms by faithfulness to stressed vowels and vowels belonging to PRoots, which is the primary motivation for morphoprosodic parsing in D2.

This analysis of hiatus resolution causes a problem for Type A forms, where deletion in M would be predicted to behave counter to fact. This can be fixed, following HHK (p. 16), by positing an extra node between D1 and M — I refer to it as D1′ — which resolves hiatus in a different manner before the "reduplicant" reaches the mother node. This is explained in Section 5.5. The full derivations for the four different types are summarized in (81), where bolded mappings indicate important differences between types.

(81) *Preview of MDT derivations*



5.2 D1 cophonology: truncation

We will first consider the phonology of the "reduplicant", i.e. D1. Following IZ (p. 95) and HHK, D1 preferentially truncates the input down to two syllables (82a,b,c).

(82) *Input-Output Mapping in D1*

		Input		Output
a.	**Type A:**	/beiha/	→	[bé.i]
b.	**Type B:**	/huneya/	→	[hú.ne]
c.	**Type C:**	/atuna/	→	[á.tu]
d.	**Type D:**	/guguya/	→	[gú]

HHK take a Generalized Template Theory (McCarthy & Prince 1995, Urbanczyk 1996, et seq.) style approach, positing a high-ranked constraint requiring that the output coincide with a foot. Since constraint interaction will be crucial to the present analysis (and I argued above in Section 3.2 that feet are not involved in the pattern), I instead use foot-free stress constraints to generate the disyllabic reduplicant. The relevant constraints are defined in (83). These constraints both outrank *Struc, which functions as a size restrictor constraint, which itself outranks Max-IO.

(83) *Stress constraints for Tawala MDT analysis*

a. **StressL:** Assign one violation * if the leftmost syllable is unstressed. (*[#$\breve{\sigma}$])
b. **NonFinality:** Assign one violation * if the rightmost syllable is stressed. (*[$\acute{\sigma}$#])

(84) *Additional constraints*

a. ***Struc:** Assign one violation * for each segment in the output.
b. **Max-IO:** Assign one violation * for each input segment w/o an output correspondent.

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

The tableau in (86) shows the effect of this ranking for a Type B form. The combined effect of StressL and NonFinality requires at least two syllables in the truncatum, because any candidate composed of a single syllable would have to violate one of the two constraints, since that syllable must either be stressed or unstressed. This also correctly places stress on the first syllable. It will do no harm to assume that the output of this node constitutes a Prosodic Word, such that the stress features are not assigned independently of that larger prosodic structure. However, it is crucial that this material is not parsed into a Prosodic *Root* (PRoot), which is a distinct morphoprosodic constituent. The reasons for this will be made clear below.

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

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

We begin to substantively diverge from HHK's when we come to Type D forms, which are the crucial case in HHK's argument. The current analysis proposes that these roots are truncated to a *single* syllable, rather than two syllables. The constraint *Repeat(init) (repeated in (87)) can generate this behavior, by penalizing a faithful realization of the first two syllables. If *Repeat(init) and StressL rank above NonFin, the evaluation will select the stressed one-syllable truncation candidate (89c).

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

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

(89) *D1 derivation of Type D*

/gu$_1$gu$_2$ya/				StressL	*Repeat(init)	NonFin
a.		gú$_1$.gu$_2$.ya	[100]		*!	
b.		gú$_1$.gu$_2$	[10]		*!	
c.	☞	gú$_1$	[1]			*
d.		gu$_1$	[0]	*!		

Two other faithfulness constraints, Anchor-L-IO (90a) and Contiguity-IO (90b), are necessary to rule out other viable 2σ alternatives, which satisfy *Repeat(init) and NonFin. As long as these constraints outrank NonFin, all such candidates will be eliminated, as shown in (91).

(90) a. **Anchor-L-IO:** Assign one violation * if the leftmost segment in the input does not correspond to the leftmost segment in the output.
b. **Contiguity-IO:** Assign one violation * for each pair of adjacent segments in the output which were not adjacent in the input.[18]

(91) *D1 derivation of Type D: alternative disyllabic candidates*

/gu$_1$gu$_2$ya/				Anchor-L-IO	Contig-IO	NonFin
a.	☞	gú$_1$	[1]			*
b.		gú$_1$.ga	[10]		*!	
c.		gú$_1$.ya	[10]		*!	
d.		ú$_1$.gu$_2$	[10]	*!		
e.		gú$_2$.ya	[10]	*!		

Given that hiatus will be resolved in other nodes, it is worth mentioning here that Onset is inactive in this cophonology. This can be seen in Types A and C, where onsetless syllables surface in word-medial position and word-initial position, respectively. We can minimally determine that Onset ranks below NonFin by

[18] This analysis does not require the individuated Contiguity constraints argued for in the BRCT analysis in Section 3, so I retain the more traditional definition here.

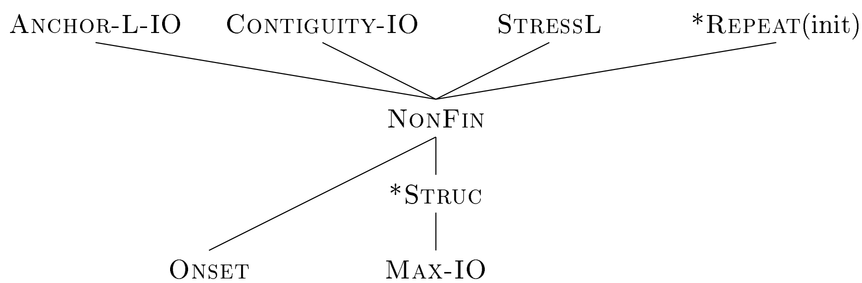
looking at an underlyingly disyllabic Type A root like /tou/ 'weep' (→ [tù-tó.u]) (cf. (5)). Deletion of the final vowel (92b) would alleviate the hiatus without deleting an initial vowel (which would violate Anchor-L-IO) or a medial vowel (which would violate Contig-IO). However, this would lead to a stressed monosyllable, akin to Type D, violating NonFin. Since the necessary output is the disyllable with hiatus (92b), we know that NonFin ≫ Onset.

(92) *No hiatus resolution in D1*

/tou/				Contig-IO	NonFin	*Struc	Onset
a.	☞	tó.u	[10]			*	*
b.		tó	[1]		*!		
c.		tú	[1]	*!	*		

Candidate (92c) is one which deletes the medial vowel to resolve the hiatus. Under this ranking, this candidate is additionally penalized by high-ranked Contig-IO. Note that this output represents the final shape of the "reduplicant" at the end of the derivation for Type A roots: [tù-tó.u], [bì-bé.i.ha], etc. The tableau in (92) makes it clear that this candidate cannot be derived in this cophonology alongside the other necessary mappings (see also HHK:15–16). Therefore, it will instead need to be derived later in the derivation. This will turn out to be the stickiest problem for this analysis. The rankings for D1 are summarized in (93):

(93) *Ranking summary for D1*



5.3 D2 cophonology: PRoot parsing and Type D C_1-deletion

The cophonology of D2 largely resembles the regular phonology of the language, which is to be expected in a prefixal partial reduplication pattern. The main thing that happens in this node is that the grammar builds a Prosodic Root (PRoot; cf. IZ:140; Downing 1998a,b; see Section 4.3 above) in the output. By having a high-ranked Max$_{\text{SEG}}$-Input-PRoot [Max-IP] constraint (94) (cf. (72e)), we motivate parsing all underlying segments into that PRoot, as shown in (95). In order for faithfulness to help derive the right outputs in the mother node (see Section 5.4) , it is crucial that D1 not build an output PRoot. Therefore, D1 needs to have a constraint against PRoots in the output — perhaps a markedness constraint like *PRoot — which outranks Max-IP. D2 needs to have the reverse ranking, where Max-IP outranks this constraint.

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

(95) Max-IP *induces PRoot parsing, e.g. for Type A*

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

In order to provide the mother node with the inputs it will need to generate the right result for Type D, D2 needs to apply initial consonant deletion just for Type D. As in D1, this deletion can again be effectuated using high-ranked *Repeat(init). If this constraint ranks above Max-IP *and* Max-IO (96), the underlying repeated identical syllables of Type D roots triggers initial-C deletion for Type D only, as shown in (97).

Since there is not general truncation in D2, we know that *Struc ranks below at least one of the Max constraints.

(96) **Ranking for initial-C deletion in Type D:** *Repeat(init) ≫ Max-IP, Max-IO

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

/g$_1$u$_1$g$_2$u$_2$ya/		*Repeat(init)	Max-IP	Max-IO	*Struc
a.	{g$_1$u$_1$.g$_2$ú$_2$.ya}	*!			6
b.	g$_1${u$_1$.g$_2$ú$_2$.ya}	*!	*		6
c. ☞	{u$_1$.g$_2$ú$_2$.ya}		*	*	5
d.	{g$_2$ú$_2$.ya}		**!	**!	4

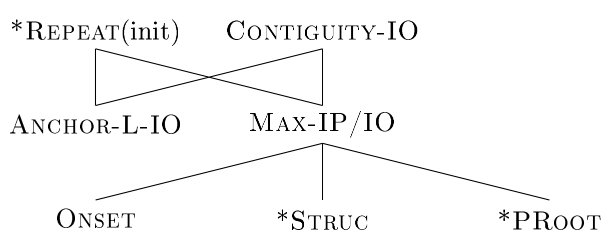
The fact that the *Repeat(init) violation is repaired by *initial* deletion (98b) and not medial deletion (98d,e) shows that Contiguity-IO ≫ Anchor-L-IO (98) in this cophonology. At least the higher-ranked Max constraint must also outrank Onset, because initial-C deletion (98b) is preferred to initial-CV deletion (98c). The low ranking of Onset (as well as the high ranking of Contiguity-IO) in D2 is consistent with the fact that hiatus is tolerated stem-internally in D2 (e.g. /beiha/ → {bé.i.ha}).

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

/g$_1$u$_1$g$_2$u$_2$ya/		*Repeat(init)	Cntg-IO	Anchor-L-IO	Max-IP/IO	Onset
a.	{g$_1$u$_1$.g$_2$ú$_2$.ya}	*!				
b. ☞	{u$_1$.g$_2$ú$_2$.ya}			*	*	*
c.	{g$_2$ú$_2$.ya}			*	**!	
d.	{g$_1$ú$_1$.ya}		*!		**	
e.	{g$_1$ú$_2$.ya}		*!		**	

The rankings proposed for D2 are summarized in (99). I have been representing the outputs of D2 also as having undergone the normal stress pattern (see Ezard 1997): right-to-left trochees, i.e., penultimate primary stress + alternating stress leftward from the penult. It is not crucial that stress be applied at D2 rather than just being applied all at once in M. If it is applied in D2, then the stress grammar spelled out in Section 3.2 above (summarized in (36)), with the constraints Align-Root-L and StressL-Red excised, will suffice. Note that, unlike in D1, the constraint StressL would be inactive/low-ranked in the D2 cophonology.

(99) *Ranking summary for D2*



5.4 M cophonology: restricted hiatus resolution

The only process in M is deletion to repair hiatus that occurs at the morpheme juncture for Type C (103). This process can be modeled using the ranking Onset ≫ Max-IO (102), as Onset penalizes the faithful hiatus candidate (103a). We can ensure that it is the first vowel that is deleted (103b) rather than the second (103c) by appealing to Max$_{seg}$-PRoot-Output [Max-PO] (100) (cf. (74)), which penalizes deletion of segments belonging to a PRoot in the input. This constraint will protect the second vowel in such cases because it belongs to the PRoot built in D2. On the other hand, in this case, the "reduplicant"-final vowel is *not* protected by any such special faithfulness constraint, because D1 crucially has not built a PRoot.[19]

[19] Note the mismatch in winning candidate (103c) between syllable boundaries and PRoot boundaries, of the same sort as encountered in our analysis of Javanese (see fn. 17 above). If there is a constraint requiring coincidence of these boundaries, or an equivalent restriction on gen, then we could use the PRoot parsing faithfulness constraints to repair the mismatch.

(100) **Max-PO:** Assign one violation * for each vowel segment which was part of a PRoot in the input that does not have an output correspondent.

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

(102) **Ranking for hiatus resolution in M:** Max$_{\acute{V}}$-IO, Max-PO ≫ Onset ≫ Max-IO[20,21]

(103) Onset-*driven reduplicant-*V_2 *deletion in M for Type C*

/á.tu$_1$-{a$_2$.tú.na}/			Max$_{\acute{V}}$-IO	Max-PO	Onset	Max-IO
a.		à.tu$_1$.{a$_2$.tú.na}			**!	
b.	☞	à.t{a$_2$.tú.na}			*	*
c.		à.tu$_1$.{tú.na}		*!	*	*
d.		t{a$_2$.tú.na}	*!			**

The ranking Onset ≫ Max-IO would predict also deletion of reduplicant-initial vowels in Type C, namely candidate (103d). One way to avoid this outcome would be to include Anchor-L-IO ≫ Onset in the rankings. Alternatively, we could use Max$_{\acute{V}}$-IO (101), a constraint protecting underlyingly stressed vowels, since the initial vowel of the "reduplicant" will always be stressed, given the activity of StressL in D1. The reason to use this constraint is that it helps explain why hiatus is left unresolved in Type D (104).

Employing both Max$_{\acute{V}}$-IO and Max-PO in our analysis of hiatus means that, when we get to Type D, we find that both of the hiatal vowels are protected by special high-ranking faithfulness constraints. That is, deletion of the first vowel (104c) is blocked by Max$_{\acute{V}}$-IO because it is stressed, and deletion of the second vowel (104b) is blocked by Max-PO because it is underlyingly in a PRoot. For this reason, hiatus must be tolerated (104a) in Type D. Max-PO also explains why there continues to be no hiatus-driven deletion "base"-internally in, e.g., Type A (105).

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

/gú$_1$-{u$_2$.gú.ya}/			Max$_{\acute{V}}$-IO	Max-PO	Onset	Max-IO
a.	☞	gù$_1$.{u$_2$.gú.ya}			*	
b.		gù$_1$.{gú.ya}		*!		*
c.		g{u$_2$.gú.ya}	*!			*

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

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

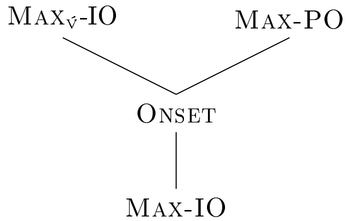
The one other constraint worth mentioning here is *Clash. As can be seen from a form like [bì.{bé.i.ha}], clash is tolerated across the juncture in M, as discussed in the context of the BRCT analysis in Section 3.2. Therefore, cophonology M must have a relatively low-ranked *Clash. A higher-ranked Max[Stress]-IO constraint would yield retention of this stress, as would a higher-ranked StressL coupled with another constraint preferring stress retention on PRoot segments. Since there are multiple analyses available, and this is orthogonal to the main point, I will not formalize these rankings. Cophonology M must also be responsible for demoting the "reduplicant's" underlying primary stress to secondary stress. I will not formalize this interaction either. The rankings proposed for M are summarized in (106).

This would have no effect on what does or does not get deleted, because all that Max-PO cares about is whether a segment was *underlyingly* contained within a PRoot.

[20] Strictly speaking, the interaction in (103) does not fix the ranking of Max-PO with respect to any of the other constraints. However, the interactions in (104) and (105) prove that Max-PO ≫ Onset, as shown here.

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

(106) *Ranking summary for M*



5.5 Cophonology D1′: an extra node for opaque hiatus-resolution in Type A

While we have now successfully generated the attested surface form of Type D roots, there remains one very substantial problem involving Type A, which was identified already by HHK (pp. 15–16). The tableau in (107) shows the result of the current grammar for a Type A root in D1. Just like Type B roots (cf. (86)), it selects a disyllabic output consisting of the first two syllables of the root.

(107) *D1 derivation of Type A*

/beiha/				CONTIGUITY-IO	NONFIN	*STRUC	MAX-IO
a.		bé.i.ha	[100]			5!	
b.	☞	bé.i	[10]			3	2
c.		bé	[1]		*!	2	3
d.		bí	[1]	*!	*	2	3
e.		bé.ha	[10]	*!		4	1
f.		bí.ha	[10]	*!		4	1

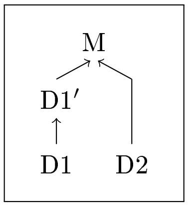
However, this is not the input that was entered into M in the Type A derivation in (105) above, which was rather /bí/, the equivalent of losing candidate (107d). If we entered /bé.i/, the output of D1 for Type A, into the input for M, we wrongly predict that the unstressed /i/ should delete under hiatus in M (just like in Type C above), yielding *[bè.{bé.i.ha}] (108b). If deletion of the /i/ were blocked by the fact that this would introduce an initial repetition (a *REPEAT(init) violation), the alternative would be *no deletion* [bè.i.{bé.i.ha}] (108a), not deletion of stressed /é/ (108c,d). If something could prefer /é/ deletion, We would also need to add STRESSL to this cophonology in order place a new stress on the /i/, in order to overcome (108c). This would not have any deleterious effects, because all reduplicated words do have an initial stressed syllable.

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

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

The formally simplest solution, as suggested by HHK (p. 16), is to introduce an additional node — which I will call D1′ — between D1 and M, as shown in (109). This additional node can be given its own cophonology. If we can adduce a constraint ranking which can map /bé.i/ → [bí], we will fix the problem. Such a ranking is provided in (111).

(109) *MDT derivational structure with D1′*



The way to do this is as follows. Onset must outrank Max$_{\acute{V}}$-IO and Max-IO (110a) to make sure stressed vowel deletion will be tolerated as a means of avoiding hiatus. Onset must also outrank Contiguity-IO to allow for the discontiguous mapping resulting from V_1-deletion, and outrank Dep[stress]-IO to allow for restressing of the /i/ (110a). Anchor-R-IO[22] must outrank Max$_{\acute{V}}$-IO and Contiguity-IO (110b) to prefer keeping the rightmost vowel instead of the stressed one, which is also the one that's adjacent to the stem-initial consonant. Lastly, StressL must outrank Dep[stress]-IO (110c) to favor inserting stress onto the newly leftmost /i/, and also outrank NonFin (110c) (just as in D1) because this means that the final vowel will now be stressed.

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

a. Onset ≫ Max$_{\acute{V}}$-IO, Max-IO, Contiguity-IO, Dep[stress]-IO [(111d) ≻ (111a)]
b. Anchor-R-IO ≫ Max$_{\acute{V}}$-IO, Contiguity-IO [(111d) ≻ (111b)]
c. StressL ≫ Dep[stress]-IO, NonFin [(111d) ≻ (111c)]

(111) *Fixing Type A in D1′*

/bé.i/	Ons	Anch-R-IO	StrsL	MaxV́-IO	Cntg-IO	Dep[strs]-IO	NonFin
a. bé.i	*!						
b. bé		*!					*
c. bi			*!	*	*		
d. ☞ bí				*	*	*	*

We'll also need Anchor-L-IO to outrank Onset (112) in order to avoid deleting the stem-initial vowel in Type C (113).

(112) **Ranking for Type C in D1′:** Anchor-L-IO ≫ Onset

(113) *Not messing up Type C in D1′*

/á.tu/	Anchor-L-IO	Onset	Anchor-R-IO	Max$_{\acute{V}}$-IO	Contiguity-IO
a. ☞ á.tu		*			
b. tú	*!			*	

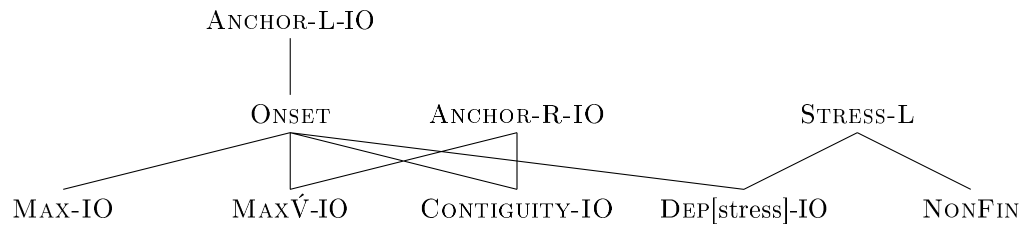
Once we implement these rankings in D1′, we achieve the desired result for Type A in M (114); in fact, there are no obvious competitor candidates. The rankings proposed for D1′ are summarized in (115).

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

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

[22] We could equivalently do this with a constraint like Max-V_2 ("*Don't delete a vowel which is underlyingly post-vocalic*"), which is the approach suggested by HHK (p. 16). Since such a constraint is otherwise unnecessary in the present analysis, whereas the mirror image Anchor-L constraint is necessary, I choose to use Anchor-R.

(115) *Ranking summary for D1′*



The problem with this approach comes down to whether the theory allows for a node like D1′ in the first place. MDT is couched within Sign-Based Morphology (Orgun 1996 et seq.; see Section 4.1). In this theory, "constructions" are composed of a mother node that selects for (up to) *two* daughter nodes. There does not seem to be any dispensation for a construction to subcategorize for a node dependent on one of the daughter nodes. Furthermore, for the construction to be reliably reduplicative, the semantic identity subcategorization typically holds of the two immediate daughter nodes, which themselves would be sisters. This identity condition could in theory be imposed between D1′ and D2, which would be the true daughter nodes in this account, but then it is not clear that D1′ would be required to take as its input the output of D1, as opposed to the unmodified root. Fully investigating these conceptual and theoretical questions would take us too far afield, so they will not be pursued further here. However, it suffices to say that using the D1′ solution would seem to expand the space of possible constructions, thus making the theory less restrictive. Yet, denying the use of such an analytical move would seem to foreclose on any convergent phonological analysis.[23]

5.6 Local discussion

This section has demonstrated that an MDT analysis of Tawala's reduplicant shape alternations is possible, but only by enriching HHK's analysis with several pieces of powerful machinery. First, it requires the use of morphoprosodic structure, and, more significantly, a highly articulated set of faithfulness constraints which can make reference to the resulting constituency. This allows MDT to capture cases of the emergence of the unmarked (TETU) in reduplication by reintroducing the faithfulness asymmetry represented in BRCT by the IO vs. BR distinction. This would seem to be a welcome result, as reduplicative TETU effects are well-documented. However, the full predictions of this articulated view of morphoprosodic faithfulness are yet to be spelled out, and it is reasonable to be concerned that there may exist ways of using these constraints to reintroduce other types of patterns which were putatively not predicted by MDT.

Second, this analysis requires the inclusion of an *ad hoc* additional node, containing a unique and highly specified cophonology. As discussed immediately above, it is not at all clear whether this is a legal move within Sign-Based Morphology, since it stipulates a node which is dependent on one of the daughters. Even putting that question aside, we can immediately recognize the expressive power of such a node. It can be placed anywhere in the analysis to fix any conceivable problem with the phonology, as long as the mapping is expressible through typical constraint interaction. We could even countenance multiple such nodes in sequence, allowing for multiple transformations which could not be derived in parallel. Such an approach would allow for the introduction of nearly unchecked opacity in a system which is already well-equipped to derive opaque interactions.

Lastly, it is worth scrutinizing how this analysis derives Type D forms. It does so by positing some deletion in D1 and some deletion in D2, such that combining the two constituents yields the highly reduced form observed in the output. These two distinct deletion operations are each observed only for Type D roots (though this is expressly motivated by a constraint which must be present in any type of analysis, namely, *REPEAT(init)) and only in this particular morphological category. It seems reasonable to worry about whether such a pattern is really learnable.

[23] It is worth noting that the problematic candidates in (108) are exactly the two other attested copying patterns for CVV roots, as detailed in Section 3.3. Therefore, if we treated one of these other patterns as the "default", this problem might disappear. Exceptional rankings can be built directly into cophonologies tied to particular roots in order to handle whatever remains as "exceptions". Working this out in detail is deferred to future work.

6 Conclusion

6.1 Discussion

This paper has investigated putatively base-dependent reduplicant shape alternations in Tawala. Such base-dependent patterns, including this one in particular, have been identified as ones which may be generable in some theories of reduplication, such as Base-Reduplicant Correspondence Theory (BRCT; McCarthy & Prince 1995), but not generable in certain other theories, namely, Morphological Doubling Theory (MDT; Inkelas & Zoll 2005 [IZ]). This paper has offered revised analyses of this pattern in both BRCT and MDT. The BRCT analysis proposed here improves upon Hicks Kennard's (2004) [HK] in that it fixes several unrecognized mispredictions, and also formalizes the analysis of stress and its relation to reduplication, as well as formalizing the analysis of exceptional copying patterns. The MDT analysis proposed here is the first which may be able to account for the set of alternations in full, potentially contradicting Haugen & Hicks Kennard's (2011) [HHK] conclusion that no such analysis is possible. However, the full suite of technologies and analytical strategies required in order to reach this convergent analysis raises a whole host of questions about the actual restrictiveness of MDT more generally, as they greatly increase the theory's power to opaquely capture complex patterns.

So what are we to conclude about these two competing theories of reduplication? HHK's argument against MDT was one of *undergeneration*: the pattern exists, MDT cannot generate it, and therefore MDT is not a sufficient theory of reduplication; BRCT can generate it, so it is a sufficient theory of reduplication. The MDT analysis proposed in this paper complicates this picture. If one believes that this analysis is not valid on theory-internal grounds or incorrect on analytical grounds, then HHK's undergeneration argument is reaffirmed, since we continue not to have a convergent MDT analysis of this pattern. If, on the other hand, one believes that this analysis — and, crucially, all the powerful technologies contained within it — is valid, then this negates the undergeneration argument against MDT, since it now *can* analyze this pattern. The basis for comparison between BRCT and MDT, in this domain at least, would then have to shift to more subtle arguments.

One such type of comparison would be regarding *overgeneration*, preferring the theory that predicts the fewest number of empirical patterns which are not attested. This is the nature of the discussion at the end of the preceding section. The powerful analytical devices required in order to capture the Tawala pattern in MDT would clearly predict all sorts of other complex patterns, many of which are surely unattested. It would, though, be difficult to quantify the extent of overgeneration introduced into MDT by these devices, versus the extent of overgeneration which has been argued to follow from BRCT (see, among many others, IZ, Kiparsky 2010, McCarthy, Kimper, & Mullin 2012). It must be noted, however, that only the introduction of vacuous nodes into reduplication constructions is a potentially novel addition to MDT. The use of faithfulness to morphoprosodic constituency was a core component of IZ's original theory; this analysis simply fleshes that out and brings it to the fore.[24] Pending further investigation into the full predictions of MDT with regards to these devices, I believe the overgeneration argument is moot.

The other type of comparison which remains available is one of parsimony. An argument by parsimony would hold that one of the two analyses is simpler and more elegant, and therefore that theory is to be preferred. This is even less quantifiable, and even more personally subjective, than the overgeneration considerations, and thus should not be given a huge amount of weight in theory comparison. However, I believe that the argument by parsimony in this case, such as it is, clearly favors the BRCT analysis. By admitting BR-faithfulness constraints into a parallel evaluation, the BRCT analysis can capture the full set of alternations, along with the stress facts, in a single constraint ranking (cf. (62)). All interactions are transparent and surface-oriented, and hold true of the language on the whole, granting that those constraints which include morphological information have a limited scope. On the other hand, the MDT analysis requires four highly differentiated cophonologies (see (93), (99), (106), and (115)), all of which contain at least some processes which are not surface true and are tailor-made for the problems at hand. For whatever it's worth, parsimony definitively favors the BRCT analysis in this case.

[24] Morphoprosodic faithfulness is not inherently bound to MDT; it could just as easily (if not more so) be implemented in BRCT. The question is whether the other components of that theory allow us to do without morphoprosodic faithfulness, and whether the architecture of MDT makes morphoprosodic faithfulness excessively powerful.

6.2 Outlook

Assuming that the MDT analysis of Tawala proposed above is to be considered valid, where does this leave us? First, it should make clear that putatively "base-dependent" reduplicant shape alternations are not a reliable domain for comparing between BRCT and MDT. The Tawala pattern fits any reasonable, theory-independent characterization of a base-dependent reduplicant shape alternation. Therefore, it must said that MDT *can* generate such patterns, and that such patterns do exist. If one were to deny that the pattern exhibits base-dependence on the grounds that MDT can analyze it, then this voids the prediction of any content in the first place.

Nevertheless, there are at least two other putative/potential types of base-dependence in reduplication:

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

a. Prosodic constituent copying (HHK)

b. Certain opaque reduplication-phonology interactions (IZ)

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

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

This leaves only reduplication-phonology interactions (Wilbur 1973) as a grounds on which to evaluate base-dependence. By reduplication-phonology interactions, we mean patterns that have been characterized as over-application or under-application opacity, back-copying, re-copying, etc. The existence of these patterns is very much a live question in the field (e.g., McCarthy & Prince 1995, IZ, Kiparsky 2010, McCarthy, Kimper, & Mullin 2012). IZ argue that none of these patterns exist as such. They take on many of the claimed instances of such patterns and provide alternative, frequently more comprehensive, analyses that do not rely on the base-dependent interpretation proposed in previous literature. Many of these debates actually hinge on *empirical questions* rather than theoretical ones. Therefore, to the extent that base-dependence is a useful concept for distinguishing between theories of reduplication, we should continue to focus our attention on reduplication-phonology interactions, not reduplicant shape alternations.

References

Alderete, John, Jill Beckman, Laura Benua, Amalia Gnanadesikan, John McCarthy & Suzanne Urbanczyk. 1999. Reduplication with Fixed Segmentism. *Linguistic Inquiry* 30(3):327–364.

Anttila, Arto. 2002. Morphologically Conditioned Phonological Alternations. *Natural Language & Linguistic Theory* 20(1):1–42.

———. 2009. Derived Environment Effects in Colloquial Helsinki Finnish. In Kristin Hanson & Sharon Inkelas (eds.), *The Nature of the Word: Studies in Honor of Paul Kiparsky*, 433–460. Cambridge, MA: MIT Press.

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

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

Becker, Michael. 2009. Phonological Trends in the Lexicon: The Role of Constraints. PhD Dissertation, University of Massachusetts, Amherst.

Beckman, Jill N. 1998. Positional Faithfulness. PhD Dissertation, University of Massachusetts, Amherst.

Berko, Jean. 1958. The Child's Learning of English Morphology. *Word* 14:150–177.

Booij, Geert E. 1985. Coordination Reduction in Complex Words: A Case for Prosodic Phonology. In Harry van der Hulst & Norval Smith (eds.), *Advances in Nonlinear Phonology*, vol. 7, 143–160. Dordrecht: Foris.

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

Coetzee, Andries W. 2009. Learning Lexical Indexation. *Phonology* 26(1):109–145. doi:10.1017/S0952675709001730.

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

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

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

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

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

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

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

Goldsmith, John A. 1976. Autosegmental Phonology. PhD Dissertation, MIT.

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

Halle, Morris & Alec Marantz. 1993. Distributed Morphology and the Pieces of Inflection. In Ken Hale & Samuel Jay Keyser (eds.), *The View from Building 20: Essays in Honor of Sylvain Bromberger*, 111–176. Cambridge, MA: MIT Press.

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.

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

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

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

———. 2007. Is Grammar Dependence Real? A Comparison Between Cophonological and Indexed Constraint Approaches to Morphologically Conditioned Phonology. *Linguistics* 45(1):133–171.

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

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.

Kraska-Szlenk, Iwona. 1997. Exceptions in Phonological Theory. In *Proceedings of the 16th International Congress of Linguists*, Paper number 0173. Oxford: Pergamon.

Leben, William R. 1973. Suprasegmental Phonology. PhD Dissertation, MIT.

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

McCarthy, John J. 1986. OCP Effects: Gemination and Antigemination. *Linguistic Inquiry* 17(2):207–263.

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

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

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

———. 1993b. Prosodic Morphology I: Constraint Interaction and Satisfaction. *Linguistics Department Faculty Publication Series* 14 (2001 version).

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

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

———. 1999. Faithfulness and Identity in Prosodic Morphology. In René Kager, Harry van der Hulst & Wim Zonneveld (eds.), *The Prosody-Morphology Interface*, 218–309. Cambridge: Cambridge University Press.

Menn, Lise & Brian MacWhinney. 1984. The Repeated Morph Constraint: Toward an Explanation. *Language* 60(3):519–541. doi:10.2307/413990.

Nazarov, Aleksei Ioulevitch. 2019. Bedouin Arabic Multiple Opacity with Indexed Constraints in Parallel OT. In Hyunah Baek, Chikako Takahashi & Alex Hong-Lun Yeung (eds.), *Proceedings of the 2019 Annual Meeting on Phonology*, 1–12. Washington, DC: LSA. doi:10.3765/amp.v8i0.4687.

Nespor, Marina & Irene Vogel. 1986. *Prosodic Phonology*. Dordrecht: Foris.

Orgun, Cemil Orhan. 1996. Sign-Based Morphology and Phonology with Special Attention to Optimality Theory. PhD Dissertation, University of California, Berkeley.

———. 1998. Cyclic and Noncyclic Phonological Effects in a Declarative Grammar. In Geert Booij & Jaap van Marle (eds.), *Yearbook of Morphology 1997*, 179–218. Dordrecht: Springer. doi:10.1007/978-94-011-4998-3_7.

———. 1999. Sign-Based Morphology: A Declarative Theory of Morphology-Phonology Interleaving. In Ben Hermans & Marc van Oostendorp (eds.), *The Derivational Residue in Phonological Optimality Theory*, 247–267. Amsterdam: John Benjamins.

Orgun, Cemil Orhan & Sharon Inkelas. 2002. Reconsidering Bracket Erasure. In Geert Booij & Jaap van Marle (eds.), *Yearbook of Morphology 2001*, 115–146. Dordrecht: Springer.

Pater, Joe. 2000. Non-Uniformity in English Secondary Stress: The Role of Ranked and Lexically Specific Constraints. *Phonology* 17(2):237–274.

———. 2007. The Locus of Exceptionality: Morpheme-Specific Phonology as Constraint Indexation. In Leah Bateman, Michael O'Keefe, Ehren Reilly & Adam Werle (eds.), *Papers in Optimality Theory III*, 259–296. Amherst, MA: GLSA.

———. 2008. Gradual Learning and Convergence. *Linguistic Inquiry* 39(2):334–345.

———. 2009. Morpheme-Specific Phonology: Constraint Indexation and Inconsistency Resolution. In Stephen Parker (ed.), *Phonological Argumentation: Essays on Evidence and Motivation*, 123–154. London: Equinox.

Prince, Alan. 1983. Relating to the Grid. *Linguistic Inquiry* 14(1):19–100.

Prince, Alan & Paul Smolensky. [1993] 2004. *Optimality Theory: Constraint Interaction in Generative Grammar*. Malden, MA: Blackwell Publishing.

Riggle, Jason. 2006. Infixing Reduplication in Pima and its Theoretical Consequences. *Natural Language & Linguistic Theory* 24(3):857–891. doi:10.1007/s11049-006-9003-8.

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

Sande, Hannah & Peter Jenks. 2018. Cophonologies by Phase. In Sherry Hucklebridge & Max Nelson (eds.), *NELS 48: Proceedings of the Forty-Eighth Annual Meeting of the North East Linguistic Society*, vol. 3, 39–53. Amherst, MA: Graduate Linguistics Student Association.

Sande, Hannah, Peter Jenks & Sharon Inkelas. 2020. Cophonologies by Ph(r)Ase. *Natural Language & Linguistic Theory* 38(4):1211–1261. doi:10.1007/s11049-020-09467-x.

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

Selkirk, Elisabeth. 1984. *Phonology and Syntax: The Relation Between Sound and Structure*. Cambridge, MA: MIT Press.

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

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

Sproat, Richard. 1986. Malayalam Compounding: A Non-stratum Ordered Account. In Mary Dalrymple, Jeffrey Goldberg & Kristin Hanson (eds.), *Proceedings of the Fifth West Coast Conference on Formal Linguistics*, 268–288. Stanford: Stanford Linguistics Association.

Stanton, Juliet. 2014. A Cyclic Factorial Typology of Pama-Nyungan Stress. In John Kingston, Claire Moore-Cantwell, Joe Pater & Robert Staubs (eds.), *Supplemental Proceedings of the 2013 Annual Meeting on Phonology*, Washington, D.C.: Linguistic Society of America.

———. 2015. Factorial Typology and Accentual Faithfulness. In Ulrike Steindl, Thomas Borer, Huilin Fang, Alfredo García Pardo, Peter Guekguezian, Brian Hsu, Charlie O'Hara & Iris Chuoying Ouyang (eds.), *WCCFL 32: Proceedings of the 32nd West Coast Conference on Formal Linguistics*, 54–63. Somerville, MA: Cascadilla Proceedings Project.

Stanton, Juliet & Sam Zukoff. 2018. Prosodic Identity in Copy Epenthesis: Evidence for a Correspondence-Based Approach. *Natural Language & Linguistic Theory* 36(2):637–684. doi:10.1007/S11049-017-9385-9.

———. 2021. Emergence of the Unmarked in Scottish Gaelic Copy Epenthesis. Paper Presented at Epenthsis and Beyond, Stony Brook University, September 17–19, 2021. https://www.samzukoff.com/epenthesis2021handout.

Sumukti, Rukmantoro Hadi. 1971. Javanese Morphology and Phonology. PhD Dissertation, Cornell University.

Suzuki, Keiichiro. 1998. A Typological Investigation of Dissimilation. PhD Dissertation, University of Arizona.

Tesar, Bruce & Paul Smolensky. 1998. Learnability in Optimality Theory. *Linguistic Inquiry* 29(2):229–268.

———. 2000. *Learnability in Optimality Theory*. Cambridge, MA: MIT Press.

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. In Keiichiro Suzuki & Dirk Elzinga (eds.), *Proceedings of South Western Optimality Theory Workshop 1995*, Tucson, Arizona: Department of Linguistics, University of Arizona. https://repository.arizona.edu/handle/10150/227280.

———. 1998. Identity Avoidance in Phonology and Morphology. In Steven Lapointe, Diane Brentari & Patrick Farrell (eds.), *Morphology and its Relation to Phonology and Syntax*, 216–246. Stanford: CSLI Publications.

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.

———. 2017a. Indo-European Reduplication: Synchrony, Diachrony, and Theory. PhD Dissertation, MIT. https://www.samzukoff.com/zukoffdiss.

———. 2017b. The Reduplicative System of Ancient Greek and a New Analysis of Attic Reduplication. *Linguistic Inquiry* 48(3):459–497. doi:10.1162/ling_a_00250.

———. 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.

———. 2023. The Mirror Alignment Principle: Morpheme Ordering at the Morphosyntax-Phonology Interface. *Natural Language & Linguistic Theory.* 41(1):399–458. doi:10.1007/s11049-022-09537-2.