

# Tagalog Reduplication

## *A Correspondence-Based Approach*

*Ben Keil*

### 0. Introduction

Tagalog is the official language of the Philippines, and has a native speaker population of around seventeen million. Tagalog has at least two reduplicative morphemes, one of which is monosyllabic, while the other is comprised of two syllables. In this paper, I give a treatment to a handful of phonological phenomena related to the bisyllabic reduplicant; namely, I give an analysis of the amount of phonological material copied, the placement of stress on the reduplicant, and the where the reduplicant goes in relation to the stem. To place this work in context, I will show how the analysis can be extended to account for the patterns of Balangoa and Yidin<sup>y</sup> reduplication and compare my analysis with a Marantzian style analysis of copying to a CV-skeleton.

### 1. The Size of the Reduplicant

The reduplicant under consideration here is always a two syllable morpheme. This amount of phonological material is often identified as a “prosodic foot.” I posit, therefore, that there is a constraint that requires the reduplicant to be at least a foot:

- (1) **RED=FOOT**: The reduplicant must contain sufficient phonological material to license a foot. Assess a candidate one violation if the material of the reduplicant is too sparse.

The reduplicant is never more than two syllables because there is also a constraint working against longer forms. This constraint is **\*STRUCTURE-SYLLABLE**, and it assesses a candidate one violation for every syllable that it is comprised of. **\*Structure-Syllable** dominates **MAX-BASE/REDUPLICANT** the constraint that would enforce copying the entire base into the reduplicant. The tableaux in (3) and (4) demonstrate the way in which these three constraints work to enforce a bisyllabic reduplicant.

- (2) **MAX BASE/REDUPLICANT (MAXBR)**: Assess a candidate one violation for every segment in the Base which does not have a correspondent in the Reduplicant.

- (3) **RED=FOOT** >> **\*STRUCT-SYL**, because /RED+par'eho/ → [.pa.'re.pa.'re.ho.], not \* [.pa.pa.'re.ho.]:

/RED+par'eho/	<b>RED=FOOT</b>	<b>*STRUCT-SYL</b>	<b>MAXBR</b>
☞ [.pa.'re.pa.'re.ho.]		*****	**
* [.pa.pa.'re.ho.]	*!	****	****

- (4) **\*STRUCT-SYL** >> **MAXBR**, because /RED+par'eho/ → [.pa.'re.pa.'re.ho.], not \* [.pa.'re.ho.pa.'re.ho.]:

/RED+par'eho/	<b>RED=FOOT</b>	<b>*STRUCT-SYL</b>	<b>MAXBR</b>
---------------	-----------------	--------------------	--------------

☞[.pa.'re.pa.'re.ho.]		*****	**
*[.pa.'re.ho.pa.'re.ho.]		*****!	****

The reduplicant, when it is smaller than the base always contains material from the beginning of the base rather than the material from the end of the word. This is motivated by a constraint that requires the leftmost segment in the reduplicant to correspond to the leftmost segment in the base. This kind of correspondence between “edgemo” segments is called anchoring. Tagalog has active constraints that anchor both the left and right edges of the reduplicant to the corresponding edges of the base, but the constraint militating for left-anchoring outranks the constraint militating for right-anchoring, as shown in (6).

(5) **ANCHOR BASE/REDUPLICANT [LEFT,RIGHT] (ANC-BR-[L,R])**: Assess a candidate one violation if the Leftmost [Rightmost] segment of the Reduplicant is not in correspondence with the Leftmost [Rightmost] segment of the Base.

(6) **ANC-BR-L** >> **ANC-BR-R**, because /RED+sunda'lu/ → [.sun.'da.sun.da.'lo.], not \* [.da.'lo.sun.da.'lo.]:

/RED+sunda'lu/	<b>ANC-BR-L</b>	<b>ANC-BR-R</b>	<b>*CODA</b>	<b>MAXBR</b>
☞[.sun.'da.sun.da.'lo.]		*	**	**
*[.da.'lo.sun.da.'lo.]	*!		*	***

The anchoring constraints also explain the pattern of coda copying in the Tagalog reduplicant. The final coda of bisyllabic stems is copied, but if the second syllable of a word with three or more syllables has a coda, it is not copied into the reduplicant. This is a kind of emergence of the unmarked. The coda-less syllable emerges when right-anchoring is not possible; in two syllable words, the rightmost segment of the reduplicant can be in correspondence with the rightmost segment of the base, but this is not possible in polysyllabic roots because the right edge of the base is out side of the two syllable target of reduplication. The fact that codas are not copied when they do not need to be copied is the effect of **\*CODA**---the constraint against syllables having codas---outranking **MAX BASE/REDUPLICANT** but being outranked by **ANCHOR BASE/REDUPLICANT RIGHT**, an interaction demonstrated in the following tableaux.

(7) **ANC-BR-R** >> **\*CODA**, because /RED+?akb'aj/ → [.?ak.'baj.?ak.'baj.], not \* [.?ak.'ba.?ak.'baj.]:

/RED+?akb'aj/	<b>ANC-BR-R</b>	<b>*CODA</b>	<b>MAXBR</b>
☞[.?ak.'baj.?ak.'baj.]		****	
*[.?ak.'ba.?ak.'baj.]	*!	***	*

(8) **\*CODA** >> **MAXBR**, because /pa+RED+balukt'ot/ → [.pa.ba.'lu.ba.luk.'tot.], not \* [.pa.ba.'luk.ba.luk.'tot.]:

/pa+RED+balukt'ot/	<b>ANC-BR-R</b>	<b>*CODA</b>	<b>MAXBR</b>
☞[.pa.ba.'lu.ba.luk.'tot.]	*	**	****
*[.pa.ba.'luk.ba.luk.'tot.]	*	***!	***

There is a candidate for the tableau in (8) that would defeat the actual winner with the constraints that I have mentioned so far. This is a candidate that maintains anchoring on both the right and left edges, skipping material in the middle of the base, e.g., [pa.ba.'tot.ba.luk.'tot]. This violates a highly ranked constraint against skipping the middle of a string when copying: **CONTIGUITY BASE/REDUPLICANT**.

(9)**CONTIGUITY BASE/REDUPLICANT (CONT-BR)**: Assess a violation to a candidate for every pair of contiguous segments in the Reduplicant whose correspondents in the Base are not contiguous.

(10)**CONT-BR**  $\gg$  {**ANC-BR-R**, \***CODA**, \***MaxBR**}, because /pa+RED+balukt'ot/  $\rightarrow$  [.pa.ba.'lu.ba.luk.'tot.], not \* [.pa.ba.'tot.ba.luk.'tot.]:

/pa+RED+balukt'ot/	<b>CONT-BR</b>	<b>ANC-BR-R</b>	* <b>CODA</b>	<b>MaxBR</b>
☞ [.pa.ba.'lu.ba.luk.'tot.]		*	**	****
* [.pa.ba.'tot.ba.luk.'tot.]	*!		***	***

Contiguity stops a number of nasty things from happening. Apart from preventing middles to be skipped in order to satisfy both anchorings, it also makes sure that initial syllable codas are not omitted to satisfy \***CODA** or and that larger syllables later in the word do not displace the second syllable in order to satisfy **MaxBR** (both effects are visible in (10)). Moreover---due to the intervention of MaxIO, the constraint against segments in the input without correspondents in the base---, the later syllables may not be dropped from the base in order to bring the right edges into correspondence, nor can syllables be dropped from the base for the purpose of satisfying \***STRUCT-SYL** or **MaxBR** as shown in (11).

(11)**MaxIO**  $\gg$  {**ANC-BR-R**, **MaxBR**, \***STRUCT-SYL**}, because /RED+sunda'lu/  $\rightarrow$  [.sun.'da.sun.da.'lo.], not \* [.sun.'da.sun.'da.]:

/RED+sunda'lu/	<b>MaxIO</b>	<b>ANC-BR-R</b>	<b>MaxBR</b>	* <b>STRUCT-SYL</b>
☞ [.sun.'da.sun.da.'lo.]		*	**	*****
* [.sun.'da.sun.'da.]	*!*			****

Finally, glottal stops are not copied into the final coda position, even to satisfy **ANCHOR RIGHT BASE/REDUPLICANT**. I have implemented this in my analysis by placing \***CODA-?** between **MaxIO** (bases have glottal stops in their codas) and the right anchor constraint.

(12)**MaxIO**  $\gg$  \***CODA-?**, because /RED+laj'o?/  $\rightarrow$  [.la.'ju.la.'jo?], not \* [.la.'ju.la.'jo.]:

/RED+laj'o?/	<b>MaxIO</b>	* <b>CODA-?</b>
☞ [.la.'ju.la.'jo?]		*
* [.la.'ju.la.'jo.]	*!	

(13)\***CODA-?**  $\gg$  {**ANC-BR-R**, **MaxBR** }, because /RED+laj'o?/  $\rightarrow$  [.la.'ju.la.'jo?], not \* [.la.'ju?.la.'jo?]:

/RED+laj'o?/	* <b>CODA-?</b>	<b>ANC-BR-R</b>	<b>MaxBR</b>

☞[.la.'ju.la.'jo?.]	*	*	*
*[.la.'ju?.la.'jo?.]	**!		

## 2. Stress in the Reduplicant

The facts of stress placement in Tagalog have a fairly simple description. To the extent that it is possible, stress in the reduplicant matches stress in the base. There are an abundance of solutions to this. The stress placement in (14), for example, not only mirrors the stress of the base, but also eliminates stress clashes and reduces lapses. Additionally, because the reduplicant stress only fails to match the base when the stress on the base falls later than the second syllable (i.e., too far to the right to be copied), placing stress on any syllable other than the rightmost syllable of the reduplicant will cause a greater lapse between the two stresses.

(14)

- a. [grú.po.grú.po] matches stress between the base and the reduplicant.
- b. \*[gru.pó.grú.po] has a stress clash.
- c. [pa.ba.lú.ba.luk.tót] does not match the stress of the base, but comes as close as possible---it is off by one. It suffers from two minor lapses ([pa.ba] and [ba.luk]).
- d. \*[pa.bá.lu.ba.luk.tót] has stress that is off by two (if phonology “counts”), and suffers from an extended lapse ([lu.ba.luk]).

(15) Possible solutions:

- a. **IDENTICAL BASE/REDUPLICANT STRESS** outranks **ALIGN(STRESSED-SYLLABLE, R, FOOT, R)** which chooses stress when identity is not possible
- b. Active **\*CLASH**, **\*LAPSE**, and **\*EXTENDED LAPSE** constraints.

There must be fifty ways for the stress to leave its leftmost syllable lover and appear on the rightmost syllable of the reduplicant. but whatever the case in Tagalog is, it cannot subvert the contiguity of the reduplicant to achieve its aims, as demonstrated in (16).

**CONT-BR** >> **ID-BR-STR**, because /pa+RED+balukt'ot/ → [.pa.ba.'lu.ba.luk.'tot.], not \*[.pa.ba.'tot.ba.luk.'tot.]:

/pa+RED+balukt'ot/	<b>CONT-BR</b>	<b>ID-BR-STR</b>
☞[.pa.ba.'lu.ba.luk.'tot.]		*
*[.pa.ba.'tot.ba.luk.'tot.]	*!	

An additional thing to mention about stress in Tagalog is that it tends to shift right under suffixation. This is implemented in my analysis as an ad hoc constraint called “**SHIFT**” which penalizes a candidate for not undergoing this stress shift. Crucially, this constraint outranks the constraint that militates against the stress of the base differing from the stress of the input (assuming that the input has stress!).

(16) **SHIFT**  $\gg$  **ID-IO-STR**, because /RED+s'anto+han/  $\rightarrow$  [.san.'tu.san.'tu.han.], not \*['san.tu.'san.tu.han.]:

/RED+s'anto+han/	<b>SHIFT</b>	<b>ID-IO-STR</b>
☞ [.san.'tu.san.'tu.han.]		*
*['san.tu.'san.tu.han.]	*!	

### 3. Placement of the Reduplicant

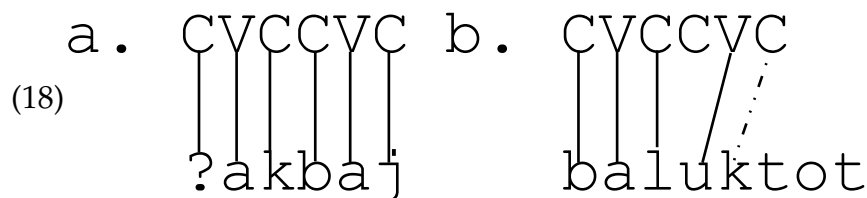
The reduplicant is a prefix, which in my analysis can be guaranteed by the alignment constraint in (17):

(17) **ALIGN[RED,R,BASE,L] (PREFIX)**: assess a candidate one violation for every segment intervening between the right edge of the reduplicant and the left edge of the base.

/RED+men'os/	<b>PREFIX</b>
☞ [.me.'nos.me.'nos.]	
*[.me.me.'nos.'nos.]	*!*

### 4. Marantzian CV-Skeleton

Tagalog offers a significant challenge to the analyst using a CV-skeleton copying approach. The crucial forms to demonstrate this are [ʔak.baj.ʔak.baj] and [ba.lu.ba.luk.tot]. The skeleton has to be at least CVCCVC to account for the first form, but then it is very unclear what would stop the the second form from copying [ba.luk]:



It seems that a easy way to extend the Marantzian framework would be to add a diacritic for some prosodic boundaries. The Tagalog framework would then be something like CVCCV(C)), where the parentheses show grouping. The [j] of [ʔakbaj] would match the template and be copied while the [k] of [baluktot] would not meet the criterion. Cross-linguistically, there are a number of languages that allow only word final codas. If this tendency is due to a perceptual advantage for word final codas, then perhaps this would be a useful thing to encode.

### 5. Balangao

The reduplicative morpheme in Balangao behaves just like the morpheme in Tagalog, with the exception that it does not a coda for the second syllable ever. The emergent phonology of the reduplicant is even less marked in Balangao than it is in Tagalog. This is due to one of the faithfulness candidates having a lower ranking in Balangao that it

does in Tagalog: the right anchoring constraint. Crucially, in Balangao, the phonology would rather avoid codas than maintain an anchoring (i.e., whereas Tagalog's phonology has the opposite preference).

## 6. Yidin<sup>y</sup>

Just as Balangao copied fewer codas than Tagalog, Yidin<sup>y</sup> copies more. The Yidin<sup>y</sup> reduplicant has a coda in the second syllable every time that the second syllable of the base has a coda. In a sense, this means that there is some special correspondence between the prosodic nodes of the base and reduplicant. I posit that a constraint like (19):

(19) **IDENTICAL BASE/REDUPLICANT RHYME [BRANCHING]**: Corresponding rhyme nodes in the Reduplicant and Base nodes must have the same branching status. Assess a violation if either the rhyme in the reduplicant is branching and the rhyme in the base is not, or if the base has a branching rhyme which corresponds to a non-branching rhyme in the reduplicant.

The critical ranking for Yidin<sup>y</sup> is that the constraint in (19) outrank \***CODA**. It must also be outranked by **MAXIO**, or else both (19) and \*Coda could be satisfied by eliminating the coda in the base. This suggests that there might be pathological consequences for (19) in the factorial typology, but perhaps (19) can be skewed in such a way that it only ever involves additional copying and never motivates base-deletion.

## Reduplication in Frisian

*I'm clueless*

To the best of my knowledge there is no reduplication in Frisian, but it seems likely that there would be many pseudo-reduplicative forms in the language. As of yet, however, I have not located any. The only dictionary I have available at this point is a 2,500 word Berlitz-style traveler's dictionary.