Sliding-window reduplication: an overgeneration problem for BR-correspondence theories

J. Cooper Roberts

June 27, 2024

1 Introduction

[1]Reduplication is a process that exhibits interesting variation along several different dimensions. There is of course the issue of reduplicant size, which can be as small as consonant and as large as a syntactic constituent, with intermediate units of prosodic structure bridging the two. Reduplication can also differ in locality, with some seeming to copy from the opposite side of a root (cf. (1), (2)). Reduplication can also be infixal in nature, again varying in terms of locality (cf. (3), (4)). A copy need not even be a continuous substring of its base, with some reduplication processes concatenating base edges (5).

(1) tʲilparku tʲilpa-tʲilparku 'type of bird(y)' Diyari (Austin, 1981)

(2) nute nute-nut 'ground' Chukchi (Marantz, 1982)

(3) magizem ma<gize>gizem '(very) strong' Pazeh (Blust, 1999)

(4) copóksin copok<có:>sin 'to be a hill' Koasati (Kimball, 1988)

(5) malaboŋ maŋ-malaboŋ 'flying fox' Ulu Muar Malay (Kroeger, 1989)

In Optimality Theory, where the phonologies of individual languages differ in the relative ranking of constraints, this data poses an interesting challenge. How do we account for this rich variation, while excluding unattested patterns? One account for this variation comes from McCarthy and Prince (1995) (abbreviated *M&P*), which uses constraints in (6) to enable anchoring/alignment mismatches. Grounded in the framework of Correspondence Theory (McCarthy and Prince, 2004), reduplication is thought of here as correspondence between a reduplicant and a base. This is particularly advantageous for deriving phonological under/overapplication in reduplicative environments.

(6) a. BR-ANCHOR({L,R}): Assign a violation mark for every reduplicant that does not copy the {left, right} edge of its base.
b. BR-CONTIG: Assign a violation mark for every reduplicant that is not a continuous substring of its corresponding base.

In tandem with alignment constraints that target the reduplicant in question, it is possible to derive many of these patterns. Crucially, though, this system also predicts that some forms of reduplication are impossible. Cases like *taka-patakan* in (7), which violates both BR-ANCHOR constraints, is excluded regardless of any ranking. This is a desirable effect, as pre/suffixal reduplication which copies the inside of a base (as opposed to at least one of the edges) is unattested.

To derive the internal reduplicant forms, an additional constraint on locality is necessary (e.g, Lunden (2006)).[2] With this minimal addition, however, we account for the entire typology exemplified in (1)-(5), shown in the tableau in (9).

[1]This work, in its current instantiation, is indebted to Adam Albright and participants at the MIT MorPhun workshop. Errors are my own. I welcome all comments and criticism, which can be sent to `jcrobert [at] mit [dot] edu`. Excessive whitespace used only to improve readability of tableaus interspersed throughout text.

[2]My impression of the literature suggests that LOCALITY is a replacement to BR-ANCHOR constraints, not a supplement for it. I treat it as a supplement here as I'm not sure how to derive edge-construed internal reduplicants without BR-ANCHOR.

RED($\mu\mu$) + patakan	Align(RED,L)	Align(RED,R)	BR-Anchor(L)	BR-Anchor(R)	BR-Contig
pata-patakan		*		*	
patakan-pata	*			*	
kan-patakan		*	*		
patakan-kan	*		*		
pan-patakan		*			*
*taka-patakan		*	*	*	

(7)

(8) Locality: Assign a violation mark to every candidate where there is any intervening substring between the edge of the base and the edge of the reduplicant.

RED(μ) + bedegéle	Al(RED,'σ)	BR-Anch(L)	BR-Anch(R)	Loc
bede<be>géle			*	*
bede<le>géle		*		*
bede<de>géle		*		
bede<ge>géle		*		

(9)

Overall, the approach of M&P is a relatively successful one; not only does it generate the typology, but it also provides a more streamlined account of the facts than previous frameworks, e.g., the template model (Marantz, 1982). However, there is problem concerning the copying of marked structures which poses a problem for M&P's proposal. Consider this case of partial reduplication in Tagalog (McCarthy and Prince, 1986), presented in the tableau in (10). Here, the target of RED is a syllable with a complex onset. It is an independent fact of Tagalog that complex onsets are generally dispreferred, so the phonological grammar tries to prevent having such a sequence in the reduplicant. There are a number of ways to do this, but the issue is resolved in Tagalog by simply dropping the second consonant in the cluster.

RED + trabaho	Al(RED,L)	Loc	*CCV	MaxRed
→ a. ta-trabaho			*	*
b. tra-trabaho			**!	
c. tra<ba>baho	*!**		*	
d. ba-trabaho		*!	*	

(10)

Since OT is a tool for capturing phonological typology, though, it is reasonable to conclude that other phonologies may have different rankings of these constraints. For example, we could imagine a grammar where every other constraint outranks *CCV (tableau provided in (11)). As it turns out, there are attested cases of such repairs for reduplicants; in Chamorro, a reduplicant can move one consonant into the base to avoid a consonant cluster (12).

RED + trabaho	*CCV	Loc	MaxRed	Al(RED,L)
→ a. tra<ba>baho	*			***
b. tra-trabaho	**!			
c. ta-trabaho	*		*!	
d. ba-trabaho	*	*!		

(11)

(12) a. dankolo dankolo-lo '(very) big'
b. metgot metgo<go>t '(very) strong'

What is not attested, however, is a grammar like that of (13). In this case, it is Locality that is outranked by all other constraints. As a result, the best candidate is one where RED copies something further away yet less marked.

RED + trabaho	*CCV	Al(RED,L)	MaxRed	Loc
→ a. ba-trabaho	*			*
b. tra-trabaho	**!			
c. ta-trabaho	*		*!	
d. tra<ba>baho	*	**!*		

(13)

Here lies the problem for M&P. According to the constraints in our toy analysis, there are three possible repairs for the reduplication of marked sequences (14). Despite all the reduplication patterns in the world's languages, though, only (14-a) and (14-b) exist. The repair in (14-c), which I refer to as *sliding-window reduplication* (SWR), is not realized in any language. For some reason, Locality stubbornly refuses to be violated in this context. This is especially puzzling when we remember that nonlocal reduplication exists outside the context of reduplicant repair.

(14) If the target of RED is marked...
a. ✓Copy units which satisfy RED, then phonologically-reduce (e.g., (10))
b. ✓Move next to a string bearing units which satisfy RED, then copy (e.g., (11))
c. *Find another string bearing units which satisfy RED, copy without moving (e.g., (13))

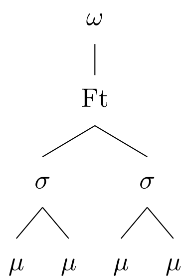
Ergo, we now have a puzzle on our hands. As far as I am aware, this is a novel observation about Base-Reduplicant Correspondence theories of reduplication, so there is no off-the-shelf solution. In this squib, I work towards an analysis based on Serial Template Satisfaction (McCarthy et al., 2012), which in turn is grounded in Harmonic Serialism (McCarthy, 2000). Specifically, I limit the power of Gen to prohibit nonlocal reduplication. This is complemented by alignment constraints which differentiate between 'headed' structure (i.e., parts of the representation which bear phonetic content) and empty templates. Working in tandem, they are able to derive most of the reduplication typology while disallowing the unattested repair in (13). The structure is as follows: I begin by introducing Serial Template Satisfaction and Harmonic Serialism. I follow this section with a presentation of my analysis which is grounded in the previous two frameworks. I will show that with some tweaking of STS's machinery, it is possible to derive most of the attested reduplication patterns, all of the correct markedness repairs, but crucially, not SWR. I conclude with a summary of the findings, some shortcomings, and ideas for future work.

2 About Serial Template Satisfaction

Serial Template Satisfaction (henceforth STS) is an account for reduplication proposed McCarthy et al. (2012) and framed as an alternative to M&P's BR-Correspondence approach. As mentioned in 1, STS assumes Harmonic Serialism (henceforth HS), a variant of Optimality Theory with some key differences. In vanilla OT, Gen—the component responsible for generating candidates which are then evaluated by the phonological grammar—is completely unbounded, meaning that the candidates to be evaluated comprise an infinitely-large set. In HS, however, Gen is limited to making only one change to the underlying representation at a time. The finite set of candidates produced by Gen are then evaluated by the grammar and an optimal candidate is chosen. Then, more candidates are created by Gen from this optimal candidate, repeating the cycle. A winner is chosen when no amount of further changes would create a more-optimal form (i.e., the optimal candidate is faithful to the previous optimal candidate).

McCarthy et al. (2012) make the assumption of Marantz (1982) that RED is a template with no phonetic material of their own. To go about filling this template, STS introduces two procedures to Gen, the first being Copy(X). This operation can copy any string of constituents of type X from the input, where X is a unit of the prosodic hierarchy (15) or segmental phonemes. Crurcially, copying any type of structure from the base entails copying the contents of that structure. For example, with the input *kan.pum*, Copy(σ) can copy *kan* and *kanpum* but not *ka* or *kanpu*. Furthermore, discontinuous copying is impossible; i.e., Copy(seg) can copy *kanpu*, *anpum* from based *kanpum* but not *kapu*.

(15)



Any X must be dominated by the unit which dominates it on the prosodic hierarchy, and this also applies to strings generated by Copy(X). The second operation, Insert(X) provides this extra structure by inserting at most one constituent into a candidate. Of course, whatever is inserted must also be hosted by its respective dominating prosodic unit. A mora cannot exist outside of a syllable, which in turn cannot exist outside of a foot, etc. General constraints on well-formed prosodic units protect against illicit patterns (e.g., (16)).

(16) $*[\sigma\ \sigma\ \sigma\ \sigma\ \sigma]_{Ft}$

In an OT framework, any possible change has a respective faithfulness constraint which resists that change. STS is no different, with constraints that punish applications of Copy(X) and Insert(X). In this sense, different sizes of reduplication can be thought of as the result of ranking differences between *Copy constraints and *Insert constraints. These are complemented by constraints on prosodic templates like (17-a) (Selkirk, 1995), as well as a novel constraint (17-b).

(17)
a. Hd(X): Assign a violation mark to every constituent of type X that does not contain a constituent of type X-1 as its head.
b. Copy Locally: To a candidate produced by Copy(X), assign as many violations as there are Xs intervening between the original X string and its copy.

The following tableaus are taken from McCarthy et al. (2012) to demonstrate their account. (18)-(19) shows one reduplicative process in Manam, where the final metrical foot is copied. FirstRED introduced a prosodic template the size of a foot, which now must be populated by syllables. Manam has a strong cline towards foot binarity, which eliminates candidates which do not fill the template with two syllables. If Insert(σ) can only add one syllable at a time, the fastest way to fill this template is with Copy(σ). Copy Locally ensures that the closest two syllables will fill this template, and as a bonus there are no unheaded prosodic structures. Thus, *salaga-laga* wins step 1 in (18). In (19), nothing can be done to further optimize the form, so the phonological grammar is done.

salaga + RED$_{Ft}$	FtBin	Hd(Ft)	Hd(σ)	CopyLoc	*Copy(σ)
→ sa$_\sigma$[la$_\sigma$ga$_\sigma$]$_{Ft}$ [la$_\sigma$ga$_\sigma$]$_{Ft}$					*
sa$_\sigma$ [la$_\sigma$ga$_\sigma$]$_{Ft}$ Ft	*!	*			
sa$_\sigma$ [la$_\sigma$ga$_\sigma$]$_{Ft}$ [σ]$_{Ft}$	*!		*		
sa$_\sigma$[la$_\sigma$ga$_\sigma$]$_{Ft}$ [ga$_\sigma$]$_{Ft}$	*!				*
sa$_\sigma$[la$_\sigma$ga$_\sigma$]$_{Ft}$ [sa$_\sigma$la$_\sigma$]$_{Ft}$				*!	*

(18)

sa$_\sigma$[la$_\sigma$ga$_\sigma$]$_{Ft}$ [la$_\sigma$ga$_\sigma$]$_{Ft}$	FtBin	Hd(Ft)	Hd(σ)	CopyLoc	*Copy(σ)
→ sa$_\sigma$[la$_\sigma$ga$_\sigma$]$_{Ft}$ [la$_\sigma$ga$_\sigma$]$_{Ft}$					
sa$_\sigma$[la$_\sigma$ga$_\sigma$]$_{Ft}$ [la$_\sigma$ga$_\sigma$]$_{Ft}$ Ft	*!	*			
sa$_\sigma$[la$_\sigma$ga$_\sigma$]$_{Ft}$ [la$_\sigma$ga$_\sigma$]$_{Ft}$ [ga$_\sigma$]$_{Ft}$					*

(19)

A case from Balangao is provided in (20)-(22). RED also introduced a foot-sized template in this case, but curiously, Balangao does not copy the coda of the second syllable (*tayna-taynan*, not **taynan-taynan*). This leads McCarthy et al. to conclude that it is actually segments being copied, not syllables. They capture this with a highly-ranked constraint against syllable-copying, favoring candidates which insert syllables templates. These syllable templates are then filled by a string of segments, yielding *tayna-taynan* in (22).

RED$_{Ft}$ + taynan	*Copy(σ)	Hd(Ft)	FtBin	Hd(σ)	*Copy(seg)
→ [σ]$_{Ft}$ [tay$_\sigma$nan$_\sigma$]$_{Ft}$			*	*	
Ft [tay$_\sigma$nan$_\sigma$]$_{Ft}$		*!	*		
[tay$_\sigma$nan$_\sigma$]$_{Ft}$ [tay$_\sigma$nan$_\sigma$]$_{Ft}$	*!				

(20)

McCarthy et al.'s proposal is designed to capture some generalizations that previous work on reduplication missed. This approach accounts for a lack of coda-skipping (e.g., **pati-paltirku*) or simple syllable copying (e.g., pat-pata) in reduplication, patterns which are unattested but generable under previous frameworks. The issue at the focal point of the current study was not considered by McCarthy et al., but the analytical machinery nevertheless seems promising. HS's central tenet of small changes and gradual evaluation may well be useful in accounting for the different kinds of repair strategies available for marked structure reduplication. This detail, paired with some additional constraints and a curbing of Gen's capabilities, will prove to be very useful in my analysis.

$[\sigma]_{Ft}$ $[\text{tay}_\sigma\text{nan}_\sigma]_{Ft}$	*Copy(σ)	Hd(Ft)	FtBin	Hd(σ)	*Copy(seg)
→ $[\sigma\sigma]_{Ft}$ $[\text{tay}_\sigma\text{nan}_\sigma]_{Ft}$				**	
$[\sigma]_{Ft}$ $[\text{tay}_\sigma\text{nan}_\sigma]_{Ft}$			*!	*	
$[\text{ta}_\sigma]_{Ft}$ $[\text{tay}_\sigma\text{nan}_\sigma]_{Ft}$			*!		*
$[\text{tay}_\sigma]_{Ft}$ $[\text{tay}_\sigma\text{nan}_\sigma]_{Ft}$	*!			*	

(21)

$[\sigma\sigma]_{Ft}$ $[\text{tay}_\sigma\text{nan}_\sigma]_{Ft}$	*Copy(σ)	Hd(Ft)	FtBin	Hd(σ)	*Copy(seg)
→ $[\text{tay}_\sigma\text{na}_\sigma]_{Ft}$ $[\text{tay}_\sigma\text{nan}_\sigma]_{Ft}$					*
$[\sigma\sigma]_{Ft}$ $[\text{tay}_\sigma\text{nan}_\sigma]_{Ft}$				*!*	
$[\text{ta}_\sigma\sigma]_{Ft}$ $[\text{tay}_\sigma\text{nan}_\sigma]_{Ft}$				*!	*

(22)

3 Proposal

The main idea of my proposal is this: Gen is incapable of nonlocal copying. What we perceive as nonlocal reduplication is actually phonological material that was copied earlier in the derivation, when the reduplicant and what was copied were actually adjacent. Using a combination of carefully ranked Align and Headedness constraints, it is possible for a template to be filled and then displaced to a separate location. Because Gen is incapable of nonlocal copying, reduplicants are construed with segments at one of the base edges or—if it is an infixed reduplicant—around its pivot. Ergo, sliding-window reduplication is impossible because forms like *ba-trabaho* are never evaluated by the phonology in the first place.

Let us begin by strengthening some of the assumptions of STS. First, I make explicit the operation Move(X),[3] which works with Copy(X) and Insert(X) to create candidates for evaluation. Its candidates are punished by the respective faithfulness constraint Linearity.[4]

(23) Move(X): Move a continuous string of constituents of type X to a different position relative to any other constituent of type X.[5]

I will also posit a version of Copy(X) that is significantly less powerful than that of vanilla STS. McCarthy et al. assumed that nonlocal copying was marked yet possible, constraining its realization with a markedness constraint Copy Locally. As mentioned previously, I assume that only local copying is available.

(24) Copy(X): Copy a continuous string of constituents of type X in the immediate periphery of the template which the copied string will fill s.t. one edge of the base must coincide with the edge of the template.

Recall the basic rule of Gen in HS where at most one change can be made per candidate per round of evaluation. Because of this restriction, there will never be an instance of Move(X) and Copy(X) applying to the same candidate simultaneously.

With these tools, reduplicating material from word edges can be thought of as a very highly-ranked constraint on headedness. Unheaded constituents (i.e., templates) must be filled as soon as possible, so Copy(X) is bound to be the most optimal form in the first round. This urgency to fill the vacant template promotes copying followed by phonological reduction, as moving to find a better target to copy cannot be done in a single turn. Thus, we have successfully derived the first repair (14-a). An example derivation of Tagalog *ta-trabaho* is provided in (25)-(26). At the first step, the highly-ranked headedness constraint and

[3] I suspect that McCarthy et al. (2012) and others working in HS assume something like Move(X), but I make it explicit here. The original work on STS does not discuss internal reduplication at all, so I assume that McCarthy et al. did not assume it to be a relevant part of Gen.

[4] I also assume that on the first round of evaluation, all components entering in the input are linearly concatenated, including infixes. This is against the view of Yu (2007), who advocates for a view of infixes where they are "directly inserted" into their pivots.

[5] e.g., *pa.ta.ka.la* → *ta.ka.pa.la* is generable under Copy(σ), but ta.kpa.a.la is not.

constraint on nucleic consonants ensures *tra-trabaho* will win. It is in the second step of the derivation where markedness constraints on complex onsets gives us phonological reduction.

RED$_\sigma$+trabaho	AL(RED,L)	HD(σ)	*C-NUC	*COPY(seg)	*COMPONS
→ tra-trabaho				*	**
σ-trabaho		*!			*
t-trabaho			*!	*	*
tra$<\sigma>$baho	*!				*

(25)

tra-trabaho	AL(RED,L)	HD(σ)	*C-NUC	*COPY(seg)	*COMPONS
→ ta-trabaho					*
tra-trabaho					**!

(26)

We can apply a similar method to derive nonlocal internal reduplication, which I demonstrate for the Koasati data in tableaus (27)-(28). Once again, the candidate which copies will win out in the first round. In the second round, candidates must now face the constraint BESTRESSED(X). The generalization for this reduplication pattern in Koasati is that the reduplicant becomes the penultimate syllable, i.e. the position of regular stress. In this way, BESTRESS(RED) will tolerate candidates where the reduplicant is the penultimate syllable (via displacement from MOVE(σ)), but an unmoved reduplicant will incur a fatal violation.

RED$_\sigma$+copoksin	HD(σ)	*COPY(seg)	BESTRESS(RED)	LIN
→ co-copoksin		*	*	
σ-copoksin	*!		*	
co$<\sigma>$poksin	*!		*	*

(27)

co-copoksin	HD(σ)	*COPY(seg)	BESTRESS(RED)	LIN
→ copokcosin				*
co-copoksin			*!	

(28)

The move-and-reduplicate repair (14-b) is a bit trickier to implement under this system, but it can still be done. In tableaus (29)-(30) is an analysis of Chamorror *metgogot*. Simply copying the final syllable as a whole would create an extra coda (my constraint is specifically designed to punish codas which are word-internal), so it would be better to copy only a portion of the final syllable. However, GEN cannot copy *go* outright since that string is not sufficiently local to the template. One option might be to copy *ot*, but this does not occur. One way to think of this might be that RED wants the onset of the final syllable of the base. I've formalized this with the constraint ONS$[_]_{RED}$. Finally, a contextual markedness constraint on syllable headedness prevents the syllable from being filled. Thus, the best option is to infix the template. In the second round, the template is not in range of *go*, so it can be copied licitly and without the addition of a coda or an extra consonant cluster.

metgot+RED$_\sigma$	ONS$[_]_{RED}$	NOCODA/word-int	HD(σ)/_#	HD(σ)	*COPY(seg)
→ met$<\sigma>$got		*		*	
metgot-σ		*	*!	*	
metgot-got		**!			*
metgot-ot	*	*!			*

(29)

met$<\sigma>$got	ONS$[_]_{RED}$	NOCODA/word-int	HD(σ)/_#	HD(σ)	*COPY(seg)
→metgogot		*			*
met$<\sigma>$got		*		*	
metgotgot		**!			*

(30)

An alternative to the approach I've outlined in (29)-(30) would be to first copy the final syllable in its entirety and then phonologically reduce the first. McCarthy et al. (2012) actually discuss this in their discuss of STS, and they note that it is a theoretical problem. Is it better to copy and then phonologically-reduce what is copied, or to have multiple copying events which skip the marked content? It is a difficult question, but they offer no answer to general cases. Given that the question is outside the scope of this squib, I will simply conclude that both are possible with the right combinations of constraints and tools in GEN.

Demoting the ranking of HD(X) can also generate patterns of local internal reduplication. For the sake of brevity, I will not divulge the details here. However, it is easy to see how markedness and alignment constraints can punish copying without moving, leading to EVAL to keep the candidate where a template has moved to some word-internal position. From this point, a second round of the derivation can copy something locally to satisfy the template.

Most importantly, though, it should be clear from the discussion so far that SWR is not possible under this framework. To do so, a template would first need to move into the word, then copy something to satisfy the specifications of the template, and then move back to its initial starting position. As far as I can tell, this is impossible, or it's at least highly unlikely that this would be favored over any other candidate over the course of three derivations. So, our modified STS has generated the desired predictions.

4 Concluding Discussion

This squib began with a novel observation concerning overgeneration in Base-Reduplicant correspondence approaches to reduplication. When the target of reduplication is a marked structure, a language's phonology will either phonologically-reduce the reduplicant or move slightly into the word to copy a less-marked constituent. What the phonology will never do, however, is shop around the inside of the base to find a more-optimal reduplicant. While such a reduplication pattern is completely unattested, it is nonetheless possible to compute in BR-Correspondence frameworks. To remedy this, I have suggested an alternative based on Serial Template Satisfaction, a Harmonic Serialism-based approach to reduplication. By strengthening some of its limitations, I have shown that it is possible to derive the two attested repairs without producing SWR. Furthermore, I have shown that local and nonlocal reduplication (both infixal and suffixal varieties) are also derivable under this system.

One thing I did not discuss in my analysis, though, is discontinuous reduplication (e.g., Ulu Muar Malay (5)). Indeed, such cases pose a problem for my analysis as discontinuous and nonlocal copying is not permitted in my system. One approach might be to say that such reduplication is actually a very extreme form of phonological reduction which just so happen to preserve the edges of the base.[6] This is tenable, but further empirical evidence is necessary before any hard conclusions are drawn.

Another problem with this proposal has to do with copying direction. In BR-Correspondence, we could happily throw the entire notion away as we could instead rely on Anchor constraints, and this was not a problem for McCarthy et al. (2012) because they did not consider cases of internal reduplication. However, if we are now allowing the system to generate candidates which have underspecified templates in the middle of a word, this becomes a problem once again. For example, after step 1 of deriving *metgogot* we were left with the candidate *met*$< \sigma >$*got*. Recall that Gen must copy locally, but it technically has two options for syllables from which it can copy: *met* and *got*. We could place a constraint to favor one over the other, but this is doomed to become conjecturist. Continuing to work in this framework where locality matters means solving this problem at some point.

There is one final issue I will bring to light before concluding this squib. In this framework, I must assume that some things are prefixes and some things are suffixes. At the very least, they must start their derivational lives as one of the two. In deriving *tatrabaho*, for example, copying must happen locally and at the very first step of the derivation. This can only work if RED were a prefix, since if it were a suffix it would have to move (and if it has to move, it cannot copy). So, for theory-internal reasons, I must conjecture RED is a prefix. This is not the worst problem, especially since RED superficially resembles a prefix, but we lose some of the agnosticism that alignment constraints could afford us. Under this view, there were no prefixes or suffixes; instead, alignment constraints targeted different lexical items and controlled which side of the word they would appear. Forcing ourselves to adopt the belief that pre/suffixhood can be specified before the phonology may quickly create problems if we are not careful.

In summary, I've provided a theoretical account for an overgeneration problem in BR-Correspondence frameworks. It is not perfect, and critics may rightfully wonder whether it is too constrained or overly-reliant on certain theoretical assumptions. I will maintain, however, that this is a step in the right direction for avoiding excessively-powerful phonologies. Hopefully, the issues I've mentioned will be amended in future work.

References

Austin, P. (1981). A grammar of diyari, South Australia. Cambridge University Press Cambridge.

Blust, R. (1999). Notes on pazeh phonology and morphology. Oceanic Linguistics, pages 321–365.

Kimball, G. (1988). Koasati reduplication. In In honour of Mary Haas: from the Haas Festival Conference on Native American Linguistics, volume 43142.

Kroeger, P. R. (1989). Discontinuous reduplication in vernacular malay. In Annual Meeting of the Berkeley Linguistics Society, pages 193–202.

Lunden, A. (2006). Reduplicant placement, anchoring, and locality. Williamsburg, VA: College of William and Mary.

Marantz, A. (1982). Re reduplication. Linguistic inquiry, 13(3):435–482.

[6] This might be the approach McCarthy et al. (2012) advocate for, but I am not sure.

McCarthy, J. and Prince, A. (1986). Prosodic phonology. Unpublished paper, University of Massachusetts, Amherst.

McCarthy, J. J. (2000). Harmonic serialism and parallelism. In North East Linguistics Society, volume 30, page 8.

McCarthy, J. J., Kimper, W., and Mullin, K. (2012). Reduplication in harmonic serialism. Morphology, 22:173–232.

McCarthy, J. J. and Prince, A. (1995). Faithfulness and reduplicative identity. Linguistics Department Faculty Publication Series, page 10.

McCarthy, J. J. and Prince, A. (2004). Faithfulness and identity in prosodic morphology. Optimality Theory in Phonology: A Reader, pages 77–98.

Selkirk, E. (1995). Sentence prosody: Intonation, stress, and phrasing. The handbook of phonological theory, 1:550–569.

Yu, A. (2007). A natural history of infixation, volume 15. OUP Oxford.