Reduplication as Projection

Charles Reiss & Marc Simpson
Concordia University

November 30, 2009

Contents

1 Introduction

Study of reduplication leads to basic issues that are of fundamental interest for an understanding of the language faculty. Issues of constituency, copying, concatenation, derivational ordering (or its lack) all arise in the study of the phonology of reduplication, the focus of this paper. Consider, for example, a reduplicated form such as Warlpiri *kurdukurdu* 'children', corresponding to the singular *kurdu* 'child'. It is natural to ask which token of the string *kurdu* in the reduplicated form is the base to which the other token of the string is affixed, either as prefix or suffix. The same problem arises when only part of the root is reduplicated, as in Samoan verb pluralizing reduplication, where two copies of the penultimate syllable in the singular appear: *a***lo***fa* 'she, he loves', *a***lolo***fa* 'they love'. The reduplicant appears to be infixed into the root, but which token of *-lo-* is the one that is infixed?

As far as we can tell, most models of reduplication have no answer to these questions. Instead of proposing an answer, we develop a model of reduplication building on recent work by Frampton (2004), Raimy (2000), and especially Halle (2008) in which these questions do not even arise. In addition, there is important work on the morphosyntax and semantics of reduplication (Inkelas and Zoll, 2005), issues that we will not address.[1] The majority of the ideas we present are either explicit or implicit in some earlier work, and we will focus on presenting a unified analysis that draws on these insights, rather than tracing the genealogy of each idea. Our main contributions will be first, to sketch a new typology of reduplication phonology, mainly by developing Halle's insights, and second, to provide a structural analysis of a wide range of reduplicated forms including backcopying phenomena and fixed segmentism.

The Optimality Theoretic (OT) literature on reduplication represents a largely independent stream of research from our main sources. One of the results of this paper is an argument that the claims in the early OT literature that certain reduplication phenomena provide strong evidence for the strictly two-level parallel evaluation models of the original OT models are unfounded.

[1] A project for future research will be to explore whether the scheme we develop, which generates a surprisingly wide range of surface patterns, may be used to show that there is actually more phonological reduplication than Inkelas & Zoll suspected.

Recent versions of Optimality Theory, such as McCarthy's Harmonic Serialism model (McCarthy, 2000) and Kiparsky's Stratal OT (Kiparsky, 2007) embed the parallelism of early OT within multi-level derivational models. The Optimality Theoretic aspect of such hybrid models are presumably maintained in order to exploit the benefits of parallel constraint evaluation provided by OT, including the Base-Reduplicant Identity Conditions (McCarthy and Prince, 1995) touted as probative of the necessity of non-derivational models. We will argue that, instead, the relevant data, in the context of our model, provides very strong support for a derivational model based on ordered phonological rules. The paper thus contributes to debate on the architecture of grammar.

1.1 A partial schematic typology

An informal presentation of reduplication phenomenon can be provided in terms of a morpheme containing a hypothetical string of segments like *patiku*. Cross-linguistically, one finds the following kinds of reduplication structures (as well as others), which are typically described in terms of a Base and Reduplicant (Red) as follows. In standard typology, a partial Red is a Red that is a partial copy of Base; a full Red is a total copy.

(1) Some schematic reduplication patterns

a. *papatiku*—the Base *patiku* with the partial Red *pa* prefixed to the Base
b. *kupatiku*—the Base *patiku* with the partial Red *ku* prefixed to the Base
c. *patikupa*—the Base *patiku* with the partial Red *pa* suffixed to the Base
d. *patikuku*—the Base *patiku* with the partial Red *ku* suffixed to the Base
e. *patitiku*—the Base *patiku* with the partial Red *ti* to the Base
f. *patikupatiku*—with two possible analyses, full Red prefixed or suffixed to the Base

Let's leave (e,f) to the side for now. According to the characterizations given, (a,b) form a natural class of prefixing reduplication, the difference between the members consisting in which end of the Base provides the material in the Red. Similarly, (c,d) form a natural class of suffixing reduplication, with the members of the class differing as in the previous case. Note that the notions of prefix and suffix are dependent on the notion of Base. A prefix/suffix is prefixed/suffixed *to* something, the Base.

Forms (a-e) may also be grouped together in contrast to (f): the former can be described as examples of partial reduplication and the latter as total or full reduplication. So in one analysis, the form in (f) is analyzed as Base-Red, where the Red is a full copy of the Base to which is is suffixed, and in the other analysis, the form is analyzed as Red-Base, in which a full copy of the Base is prefixed to the Base.

1.2 Reduplication without Base

The simple descriptive apparatus underlies most work on reduplication, in particular the OT literature on Base-Reduplicant Correspondence constraints, but also some derivational work. We will argue, however, that it is fundamentally mistaken, failing to cut reduplicative nature at her joints.

Our proposal[2] is the following:

- There is no Base and no Red in reduplication surface forms
- Without Base and Red, the notions of prefixation and suffixation can be dispensed with
- Reduplication involves projection (formalized below) of a string s into a structure containing linearly ordered full and partial copies of s

Thus neither of the following two analyses are correct for a form like *patiku-patiku*:

(2) No affixation

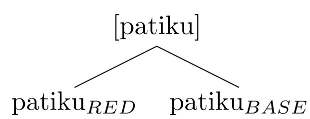
a. Not this: patiku$_{RED}$-patiku$_{BASE}$

b. Not this: patiku$_{BASE}$-patiku$_{RED}$

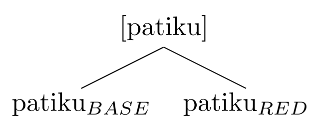
We propose to generate reduplicated forms with a hierarchical structure, again, without Base and Reduplicant, so the following are also *not* appropriate:

(3) Reduplication as multiple projection—no affixation

a. Not this:

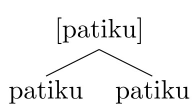


b. Not this:



It is not the case that the form consists of a Reduplicant prefixed to a Base, as in (3b); and it is not the case that it consists of a Reduplicant suffixed to a Base, as in (3b). Instead, our analysis of form (1f) is the following structure:

(4)



[2] We reiterate here our debt to Raimy, Frampton and Halle.

The linear order of the output form (1f) is read off of the terminal nodes of the tree (4). Neither copy, the lefthand or the righthand, has priority as Base to which the other is affixed—there are just two copies of the input. The question of whether (1f) is an example of prefixation or suffixation evaporates. Looking ahead, once we have structure, we can potentially refer to the structure in formulation of phonological rules.

Two related issues require immediate clarification:

- How is the domain of reduplication delimited
- What triggers the projection of a structure as in (4)

We propose that reduplicative *morphology* involves the insertion of elements into a morphological structure that are interpreted by the *phonology*. These elements are interpreted in such a way that they delimit domains of projection and specify the nature of those projections. For a form like *patikupatiku*, we assume an input bounded by square brackets:

(5) Output of the morphology and input to the phonology: [patiku]

Assuming that /patiku/ is a morpheme, the reduplicative morphology inserts a left bracket at the left edge of the morpheme and a right bracket at the right edge. The matching brackets have the following interpretation:

- Material between brackets constitute a domain we call Dup-domain
- Project the structure between brackets into a branching structure, once to the left (L-Proj) and one to the right (R-Proj)

In literature that makes use of the notions of Base and Reduplicant, the Base typically is identified with a root morpheme of a complex stem of root and affixes. In contrast, the Dup-domain, once defined by brackets is purely *phonological*, for our purposes a string of segments and brackets. This immediately brings us to a point of differentiation with the typology sketched above.

The brackets defining a Dup-domain can be inserted in various, phonologically defined positions in a morphological form. For example, the brackets can be inserted around the first syllable of a morphological input form, or around the last syllable:

(6) a. [pa]tiku
b. pati[ku]

The inputs in (6) yield the following projection trees:

(7) a. [pa]tiku
pa pa

b. pati[ku]
ku ku

The terminal strings of these trees are just linearized in place with the rest of the input, yielding the following:

(8) a. papatiku

b. patikuku

We can now see that once the Dup-Domain is defined in purely phonological terms, the forms in (6) can be understood as exactly the same as *patikupatiku*. The relevant structure is just double projection of an input, *s*:

(9) [s]

s s

The output is linearized as *ss*. The forms consist of two full copies of the Dup-Domain, there is no need to appeal to the notions of prefixation or suffixation. Of course *s* may be flanked by other material:

(10) x[s]y

s s

The output is linearized as *xssy*.

This last example suggests that the Dup-Domain is not necessarily located at the edge of morpheme. If the brackets are inserted, say around the penultimate syllable of a stem, then we get the following: pa[ti]ku.

(11) pa[ti]ku

ti ti

The output is linearized as *patitiku*. Since we have left morphology behind upon insertion of the brackets, there is no question of infixation here, we just have projected two copies of a phonological Dup-Domain, as before. The unavailibilty of morphological information inside the phonology is consistent with standard practice in generative phonology—in general, it is assumed to be preferable to find analyses that do not interleave information from the two domains, since a pure phonological account is more general than a mixed one. The following four forms now constitute a natural grouping of patterns, in our model without Base and Red:

(12) a. papatiku

b. patikuku

c. patitiku

d. patikupatiku

Of course, we can extend this to cases that define the Dup-Domain in terms of larger units, say disyllabic feet:

(13) a. [pati]ku ⇝ patipatiku
b. pa[tiku] ⇝ patikutiku

So, once we are into the phonology, many superficially divergent patterns reduce to the mechanism of double projection. The difference between infixation, prefixation and suffixation of other models disappears, as does the distinction between total and partial reduplication of *morphological* units, for the cases discussed thus far. These cases fall into Halle's 2008 category of "simple reduplication" which "involves the copying of a sequence of contiguous segments in a word" with the two copies concatenated with no intervening material. Halle explains (p.326) that "the repeated material is always a contiguous subsequence; except for being contiguous, however, the substrings do not [necessarily] possess well-recognized linguistic properties. For example, the [repeated substrings in his examples] are not coextensive with either the morphemes or the syllables that make up the word".

Clearly the cases in (1b,c), repeated in (14) will require additional machinery.

(14) Beyond square brackets

i. *kupatiku*—the Base *patiku* with the partial Red *ku* prefixed to the Base
ii. *patikupa*—the Base *patiku* with the partial Red *pa* suffixed to the Base

In section 2, we will develop that machinery, but demonstrate that such forms also make use of the square brackets invoked thus far.

In the remainder of this section, we provide some cases of reduplication that can be handled with just the mechanism of square brackets.

1.3 Total Reduplication in Kham

Kham (Watters, 2002, p. 148) provides an example of reduplication in which the Dup-Domain is coterminous with a morpheme—the morphology inserts square brackets at the edge of the morpheme.

(15)

Morpheme Type	Reduplicated Form	Gloss
Monosyllabic	kik-kik	'choking'
	phur-phur	'strung tightly'
	chõː-chõː	'squirting'
	sür-sür	'sour-like'
Bisyllabic	kutu-kutu	'in small pieces'
	pərəp-pərəp	'dripping'
	cherla-cherla	'ragged'
	zəhrãː-zəhrãː	'streaked'
Trisyllabic	kuturu-kuturu	'crispy, crunchy'
	phiriri-phiriri	'spinning'
	zigəra-zigəra	'with drooping eyelids'
	khopəlyak-khopəlyak	'tossing one at a time'

Of these forms, Watters notes that "exact[3] reduplication copies all consonants and vowels along with surprasegmental material, and can occur over one, two, or three syllables" (Ibid). Employing our schematic we can express the reduplicated form for 'choking' as follows:

(16)

[kik]

kik kik

As noted above, it is impossible for the analyst to determine whether the left or right copy is the original *base* in totally reduplicated forms. Our rejection of the notion of Base vitiates this question, but also, in our view is more consistent with the spirit of the generative program in that we do not derive surface forms (Red) from other surface forms (Base). Instead, we derive all parts of the surface output from underlying input forms. Just as the English plural [kæts] *cats* is derived from a root and suffix, not from the singular *cat*, the two parts of a reduplicated form like *cherlacherla* are derived from a root /cherla/—one part of the surface form is not derived from the other. This is an important point, and shall be developed further.

1.4 Samoan Infixation

If one accepts accepts as fundamental the difference between prefixing and infixing reduplication, then the following Samoan data (Broselow and McCarthy, 1983, p. 30) suggests that both forms can occur within a single language.

(17)

	Singular	Plural	Gloss
a.	taa	*ta*taa	'strike'
	tuu	*tu*tuu	'stand'
b.	nofo	*no*nofo	'sit'
	moe	*mo*moe	'sleep'
c.	alofa	a*lo*lofa	'love'
	savali	sa*va*vali	'walk'
	maliu	ma*li*liu	'die'

The forms in (17a) and (17b) appear to exhibit a *prefixing* of phonological material (a reduplicative morpheme RED is drawn from the initial CV of the Base and then prefixed); those in (17c) have traditionally been labelled as reduplicative *infixation* (where RED is drawn from a medial CV in the Base and then *infixed* to yield the reduplicated output).

This differentiation raises two important questions. First of all, are the forms in (17a) and (17b) really *fundamentally* different from those in (17c)? Clearly they are not. As many scholars have proposed, the unifying property of all the Samoan cases is that the onset and first mora of the penultimate syllable is being reduplicated. This is exemplified by the (Broselow and McCarthy, 1983, p. 53) analysis:

[3] 'Exact' is sometimes used in place of 'total' or 'full' reduplication. All three terms are synonymous in the literature.

> In Samoan, what appears to be internal reduplication is simply a special case of prefixation: since this language has penultimate stress, we claim that CV is prefixed to the stressed syllable — or equivalently, to the metrical foot — of the word. Then, as in other reduplications, the melody of the constituent to which the reduplicative affix is attached is copied, and association proceeds from left to right, as it normally does in prefixal association.

Secondly, in what sense can we say that the substrings in (17c) are truly infixed? If one accepts the existence of a Base and Red, then one is forced to explain how Red is aligned at the left edge of some Bases (prefixed) and inserted inside other Bases (infixed).

Under our Base-less model, there can be no infixation. The DUP-DOMAIN is just the phonological entity "penultimate syllable" which is where it is, at the left edge of the root in two syllable forms, and medial in longer forms. The phonological DUP-DOMAIN is not aligned with a morpheme or morphological word.

Consider an alternative presentation of the data in (17), where the phonological constituents that undergo duplication are highlighted in bold:

(18)

	Singular	Plural	Gloss
a.	**ta**a	*ta*taa	'strike'
	tuu	*tu*tuu	'stand'
b.	**no**fo	*no*nofo	'sit'
	moe	*mo*moe	'sleep'
c.	a**lo**fa	a*lo*lofa	'love'
	sa**va**li	sa*va*vali	'walk'
	ma**li**u	ma*li*liu	'die'

It appears that in Samoan all that is required to generate the plural forms in (18) is a *total* or *full* reduplication of well-defined *substrings* in the singular base (here, the segments in the penultimate syllable). Again, we needn't privilege the stem and explain modifications to its RED copy; rather, our production is akin to that in (10), applying to the bold substrings in the singular column to generate the requisite plural. In addition, there is no need for an *affixational* analysis to account for forms like *alolofa*: 'infixation' can be interpreted here as a descriptive term applied to an output that has been generated through full reduplication.

(19) (a) sa[va]li ⇒ savavali
va va

(b) [no]fo ⇒ nonofo
no no

In the following cases, the morphology again inserts a left and right bracket ([...]) with no intervening braces or angle brackets. Quite simply,

(20) MANGYARRAY **g[ab]uji** ‘old person(s)’

Juncture insertion rules:

- Insert a [juncture to the left of the timing slot linked to the first vowel of the word.
- Insert a] juncture to the right of the timing slot linked to the consonant directly preceding the second vowel of the word.

Derivation:

g-[ab]**-uji** ⇒ *gababuji*

ab ab

(21) AGTA **[bar]i** ‘(my whole) body’

Juncture insertion rules:

- Insert a [juncture to the left of the first timing slot of the word.
- Insert a] juncture to the right of the timing slot linked to the consonant directly following the first vowel of the word.

Derivation:

[bar]**-i** ⇒ *barbari*

bar bar

The difference between Mangyarray and Agta in the above examples rests with the juncture insertion rules — here, marking the DUP-DOMAIN. In Mangyarray (20), the left bracket juncture, [, is inserted relative to the first vowel of the word; in Agta (21), juncture insertion merely references the first timing slot. Similarly, the insertion of] for Mangyarray references the consonant preceding the second vowel of the word; Agta differs by employing a similar insertion relative to the *first* vowel. In both cases we observe full reduplication of the stem as the *structural* input to PROJECT is the same.

This example serves to demonstrate the modularity of our approach; morphology inserts the junctures and the phonology interprets them via projection.

1.5 Summary

In this section we have provided a basic schematic presentation of reduplication that does not rely on affixation of a Reduplicant to a Base. This allowed us to unify several disparate patterns as binary projection structures whose output is linearized with flanking material. In (§2) we formalise the *projection* operation and extend the model to account for cases where copies are non-contiguous on the surface.

2 The Full Scheme

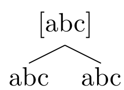
In this section, we outline the structure and operation of our approach.

2.1 The Reduplication Domain

Our model is built around the idea of *projection*. The morphology inserts junctures (braces, angle-brackets, brackets) which are gradually erased during the *spell-out* of the reduplicated form.

For Halle, these junctures are eliminated through special relinearization rules; here, they drive the function Project. Anything placed between square brackets ([...]) is part of a reduplication domain, referred to here as Dup-Domain. We posit Project as a universal spell-out rule, generating two branches, L-Proj and R-Proj from the contents of the Dup-Domain. These projected strings are subject to further projection and may contain additional domains and angle brackets specifying *partial* projection.

To begin with, consider the simple case of a Dup-Domain shown in (22):

(22)


⇒ *abcabc*

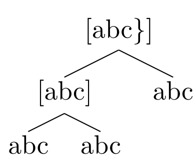
Here, Project is fed the string of segments [abc]; there are no junctures present in the input. As such, Project generates the left and right branches and terminates, yielding an output form of *abcabc* (the output of the projection is presented on the right hand side of the ⇒ arrow).

2.2 Braces as nested Dup-Domains

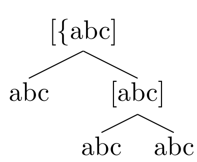
Inserted by the morphology with directional parameters, braces mark *nested projection*; Project will derive a secondary Dup-Domain in the course of iterative derivation.

A right brace, }, will project segments to its left into a new Dup-Domain in L-Proj. Similarly, material to the right of the left brace, {, is projected as a new Dup-Domain in the *right branch* (R-Proj).

The application of braces is shown below in (23):

(23) (a)


⇒ *abcabcabc*

(b)


⇒ *abcabcabc*

Here we see two possible derivations of a single brace in the input string. In the first case (23a), a *right* brace has been inserted at the end of the Dup-Domain, causing the substring to its *left* to be projected as a new Dup-Domain into the *left* branch. The right branch projects normally — cf. (22) for comparison.

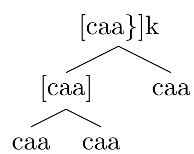
The tree in (23b) is the mirror image of (23a); in this instance, a *left* brace has been inserted at the beginning of the input string, causing projection of material to its *right* into the *right* branch, R-Proj. These two derivations are structurally ambiguous — they both generate triplication of the segment string in the initial Dup-Domain. This triplication is achieved *via* an initial branching followed by branching in one branch at the next level.

2.2.1 Triplication

Triplication in Mokilese occurs in progressive forms with monosyllabic roots. We cannot determine which of the structures in (23) corresponds to the Mokilese situation, so our choice of a right brace, }, inserted at the end of the Dup-Domain, is arbitrary:[4]

(24) Mokilese **[caa}]k** 'bending'

Derivation:



⇒ *caacaacaak*

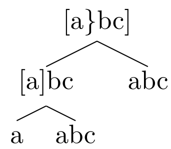
Once more, note that we exclude the segment *k* from the Dup-Domain by placing the brackets appropriately. The *phonology* of reduplication does not have to deal with the *k*—there is just full triplication from our perspective.

2.2.2 Partial Insertions

When a brace is not placed adjacent to the square bracket of the same directionality, it will not have scope over the whole initial Dup-Domain. We call this a "partial insertion" since it defines a new projection domain that is just part of the initial one defined by the square brackets. Consider the following partial brace insertions:

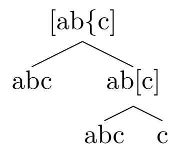
[4]The fact that we have no argument bearing on which structure in (23) is correct for Mokilese, is, of course, a weakness of our approach—like all current linguistic theories, we are faced with what we call "the problem of too many solutions". Addressing this problem is the subject of a current SSHRC research grant to Reiss.

(25) (a)



⇒ *aabcabc*

(b)



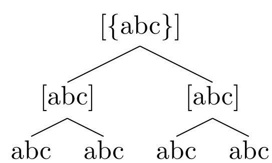
⇒ *abcabcc*

In (25a), Project rewrites the right brace, }, as a Dup-Domain in the left branch (L-Proj). R-Proj is projected as expected. Recursing, L-Proj is fed through the projection algorithm, duplicating the substring 'a' and adjoining the remainder of the node's segments — 'bc'. This yields 'a+abc' on this branch. Similarly, in (25b) L-Proj is projected as expected, while the right branch contains a new Dup-Domain that adjoins the substring 'ab'.

2.3 Quadruplication

Finally, consider the case of *full quadruplication* made possible by our juncture-driven projection:

(26)



⇒ *abcabcabcabc*

Both branches — L-Proj and R-Proj — interpret a brace to derive a new Dup-Domain. In the case of L-Proj, the right brace; in the case of R-Proj, the left. Since the junctures have been inserted at the beginning and end of the string and enclose *all* of the input segments, full reduplication occurs in both branches, generating a quadruplicated output form. Quadruplication is thus generable in our system, and it appears to be necessary for modeling real languages. The brackets that generate this pattern constitute, in our system, a single morpheme, and this view is consistent with the interpretation of the cases we have found.

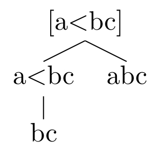
2.4 Angle Brackets — Substring Projection

Angle brackets are inserted by the morphology to specify substring projection. Anything to the *right* of a *left* angle-bracket (<. . .) will project in the *left* branch, L-Proj (i.e., material on the concave side of the bracket projects).

Similarly, anything to the *left* of a *right* angle bracket (. . . >) will project in the *right* branch, R-PROJ.[5]

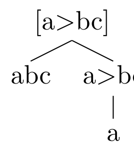
Consider the examples in (27), illustrating the two brackets individually and then the two together:

(27) (a)



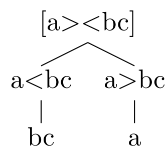
⇒ *bcabc*

(b)



⇒ *abca*

(c)



⇒ *bca*

We now have the machinery to account for the remaining forms from (1), the ones that cannot be treated as just projection of a phonological sequence.

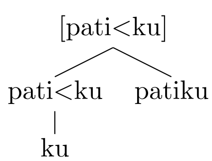
Here are the forms:

(28) a. kupatiku

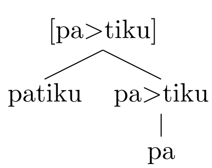
b. patikupa

Here are inputs and derivations of these forms:

(29)



⇒ *kupatiku*



⇒ *patikupa*

[5] Note that our definitions of < and > are essentially complements of those presented in Halle (2008)—we want it to always be the case that the material *inside* the bracket projects, for all the bracket-types.

We now show actual language data using angle brackets, but no braces. The Madurese and Arabic each use a single angle bracket, the Tigre intensive example uses both brackets.

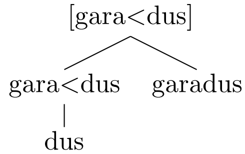
(30) MADURESE **[gara<dus]** 'fast and sloppy'

Juncture insertion rules:

- Insert a] juncture to the right of the (timing slot linked to the) last stem segment.
- Insert a [juncture to the left of the (timing slot linked to the) first stem segment.
- Insert a < juncture to the right of the (timing slot linked to the) onset of the last stem syllable.

Derivation:

[gara<dus] ⇒ *dusgaradus*



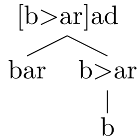
(31) LEVANTINE ARABIC **[b>ar]ad** 'shaved evenly'

Juncture insertion rules:

- Insert a [juncture to the left of the first stem segment.
- Insert a [juncture to the right of the penultimate stem segment consonant.
- Insert a > juncture to the left of the penultimate stem vowel.

Derivation:

[b>ar]ad ⇒ *barbad*



(32) TIGRE (intensive) **m[sl><a:]** 'be diplomatic'

Juncture insertion rules:

- Insert [juncture before the penultimate stem consonant;
- Insert] juncture after the a: suffix;
- Insert > juncture after the last stem consonant
- Insert < juncture before the /a:/ suffix.

Derivation:

```
    m[sl><a:]          ⇒   ma:sl
      /   \
  sl<a:   sl>a:
    |       |
    a:      sl
```

Notice that the effect of the two angle brackets in Tigre is the *appearance* of metathesis. This occurs because only material from the right edge of the Dup-Domain surfaces in the L-Proj and only material from the left edge of the Dup-Domain surfaces in the R-Proj. This is *not* phonological metathesis, which involves switching the order of the unique tokens of two segments that occur in the input of a phonological rule.

2.5 Junctures aren't mandatory

Angle brackets are not *mandatory* — notice that in (27a) there is no > present in the input; in (27b), < is absent. The following inputs in (33–34) therefore yield the same output:

(33)

```
     [abc]            ⇒   abcabc
     /   \
   abc   abc
```

(34)

```
    [<abc>]           ⇒   abcabc
     /   \
  <abc   abc>
    |      |
   abc    abc
```

Note that in (34), junctures have been inserted by the morphology at the edges of the Dup-Domain. We shall omit this (redundant) insertion, instead requiring that Project iterate until all junctures have been spelled out. At the top-level of Project, braces are deleted (that is, interpreted) in the course of branching. In each branch, angle brackets are then interpreted if present, conditioning the output of terminal nodes Therefore, (33) does not require < and > edge markers as its branches have already been maximally projected.

The same holds for the braces — note their absence in (33) and (34). We address this the following section.

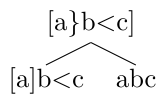
3 Complex Inputs

3.1 Interacting Junctures

We now consider some schematic forms that can be derived by combining the primitives of the system developed thus far.

Consider the following form:

(35)

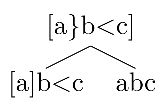


Since there are no junctures interpretable in the right branch, R-PROJ is simply *abc*. L-PROJ is more complicated; from the input [a}b<c], we need to do three things:

1. Given }, reduplicate (re-project) *a* as [a].
2. Project everything to the right of <.
3. Write out *c*.

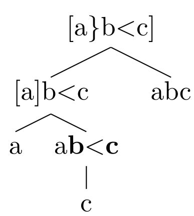
These operations are *ordered* — re-projection occurs before substring projection. Note that in (36), < is still present in the left hand branch:

(36)



Continuing projection in the left branch, two more branches are created for *a*. The remaining material is adjoined to the relevant branch — in this case, b<c is to the *right* of [a], therefore it adjoins the right branch. This is represented below; the diagram depicts the state of the intermediate representation following *secondary projection* — angle-bracket projection can now apply, exhausting this branch.

(37)



⇒ *acabc*

As expected, the secondary DUP-DOMAIN [a] branches. It is important that we stipulate that the remainder of the string (shown in boldface) attaches to the *rightmost* branch of the projection. The final stage is the partial single projection of the material inside the angle bracket, that is, just c.

Put another way: PROJECT is a recursive function[6]. When a string contains braces, these are resolved on a lower tier and the projection function re-applies. Crucially, material not in a DUP-DOMAIN attaches to *one or the other branch*, depending on the original flanking.

3.2 Two case studies

The *frequentive* in Tigre exhibits reduplication of the penultimate consonant in addition to the partial projections outlined above for the intensive. As such, we require an additional juncture insertion rule of the following form:

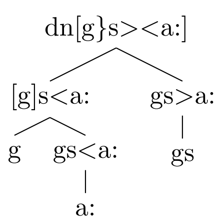
(38) *Tigre*: In the frequentive, insert a } after the penultimate stem consonant.

Incorporating (38) into the insertion rules in (32), projection proceeds as follows,

(39) TIGRE (frequentive) **dn[g}s><a:]** 'become slightly scared'

- Insert [juncture before the penultimate stem consonant;
- Insert] juncture after the a: suffix;
- *Insert } juncture after the penultimate stem consonant*;
- Insert > juncture after the last stem consonant
- Insert < juncture before the /a:/ suffix.

Derivation:



⇒ *dnga:gs*

The benefit of our approach becomes immediately apparent when we consider Temiar. Consider the following data (Broselow and McCarthy, 1983, p. 39):

Temiar Active Verbs	Bi-C Root	Tri-C Root
Perfective	kɔ̄w	slɔg
Simulfactive	kakɔ̄w	salɔg
Continunative	kwkɔ̄w	sglɔg

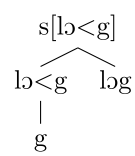
[6] Where L-PROJ and R-PROJ have the option to recurse in the course of spelling out their junctures.

Broselow & McCarthy (1983) focus on the active continuative triconsonantal case where "a copy of the root-**final** consonant is lodged to the immediate left of the root medial consonant. Thus, *slɔg* becomes *sglɔg*". They present this case as problematic for standard theories of reduplication, and they explain the phenomenon as

> infixation of an underspecified morpheme into the CV skeleton [which] induces copying of the phonemic melody of the root. Because this copy of the phonemic melody appears on a different tier from the root melody itself ... both ends of the copy are accessible to application of the association procedure. Since infixes are not subject to the prefix/suffix rubric for direction of association, the direction must be stipulated for each infix.... [Broselow & McCarthy (1983: 40)]

In our model, these phenomena can be accounted for without recourse to underspecified morphemes. We can treat *g* as the result of partial projection in the left branch and *-lɔg* as a fully projected right branch. There is no Base and no infixation, just concatenation. This is a general result—there is no infixing reduplication under our conception. The solution is sketched in (40):

(40)



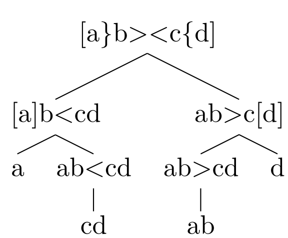
⇒ *sglɔg*

While this form, *sglɔg* reduplicated from an root /slɔg/, appears to exhibit a degree of complexity on the surface, it actually has the exact same structure as Madurese *dusgaradus*, reduplicated from /garadus/ in (30) — that is to say, we only require a single juncture for skipping. The difference is that in this Temiar case, the root-initial *s* is not in the Dup-Domain. To re-iterate: the reduplication domain is defined with reference to phonological, not morphological, boundaries. As such, Temiar can be accounted for without recourse to the complexities suggested by McCarthy & Broselow.

3.3 Greater complexity

With this simplification in mind, consider the following more complex schematic case, containing all the junctures:

(41)

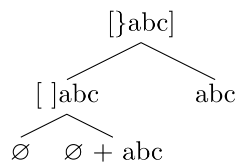


⇒ *acdabd*

This form represents the maximally complex reduplication form—it contains one brace for each direction and one angle bracket for each direction. As far as we can tell, competing models like that of Idsardi & Raimy (in progress), which model reduplication via the addition of precedence links, do not provide for an upper limit on the complexity of reduplication representations like this. Our model provides for a DUP-DOMAIN defined by square brackets, along with the possibilty for the other four junctures—and that's all. Whether or not this is ulitmately sufficient, the model at least provides an explicit, testable hypothesis.

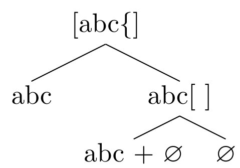
Finally, for the sake of explicitness, we illustrate using braces, that vacuous bracketing is possible, but that we will assume that brackets can be left unused. A parallel demonstration can be made for angle brackets. The following diagrams show the derivations for [}abc] and [abc{]:

(42) (a)



⇒ *abcabc*

(b)



⇒ *abcabc*

Since the scope of each brace is delimited by the facing square bracket, there is no segmental material subject to a second round of projection in these cases. It therefore follows that inputs [abc], [abc{], [}abc] and [}abc{] all lead to the same surface string—the string is technically structurally ambiguous, but since these cases involve vacuous projections, we will ignore them.

4 Fixed Segmentism

This section builds on Halle's analyses of fixed segmentism. The Optimality Theoretic analyses explains these effects through an 'imperfect copying' analysis, a mismatch of Base and Reduplicant:

> Reduplicative morphemes copy the base to which they are attached, but perfect copying is not always achieved. Incomplete copying for templatic reasons — that is, partial reduplication — has received much theoretical attention. Less has been said about cases where perfect copying is subordinated to fixed segmentism: invariant segments (or tones or features) that appear where copying might have been expected. [Alderete et. al, 1999]

Unsurprisingly, our analysis differs markedly, building on the ideas of structure and projection outlined above. We have already shown that the reduplication domain can be smaller than a morpheme (20,31); here we show that it can also be larger.

In Yoruba, nominalization involves 'the reduplicative morpheme [having] the fixed vowel *i*, whatever the vowel of the base. In the Kamrupi echo-words . . . the initial consonant of the reduplicative morpheme is replaced by fixed *s*' (Alderete et. al, 1999). Forms illustrating these patterns include the following:

(43) *Yoruba (Akinlabi 1984, Pulleyblank 1988)*

Stem	Reduplicated Form	Stem Gloss	Reduplicant Gloss
gbóná	gbí-gbóná	'be warm, hot'	'warmth, heat'
dára	dí-dára	'be good'	'goodness'

(44) *Kamrupi (Goswami 1955-6: 164)*

Stem	Reduplicated Form	Stem Gloss	Reduplicant Gloss
ghara:	ghara:-sara:	'horse'	'horse and the like'
khori	khori-sori	'fuel'	'fuel and the like'

So, the Yoruba forms *gbí-gbóná* and *dí-dára* both have the fixed segment *i*, and the Kamrupi forms *ghara:-sara:* and *khori-sori* both contain the fixed segment *s*.

4.1 Yoruba analysis

This type of form can be accounted for with ease given our scheme. The Yoruba case in (43) involves the following juncture insertion rules[7]

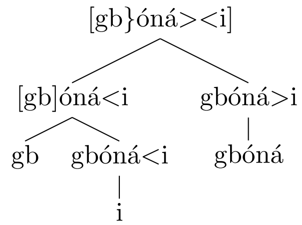
1. Insert a left bracket at the left edge of the root and right a bracket at its right edge;
2. Insert a } juncture after the onset;
3. Insert a > before the end of Dup-Domain (before]);
4. Insert a < before the end of the Dup-Domain, following <;
5. Insert templatic /i/ before the].

Given an input of *gbóná*, these *morphological* rules yield the following:

(45) gbóná ⇒ [gbóná] ⇒ [gb}óná] ⇒ [gb}óná>] ⇒ [gb}óná><] ⇒ [gb}óná><i]

[7]We note that an alternative (and equally viable) analysis for this case of fixed segmentism involves the more restricted reduplication domain and a single juncture—[i<gb]óná. Once again, we suffer from a surfeit of solutions, indicating that our theory, like all others, leaves many questions open.

To derive the reduplicated form, this underlying representation is projected by the *phonology* as follows:

(46) 

⇒ *gbígbóná*

Note that the Dup-Domain is larger than a morpheme, since it contains the root and the fixed segment.

4.2 Kamrupi analysis

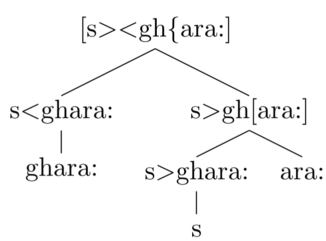
Turning to the Kamrupi case in (44), we require the following insertion rules:

1. Insert a left bracket at the left edge of the root and right a bracket at its right edge;
2. Insert a { juncture before the first vowel of the root;
3. Prepend < (insert < after the left bracket that begins the Dup-Domain);
4. Prepend >
5. Prepend the 'replacement consonant' /s/.

With an input of *ghara:*, the application of these morphological insertion rules yields:

(47) ghara: ⇒ [ghara:] ⇒ [gh{ara:] ⇒ [<gh{ara:] ⇒ [><gh{ara:] ⇒ [s><gh{ara:]

With this input, Project generates the following in the phonology:

(48) 

⇒ *ghara:sara:*

Note that in both the Kamrupi and the Yoruba we use the same mechanism needed for triplication in Mokilese (24), a brace that generates a nested Dup-Domain. In Yoruba, this nested domain is in L-Proj, as in Mokilese, since both langauges used a *right* brace, }. Despite the triplication, only two copies of the *gb* surface, since one is on the "wrong side" of an angle bracket. In Kamrupi, the nested DUP-DOMAIN is in the R-PROJ, since a *left* brace, {, is used. Despite the triplication of *ara:* in the intermediate representation, only two copies surface due to the effect of the angle bracket.

5 Explaining Under- and Over-application derivationally

The model we have developed is derived closely from work by Raimy, Frampton and Halle, as noted above. All three of these authors work in a derivational framework that does not avoid positing intermediate representations between the input, underlying representation, and the output, surface representation, of the phonology. For the most part, these models rely on a procedural model of the phonology as a complex function that can be decomposed into individual input-output mappings (simple functions) that are usually referred to as "rules". In our projection trees, the levels of the tree correspond to a sequence of representations from an input UR to an output SR.

Various versions of Optimality Theory (OT) have been proposed over the past fifteen years or so as an alternative to derivational, rule-based models, beginning with Prince & Smolensky (1993) and McCarthy & Prince (1993). There are numerous versions of Optimality Theory, but the core idea is that grammars consist of ranked, violable constraints that evaluate in parallel a universal set of competing output candidates. In most versions of OT, there is an input form and from among the infinite set of output candidates, the grammar selects a single "optimal" form that is the grammar's output for that input.

There is no question of the sociological success of OT—many prominent phonologists do, or at some time in the past did, self-identify as practitioners of OT.[8] However, the scientific value of OT as a research program remains a topic of debate, not least because of the splintering of the OT community into various mutually incompatible factions—functionalists *vs.* formalists; discretivists *vs.* gradientists; determinists *vs.* stochasticists; *etc.* In light of the continuing lack of a clear paradigm arising from the basic ideas of OT, it behooves us to consider a group of phenomena that was touted as very strong support for a model based on parallel constraint evaluation, and as contraindicating derivational models. These phenomena, known as "backcopying" effects in reduplicated forms, convinced many working phonologists, at least temporarily, of the superiority of the OT approach. In this section, we demonstrate that our model (like Raimy's derivational approach before us) is, in fact, able to handle "backcopying" in reduplication, thus demonstrating that parallel models like OT are not proven *necessary* by the existence of such phenomena. In addition, the validity of the backcopying analyses of the data will be briefly discussed.

Consider the Malay data in (49)—we assume, following discussion in the

[8]This is clearly an impressionistic claim—we have no statistics about the matter.

literature, that the reduplicated forms are synchronically derived from the same roots as the non-reduplicated ones:

(49) Overapplication in Malay

Simple		Reduplicated	
aŋãn	'reverie'	ãŋãn-ãŋãn	'ambition'
aŋẽn	'wind'	ãŋẽn-ãŋẽn	'unconfirmed news'

For the most part, nasalized vowels occur in Malay only as predictable variants of non-nasalized ones, when the vowel immediately follows a nasal consonant. As expected, the initial vowel of the simple forms in (49) is not nasalised, but, surprisingly, the initial vowels of the reduplicated forms are nasalised, even though these vowels are not preceded by a nasal consonant. Informally, this appears to be a case of overapplication of the nasalisation process, and it seems plausible that the irregularity of nasalisation on the vowel is somehow related to the fact that the other copy of the root-initial vowel is nasalized (by regular application of the process).

McCarthy and Prince (1995:119, henceforth MP) insightfully remarked on the importance of such cases for theory comparison:

> Crucial evidence distinguishing serialist from parallelist conceptions is not easy to come by; it is therefore of great interest that reduplication-phonology interactions supply a rich body of evidence in favor of parallelism.

Not surprisingly, MP argue that these cases provide strong support for non-derivational models like Optimality Theory. They assume that these Malay forms consist of a Base, B, followed by a reduplicant, R. They called the process 'backcopying', since B appears to create the nasalization environment for the initial vowel of R, but then the effects of this environment, the nasalization on the vowel, are copied or reflected back onto B. MP claim that "The most familiar theories—those with fixed rule ordering are incapable of expressing patterns in which R imposes phonology on B that then re-appears in R" (McCarthy and Prince 1995:118). Note that if we reduplicate before the nasalization applies, we expect a form like **aŋẽn-ãŋẽn* with one (incorrect) non-nasalized vowel. However, if we apply nasalization before reduplication, we derive, first *aŋẽn* then *aŋẽn-aŋẽn*, with two (incorrect) non-nasalized vowels. Neither ordering works.

MP suggest that such cases justify the adoption of Correspondence constraints, an extension of OT's standard identity constraints, the ones which are satisfied by identity between the input to the phonology and candidate output forms. The broad array of Correspondence constraints proposed by MP extends the demand for identity to relations between other kinds of elements, such as the B and R.[9] Thus, the nasalisation on the initial vowel of the reduplicated

[9]Notice that this model requires that morphological information be present in candidate output forms—the grammar has to be able to identify B and R in candidates. This is a

forms in (49) satisfies a BR Correspondence constraint that would otherwise be violated.

MP point out that over- and underapplication fall out naturally from a model endowed with constraints demanding identity between B and R:

> For the theory of reduplicative phonology, the principal interest of the architecture proposed here is this: the phenomena called over-application and underapplication follow in Correspondence Theory from the very constraints on reduplicant-base identity that permit reduplication to happen in the first place. The constraints responsible for the ordinary copying of a base also govern the copying of phonologically derived properties. (McCarthy and Prince 1995:7)

Raimy (2000) provided the first refutation of MP's strong claim about derivational models by treating reduplication as the introduction of additional precedence relations in the phonological string. However, just as basic details of Correspondence Theory remained unsolved (Idsardi and Raimy, 2008), Raimy's system never was made fully explicit, although recent work (Raimy and Idsardi, forthcoming) appears to address the issue.

Since we do not make use of the notions Base and Reduplicant, as laid out in the Introduction to this paper, our account of backcopying will have to look very different from that offered by MP. We argue that backcopying is not only non-problematic for derivational models, but that it offers strong evidence *in favor* of the derivational model we developed in this paper based on PROJECT. For the Malay case we can informally indicate here the form of the argument presented below: we need a derivation to create the environment for nasalization of the initial vowel in the reduplicated forms, apply the nasalization, then remove the material that created the environment.

In the next section, we briefly present some salient features of the MP and Raimy models, then in §5.2 we flesh out a case of backcopying in Akan that is referred to as 'underapplication'. We then return to the case of overapplication in Malay in §5.3. We will see that all the machinery needed for these cases has already been introduced in our model of reduplication sketched above.

5.1 Theoretical Background

5.1.1 Base Reduplicant Correspondence Theory (BRCT)

McCarthy and Prince (1995) propose an approach to reduplication that builds on the idea of the identity relation between base and reduplicant;

---

general property of many OT models, for instance those using alignment constraints that demand coterminous syllables and morphemes. This requirement for morphologically rich surface forms, as the output of the *phonology* is in stark contrast with traditional derivational phonology, in which morphological structure is assumed to be inaccessible once a form is passed to phonology. This property of some OT models, along with the seeming contradictory property of encoding fine phonetic detail in surface forms found in other models, was pointed out to (one of) us by Mark Hale around 1995, but seems to be ignored in the OT literature. There is some discussion in Hale and Reiss (2008), chapter 6.

> Reduplication is a matter of identity: the reduplicant *copies* the base. Perfect identity cannot always be attained; templatic requirements commonly obscure it. (McCarthy and Prince, 1995, p. 1)

This analysis follows from the theory that reduplication involves the construction of an output from two discrete morphological entities: the base and a reduplicant morpheme (RED). This is clear in the quotation above — the idea of the reduplicant *copying* the base reveals the asymmetry in the input; base is logically prior. As noted above, with the adoption of such an architecture constraints on reduplicant-base identity are fundamental in explaining these over- and under-application cases. The scheme outlined in this paper rejected the idea of an affixal morpheme driving duplication — commonly, RED or R— and instead proposed that reduplication is a result of *derivation* via *projection* from a morphologically rich underlying form.

For us, reduplication is a matter of identity only insofar as there is only ever one string of segments in the input. We have already argued that the notions of 'base' and 'reduplicant' are unnecessary. We will strengthen our claim by showing that even so-called 'back-copying' phenomena do not require Base and Reduplicant as elements of the theory, with affixing between them, and the morphologically rich surface forms they entail.

Such a claim necessitates a discussion of 'backcopying' effects that McCarthy and Prince claim cannot be handled in serial models. We provide derivational analyses of backcopying in Malay (§5.3) and Akan (§5.2), demonstrating the falsity of that claim and critiquing BASE-REDUPLICANT CORRESPONDENCE THEORY with the aid of Raimy (2000).

5.1.2 Reduplication as Affixation

We aim to demonstrate that even seemingly complex cases of reduplication are best situated within a derivational paradigm. These goals are well aligned with those presented by Raimy (2000):

> Reduplication will be shown to result from general properties of phonology and morphology and more specifically to be the result of the interaction between these two modules of grammar. (Raimy, 2000, p. 2)

Raimy proposes that reduplication results from *explicit precedence* in phonological representations, a relation that is asymmetrical, transitive and irrelexive. With enriched representations, reduplication becomes a special case of affixation. Affixation is, in turn, defined as the addition of new precedence links by the morphology.

'Total' reduplication is then elegantly represented as the addition of a precedence link (arc) from the segment preceding the end of string marker to the segment following the beginning of string marker. Total reduplication in Indonesian is represented as in (50):

(50) # → b → u → k → u → % ⇒ # → b → u → k → u → %

Note the extra link from /u/ to /b/; this form will be linearized as [buku-buku].

Following Halle (2008) we adapt Raimy's insights to our junctures and projection model.[10]

5.2 Backcopying in Akan ('Underapplication')

In this subsection we provide an analysis that combines the use of braces, angle brackets and fixed segmentism to account for reduplicated forms that exhibit rule under-application. Akan exhibits apparent backcopying effects, yielding complex patterns that are supposedly difficult to account for in a Derivational framework. As noted above, MP argue that Correspondence Theory provides the superior theoretical analysis and thereby constitutes an argument against serial models of reduplicative phonology.

Raimy, however, offers an elegant derivational solution, demonstrating that with modification to our understanding of phonological representations, backcopying can be accounted for with ease. This theory requires an interesting use of quantificational logic in phonological computation, a move that is not at all unattractive to us, (see Reiss (2003), for another use of quantification in phonology), but this strong claim provides good reason to think that our proposal, which does not rely on quantification in these contexts, is more than just a notational variant of Raimy's—both are worthy of further study, in our opinion.

Raimy (2000) sketches the background to the Akan problem thus:

> McCarthy and Prince (1995:340–345) present the interaction of palatalization and reduplication in Akan as a case of underapplication of a phonological process. This example is interesting because palatization does occur in some reduplicative forms. The prediction of when the palatization rule should underapply and when it should apply normally is the issue at hand. (Raimy, 2000, p. 18)

5.2.1 Akan Data

In Akan, velars and [h] for the most part do not occur before the mid and high front vowels: instead of [k, g, h], we find [ʨ, ʥ, ç], respectively. Consider the data in (51):

[10]Bill Idsardi has pointed out, we think correctly, that our approach relies on the precedence relations that Raimy's system makes explicit. He questions why we would want to enrich our scheme with junctures, instead of trying to derive everything from precedencearcs. It may be that our projection computations can be recast in Raimy's terms, however, we have noted above that our scheme provides a clearly limited range of possibilities for projection—there are the square brackets that define aDup-Domain and the other four optional junctures. We see this restrictiveness as a strength. In contrast, Idsardi and Raimy propose a less restricted model that can generate patterns that we do not attempt to model, including language game phenomena.

(51)	(a)	ʨɛ	*kɛ	‘divide’
	(b)	ʥe	*ge	‘receive’
	(c)	çɪ	*hɪ	‘border’

McCarthy and Prince propose that [dorsal] segments ‘are prohibited from preceding non-low front vowels (i/ɪ or e/ɛ)’ (Raimy, 2000, p. 19). They claim that [coronal] spreads ‘from the vowel onto the preceding [dorsal] segments’ resulting in the palatals that appear in (51).

The problem with this generalisation becomes apparent when we turn to reduplicated forms, shown in (52):

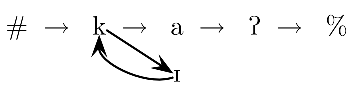
(52)	(a)	kɪ-kaʔ	*ʨɪ-kaʔ	*ʨɪ-ʨaʔ	‘bite’
	(b)	hɪ-hawʔ	*çɪ-hawʔ	*çɪ-çawʔ	‘trouble’
	(c)	ʥɪ-ʥe	*gɪ-ge		‘receive’

Why aren’t the initial consonants of *kɪ-kaʔ* and *hɪ-hawʔ* palatalized?

According to Raimy (2000, p. 19), Akan reduplicates C with a ‘fixed’ segment V which is prespecified as [+high]. Other features are determined by the following vowel in ways that will not concern us—we are only interested in forms with a high, front vowel /ɪ/ in the CV ‘reduplicant’.

If the segments [ʨ, ʥ, ç] are derived from underlying [k,g,h] before a non-low, front vowel, then palatalization apparently *underapplies* in (52a-b). To explain this phenomenon, Raimy proposes the following representation:

(53)



In (53), the underlying /k/ precedes two vowels: /k → ɪ/ and /k → a/. Building on the earlier generalisation about distribution of palatal consonants, Raimy claims that Akan has a Uniformity Parameter set to *on*. This means that the satisfaction of the ‘precedes non-low front vowel’ environment must be satisfied *for all* precedence relationships (i.e., universally quantified).[11] In other words, palatalization only applies if *every* segment following [k, g, h] is a non-low front vowel. In reduplicated Akan forms like (52a,b) this condition is not met—the /k/ and the /h/ do precede /ɪ/, but they also precede /a/. However, in (52c), the presumed underlying /g/ precede /ɪ/ and /e/, both of which are non-low front vowels. Thus palatalization applies.[12]

[11] Rather than viewing this condition as a language-wide parameter, it could also be built into individual phonological rules as part of their structural description.

[12] In the remaining discussion we adopt the analyses of McCarthy & Prince and Raimy according to which Akan exhibits underapplication. However, we are tempted to suggest that the reduplicative vowel is actually underspecified for [±back] when palatalization applies and this the ‘fixed’ segment is not a non-low front vowel at the appropriate point in the derivation. This means that Akan is actually exhibiting *overapplication* in cases like (52c) ʥɪ-ʥe from *gɪ-ge; forms like (52a) kɪ-kaʔjust show failure of overapplication, not underapplication, under this view. We think this is a promising line of exploration, one that even more strongly supports a derivational view, but we pursue the underapplication analyses here for the sake of explaining our projection scheme, and to demonstrate that anuncerapplication analysis does not force us into a parallelist model, contra McCarthy & Prince.

5.2.2 Project: dealing with underapplication

Given the representation in (53) and the data in (52), it seems reasonable to pose two questions:

- How can we deal with under-application while maintaining the projection model?
- Can we dispense with Raimy's universal quantifiers?

Since our explanation for reduplication in Akan centres on the insertion of a reduplicative vowel, represented here as /ɪ/, the morphology needs to insert this segment alongside duplication junctures. We propose the following rules for Akan:

(54) Akan reduplication morphology

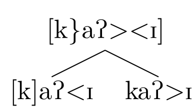
1. Insert a left bracket at the left edge of the root and right a bracket at its right edge;
2. Insert a left brace after the initial segment in the Dup-Domain
3. Insert the right angle bracket, >, before the right bracket;
4. Insert the < after the right angle bracket;
5. Finally, insert the reduplicative vowel, /ɪ/, after <.

Employing these rules, (52a) is initially derived as follows:

(55) kaʔ ⇒ [kaʔ] ⇒ [k}aʔ] ⇒ [k}aʔ><] ⇒ [k}aʔ><ɪ]

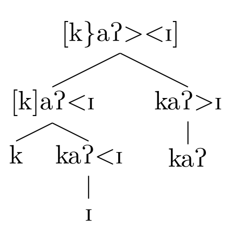
Running this input through the first round of Project yields (56):

(56)



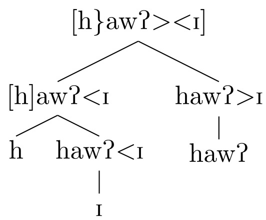
At this stage, we can try to apply the palatization rule: in both branches, /k/ occurs before /a/ — spreading of [coronal] does not occur. The full projection is shown below:

(57)



Importantly, the adjacency of /k/ and /ɪ/ in the output (kɪ-) is *not* present during its projection. The /k/ that manifests as the initial segment of the output form is from the secondary L-Proj — ‘kaʔ’ is not projected and ɪ projects in the final phase. Similarly, for (52b), ‘hɪ-hawʔ’:

(58)



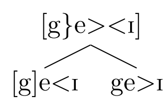
⇒ *hɪhawʔ*

We can now explain (52c) without recourse to universal quantification. Inserting our junctures and reduplicative /ɪ/:

(59) Akan reduplicative morphology: ge ⇒ [ge] ⇒ [g}e] ⇒ [g}e><] ⇒ [g}e><ɪ]

This form is fed to the phonology:

(60) First application of Project

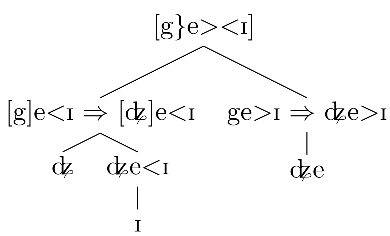


At this stage, we try to apply the palatization rule and succeed wherever *g* appears before a front vowel. Note the *g . . . e* environment in both branches —

(61)
[g]e<ɪ ⇒ [ʥ]e<ɪ
ge>ɪ ⇒ ʥe>ɪ

Finally, the full projection for /ge/ is presented below:

(62)



⇒ *ʥɪʥe*

It is *not* the reduplicative /ɪ/ that triggers this palatization — rather, it is the adjacency of /g/ and non-low front /e/. In Raimy’s model, the fixed segment /ɪ/ does not cause palatalization since the preceding consonant must occur only before an appropriate vowel along all paths for the process to apply. In our model, the fixed segment cannot cause palatalization since it is not adjacent to the consonant at the relevant point in the derivation. Although we are not able to choose between these alternatives on *a priori* grounds, we think it is useful to note that they differ in significant ways—even if they generate the same set of output forms, the models are only weakly equivalent, since the relevant factors (non-uniformity and non-adjacency at a point in the derivation) only accidentally coincide.

An examination of Akan therefore demonstrates that "underapplication" can be accounted for with a simple combination of the primitives of our system — morphological junctures, Project and a simple rule requiring adjacency of segments in order to apply.

The junctures were independently motivated by diverse phenomena such as infixation, triplication and partial reduplication. As such, we observe that back-copying is *not* a new phenomenon in need of additional theoretical machinery for its explanation. Rather, reduplicative backcopying results from the interaction of components independently needed.

5.3 Backcopying in Malay ("Overapplication")

We have already addressed an instance of rule *underapplication* in Akan, and shown that a derivational model built on the Project function can account for the data. This proposal was motivated by the desire to provide an alternative to Raimy's analysis while acknowledging the insight of his approach. Raimy (2000) also provides an analysis of the overapplication in Malay reduplication presented above, again countering the McCarthy and Prince (1995) claim that 'a derivational model of reduplication is completely unable to account for backcopying effects'. We repeat the data here:

(63)	(a)	aŋãn	'reverie'	ãŋãn-ãŋãn	'ambition'
	(b)	aŋẽn	'wind'	ãŋẽn-ãŋẽn	'unconfirmed news'

Raimy proposes the following representation for the form ãŋẽn-ãŋẽn.

(64) # → a → ŋ → e → n → %

In (64), → encodes precedences relationships between segments. Crucially, the /n/ *precedes* the initial /a/. /a/ therefore sits in two environments: [# → a] and [n → a]. This is represented by the Loop arc. We can understand the application of nasalization in Malay in terms of existential quantification—if on *any* path, a vowel is immediately preceded by a nasal, then all instantiations of the vowel are nasalized on the surface. The graph therefore linearizes as follows, generating the desired output form found in (63b),

(65) # → ã → ŋ → ẽ → n → ã → ŋ → ẽ → n → %

This analysis employs a Loop concatenator and non-linear ordering of the underlying representation to generate the correct output form in a derivational model. We now turn to account for the Malay pattern via projection.

5.3.1 'Backcopying' in the PROJECTION Scheme

There are two things that we know about the Malay cases in (63a) and (63b):

i. They *fully* reduplicate;

ii. They exhibit backcopying behaviour.

The following proposal builds on Raimy's insight that the word-initial surface /a/ *is in an environment preceded by a nasal.* Unlike Raimy, we posit linearly ordered segments in the underlying forms — this requires us to rethink the claim in (i).

As outlined above, we suggest that junctures inserted into the input by the morphology trigger recursive projection of segmental material in the DUP-DOMAIN, yielding (re)duplication.

Our first proposal is the following: /aŋen/ is not *fully reduplicated* but rather *partially triplicated.*

The junctures are inserted as follows:

(66) [{aŋen<]

This left brace marks the projection of a new DUP-DOMAIN, *aŋen*, into the right branch ("project a new DUP-DOMAIN into R-PROJ with all segmental material to my right"). The left angle bracket will be maintained in the *left* branch; everything that follows it will be projected to yield the terminal node; material on its convex side will fail to be projected into the output.

After one step of PROJECT we have the following:

(67)

[{aŋen<]

aŋen< [aŋen]

L-PROJ contains the string 'aŋen<'; R-PROJ contains '[aŋen]' — note that we have a secondary DUP-DOMAIN that will project its own left and right branches.

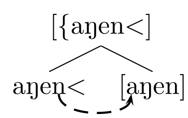
Our second proposal is that rules can be triggered after *each* projection cycle[13]. In other words, in (67), segmental rules apply between the segments (and through the junctures) in L-PROJ and R-PROJ—the string formed by concatenating the two projections is just an intermediate representation, in the standard sense in derivational phonology. . Consider,

(68) aŋen< ... [aŋen]

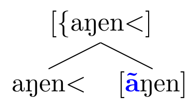
In (68), the root-final nasal /n/ of the L-PROJ precedes the word-initial /a/ of the R-PROJ's DUP-DOMAIN. This triggers nasalisation:

[13] This is where defining PROJECT recursively complicates the analysis; the ability to trigger rules after each step of the derivation is more cleanly modelled in an iterative scheme.

(69)

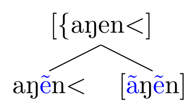


(70)



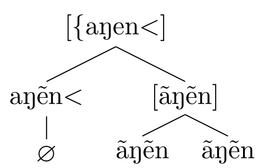
In (69), we see the inter-branch environment; (70) shows the outcome of nasalisation between the two branches. Here our focus is on the boldface **ã**; no other nasalisation has been shown. The full effects of the rule are shown below,

(71)



The change of $e \rightarrow \tilde{e}$ occurs as expected. After applying nasalization, projection proceeds, yielding the following:

(72)



⇒ ãŋẽnãŋẽn

Nothing is projected from the left hand branch as nothing falls to the right of the < marker.

6 Challenges

In this section we present two kinds of data that present challenges for the projection model of reduplication we have developed.

6.1 Thao triplication

Blust (2001) discusses reduplication and apparent triplication in Thao, "a moribund Austronesian language still spoken by about 15 elderly persons in the region of Sun-Moon Lake, central Taiwan". As one might expect, the data is unclear, and it is difficult to determine which patterns are synchronically productive. Here we restrict ourselves to discussion of one class of forms that would appear to be problematic for our model, and then present a solution based on Blust's analysis. The form *turu*, glossed as 'three', is used when counting serially or referring to non-human entities. There also exists a form *tatatturu* which is glossed as 'three at a time (of people)'. Similar pairs include *rima* 'five' / *rararima* 'five at a time (of people)' and *pitu* 'seven' / *papapitu* 'seven at a time (of people)'.

Our model can generate such forms, as developed thus far. In order to create two copies of the initial C with the fixed segment *a* followed by the full root, the insertion of two junctures is required in a reduplication domain built around the initial C.

The juncture insertion rules for forms like *papapitu* and *rararima* are as follows:

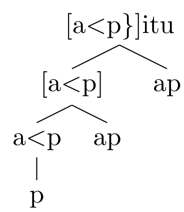
1. Insert [before the initial C and] after it;
2. Prepend a < at the beginning of the DUP-DOMAIN
3. Prepend the fixed segment *a* before <
4. Insert } before]

Given the root *pitu*, the application of the above rules yields [a<p}]itu,

(73) pitu ⇒ [p]itu ⇒ [<p]itu ⇒ [a<p]itu ⇒ [a<p}]itu

This results in the projection shown in (74):

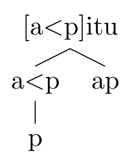
(74)



⇒ *papapitu*

On the other hand, forms like *taturu, rarima, papitu* can be generated as follows:

(75)



⇒ *papitu*

Blust suggests that the apparently triplicated forms do not represent the output of a single morphological reduplication, but rather serial reduplication. One process creates a numeral referring to humans, and another provides what Blust calls the 'distributive' reading. Informally, *papapitu* is not triplication of *pitu*, but rather is derived from *papitu.*

Interestingly, our scheme differentiates these two forms by whether or not the fourth juncture insertion rule applies. In the case of *papitu*, insertion rules (1–3) apply; for *papapitu*, the additional brace insertion—rule (4)—yields a nested reduplication domain and hence the triplicated form.

This analysis provides an interesting answer to questions such as which morphological process applies to the bare root, the one that converts numerals to refer to humans, or the distributive? Additionally, our proposal reduces this process down to simple agglutination—a single process of juncture insertion and affixation followed by projection and linearization can account for both *papitu* and *papapitu*.

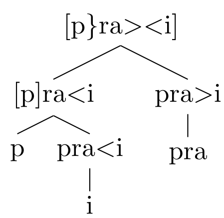
We have provided a sketch of a possible analysis of the relationship between such forms but relegate the bulk of work to future research projects. The point here is that superficially problematic data for our model may be amenable to simple alternative analyses.

6.2 Sanskrit sT clusters

There are a class of forms, traditionally considered to represent a real reduplication pattern, that occur in the history of Sanskrit that our scheme cannot handle. These are forms that contain a root that begins with *s* followed by a voiceless stop and contain a fixed segment *i* or *a*, depending on the morphological category. For example, the root *stamb^h-* reduplicates in the perfect as *tastamb^ha*. In order to maintain a simple exposition, we abstract away from complicating factors of segmental phonology such as palatalization discussed by Halle (q.v. for more Sanksrit forms) and restrict discussion to hypothetical roots that contain inital consonant clusters. A hypothetical root with a cluster of *s* followed by a stop, like *spa-* can reduplicate as *pi-spa* with the second member of the initial cluster repeated. This is in contrast to a root like *pra* or *sna* that would reduplicate as *pi-pra* or *si-sna*, with the initial consonant repeated.

Adopting the fixed segmentism developed for languages such as Yoruba, we can assume that the input for *pipra* is [p}ra><i]:

(76)



⇒ *pipra*

The problem with a form like *pispa* is that the string before the fixed segment *i* has two gaps *vis-à-vis* the underlying root, but the only way to generate gaps in the left projection is via the angle bracket relevant for the left projection, <. With only one such bracket we can only get one gap. So, we understand why these forms are problematic, in fact our model predicts that they will be more complex than forms like *pipra*.

No model of reduplication that merely makes reference to sequences of segments can handle these cases since it is necessary to distinguish among different root-initial clusters according to their sonority profile. We leave the details of such cases in our framework for future research, in the expectation that something like Halle's (2008) treatment can be adapted to our purposes.

7 Conclusions

We have shown that it is possible to account for the attested data using a simple iterative procedure, PROJECT, to derive a variety of reduplicated surface forms from simple inputs—and that this is possible without positing abstract 'base' and 'reduplicant' categories. Instead, we have proposed that the scheme outlined in Halle (2008) is more appealing from a theoretical standpoint for reasons of simplicity and elegance. Simple morphological junctures can drive seemingly diverse surface forms. We have rendered this explicit by proposing a simple algorithm suitable for serial models of phonology, and the validity of the algorithm is confirmed by our implementation.

As noted in (§5), the claim of McCarthy and Prince that cases of over- and underapplication *favour* a parallelist approach such as BASE-REDUPLICANT CORRESPONDENCE THEORY does not hold. Building on both Raimy (2000) and Halle (2008) we have proposed elegant and workable solutions to the 'problems' within a serial paradigm. Crucially, we have argued that Malay and Akan *do not* present cases of over- and under-application; rather, they can be analysed through *ordered rule application*.

Two immediate challenges arise to our proposals. Smolensky (p.c. in comments at *GLOW 2009*) suggests that if one has all the other kinds of Correspondence Constraints used in Optimality Theory, then it makes sense to handle reduplication with the same kind of constraints. The Base-Reduplicant constraints of OT reduplication models are so similar in format to Output-Output constraints and Input-Output constraints that it makes sense, if one works in OT, to assume that the specific constraint kinds are built from the same primitives as much as possible. If one does work in an OT framework, then Smolensky's point is valid; if one does *not*, however, then there is no obvious way to compare isolated components of one framework with those of another.

On the other hand, there is a positive challenge in Smolensky's suggestion which we *are* pursuing. Note that the square brackets we have been using actually serve two purposes. On the one hand, they delimit the boundaries of the string subject to projection; on the other hand, the brackets themselves trigger the double projection. These two functions can be disentangled, and the brackets can perhaps be employed in the derivation of other morphological processes. To give a simple indication of how such work would proceed, consider the possibility of using angle brackets without multiple projection to encode *subtraction* morphology. Given a bracketed domain, not subject to multiple projection, angle brackets could be used to ensure that certain segments do *not* surface. This work is further developed in our ongoing research.

References

Blust, R. 2001. Thao triplication. *Oceanic Linguistics* 40:324–335.

Broselow, E., and J. McCarthy. 1983. A theory of internal reduplication. *The Linguistic Review* 3:25–88.

Frampton, J. 2004. Distributed reduplication. *Ms., Northeastern University* URL http://www.math.neu.edu/ling/dr/DR_March1_04.pdf.

Hale, M., and C. Reiss. 2008. *The phonological enterprise*. Oxford University Press, USA.

Halle, Morris. 2008. Reduplication. In *Foundational issues in linguistic theory*, ed. Robert Freidin, Carlos Otero, and Maria L. Zubizaretta, 325–257. Cambridge, MA: MIT Press.

Idsardi, W., and E. Raimy. 2008. Reduplicative economy. In *Rules, constraints, and phonological phenomena*, ed. Bert Vaux and Andrew Nevins, 149–185. Oxford University Press, USA.

Inkelas, Sharon, and Cheryl Zoll. 2005. *Reduplication: doubling in morphology*. Cambridge; New York: Cambridge University Press.

Kiparsky, P. 2007. Reduplication in Stratal OT. *Ms.* URL http://www.stanford.edu/~kiparsky/Papers/reduplication.pdf.

McCarthy, J. J. 2000. Harmonic serialism and parallelism. *ROA-357, Rutgers Optimality Archive* URL http://roa.rutgers.edu/view.php3?roa=357.

McCarthy, J. J., and A. Prince. 1995. Faithfulness and reduplicative identity. *ROA-60, Rutgers Optimality Archive* URL http://roa.rutgers.edu/view.php3?roa=60.

Raimy, E. 2000. *The phonology and morphology of reduplication*. Walter de Gruyter.

Raimy, E., and W. Idsardi. forthcoming. *Precedence based phonology*.

Reiss, C. 2003. Quantification in structural descriptions: Attested and unattested patterns. *The linguistic review* 20:305–338.

Watters, D. E. 2002. *A grammar of kham*. Cambridge University Press.

A Perl Implementation

At present, there are two working implementations of Project: one written in Haskell, the other Perl. Both models seek to render the underlying primitives of our scheme fully explicit while providing a test-bed for investigating the typological questions raised above.

We encourage the reader to experiment with our Perl implemenation,

http://linguistics.concordia.ca/marc/projection/

to get a better feel for the nature of structured projection. Technical comments, criticisms and suggestions are welcome.