Norvin Richards Phonology Generals Paper

An Optimality-Theoretic Look at Ponapean Reduplication

Ponapean is an Austronesian language spoken on the island of Ponape. It has a remarkably complicated system of aspectual reduplication, with the reduplicant taking on a variety of forms depending on the phonological shape of the root. This system has been analyzed in a number of works, including Levin (1985) and McCarthy and Prince (1986). In what follows I will offer an analysis of Ponapean reduplication in the framework of Optimality Theory. I will try to show both that Optimality Theory is a very useful tool for dealing with the variety of reduplicants found in Ponapean, and that in order to deal with the Ponapean facts we will be forced to admit the existence of intermediate levels of representation, a move which optimality theoreticians have thus far traditionally shunned.

1. The data

The patterns of reduplication reported by Rehg and Sohl (1981) for Ponapean verbs are as follows[1]:

[1] I will be using the standard Ponapean orthography, described in Rehg and Sohl (1981), for most of the Ponapean examples in this paper. I will provide phonetic transcriptions of this orthography where necessary, but it is probably worth noting at the outset that the letter *i* can stand either for a vowel or for a glide, and that long vowels are represented by a vowel followed by *h* (e.g., *ah*=/a:/).

Pattern I: CVC, CVG-->CVC-, CVG-
lal 'to make a sound'-->lallal
dou 'to climb' -->doudou
Pattern II: CV-->C∇-
pa 'to weave' -->pahpa /pāpa/
du 'to dive' -->duhdu /tūtu/
Pattern III: VC-->VC∇C
el 'to rub or massage' -->elehl /elēl/
uk 'fast' -->ukuhk /ukūk/
Pattern IV: ∇C-->Vy-
ahn 'to be accustomed to' -->aiahn /ayān/
ohn 'hung over' -->oiohn /oyōn/
Pattern V: GV(C)-->Ge-
was 'obnoxious' -->wewas
iang 'to accompany' -->ieiang /yeyaŋ/
Pattern VI: heavy syllable base-->CV-
duhp 'to dive' -->duduhp /tutūp/
laud 'big, old' -->lalaud
pou 'cold' -->popou
mand 'tame' -->mamand
Pattern VII: polysyllables beginning with a vowel-->VCi-
alu 'to walk' -->alialu /aliyalu/
arekarek 'gritty' -->ariarekarek /ariyarekarek/
Pattern VIII: Polysyllables beginning CiyV- or CuwV--->CĪ-/CŪ-
liahn 'outgoing' -->lihliahn /lĪliyān/
luwak 'jealous' -->luhluwak /lūluwak/
Pattern IX: polysyllables beginning with a heavy syllable.
IX.a; ...whose second syllable is also heavy-->CV-
luhmwuhmw 'to be sick' -->luluhmwuhmw /lulūmwūmw/
wahntuhke 'to calculate' -->wawahntuhke /wawānchūke/
IX.b; ...whose second syllable is light-->C∇-
duhpek 'starved' -->duhduhpek /tūtūpek/
mehlel 'true' -->mehmehlel /mēmēlel/
Pattern X: NN..., NC...-->NNi-, NCi-
mmed 'full' -->mmimmed /mimmet/
mpek 'to look for lice' -->mpimpek /m̥pimpek/
Pattern XI: other polysyllables: CVC-
rere 'to skin or peel' -->rerrere
deied 'to eat breakfast' -->deideied

In looking at these patterns, it looks as though the default reduplicant is a CVC prefix; this is particularly conspicuous in

Pattern XI, where we see that polysyllables not fitting the descriptions for any of the other patterns reduplicate with a prefix of this shape.

Still, there are clearly a number of other factors in play in determining the shape of the reduplicant. Let us see if we can isolate these, starting with the monosyllables.

2. Monosyllables

Pattern I: CVC, CVG-->CVC-, CVG-

lal	'to make a sound'	-->lallal	
dou	'to climb'	-->doudou	

Pattern II: CV-->C∇-

pa	'to weave'	-->pahpa	/pāpa/
du	'to dive'	-->duhdu	/tūtu/

Pattern III: VC-->VC∇C

el	'to rub or massage'	-->elehl	/elēl/
uk	'fast'	-->ukuhk	/ukūk/

Pattern IV: ∇C-->Vy-

ahn	'to be accustomed to'	-->aiahn	/ayān/
ohn	'hung over'	-->oiohn	/oyōn/

Pattern V: GV(C)-->Ge-

was	'obnoxious'	-->wewas	
iang	'to accompany'	-->ieiang	/yeyaŋ/

Pattern VI: heavy syllable base-->CV-

duhp	'to dive'	-->duduhp	/tutūp/
laud	'big, old'	-->lalaud	
pou	'cold'	-->popou	
mand	'tame'	-->mamand	

2.1 Preliminaries

2.1.1 Pattern V

To begin with, we can eliminate Pattern V from consideration. Rehg and Sohl (1981) report that this pattern is disappearing from the language as the verbs that might be expected to undergo it move over to other patterns.

2.1.2 Final vowel deletion

The discerning reader may have noticed that no pattern is listed for roots of the form C∇. This appears not to be an accident. Rehg and Sohl (1979) list only seven verbs of this shape[2]:

1. kih- 'to give'
   koh- 'to come or go'
   loh 'to have something stuck in one's throat'
   luh 'to be incomplete, as a result of some part being removed'
   mah 'ripe'
   meh 'to bleat'
   mwoh 'to moo, of a cow'
   doh 'to be distant'

Of these, the first two are bound morphemes, obligatorily occurring with various kinds of directional suffixes, and ***meh*** 'bleat' and ***mwoh*** 'moo' are arguably onomatopoeic. There are thus only four clear cases of verb stems of the shape C∇.

Furthermore, there are fairly good reasons for believing that words ending in short vowels are generally derived from underlying forms ending in long vowels in Ponapean. Rehg and Sohl (1981, 88-89) note the following forms:

2. kihl 'skin' kiliniki 'having skin'
   ngihl 'voice' ngileniki 'having a voice'
   dihp 'sin' dipaniki 'having sin'
   nta 'blood' ntahniki 'having blood' (/n̥chāniki/)

When a consonant-initial suffix is attached to a consonant-final noun in Ponapean, a vowel generally appears between the two consonants. Interestingly, the quality of this vowel appears not to be predictable from the phonological shape of the root; an ***i*** is inserted for ***kihl*** 'skin', an ***e*** for ***ngihl*** 'voice', and an ***a*** for ***dihp*** 'sin'. These vowels will therefore apparently have to be taken to be part

[2] By contrast, there are more than thirty nouns of this shape.

of the underlying representation of the root; these words will be represented as something like ***kili***, ***ngile***, and ***dipa*** respectively, and final vowel loss will eliminate the last vowel when the word stands in isolation[3]. Interestingly enough, vowel-final nouns typically lengthen the final vowel when suffixes are added to them. By parity of reasoning with the apparently consonant-final cases, then, we apparently must assume that these words actually end in long vowels underlyingly, and that the long vowel is shortened when the word stands in isolation. This assumption will be useful in dealing with Pattern II; it looks as though we can safely assume that the vowels in these forms are actually underlyingly long.

2.1.3 Final Consonant Extrametricality

One of our first difficulties in dealing with the monosyllabic reduplication patterns will be distinguishing between patterns I and VI. Pattern VI appears to contain heavy monosyllables, but Pattern I contains CVC monosyllables, which are metrically heavy in many languages. Furthermore, the unmarked reduplicant looks like a bimoraic syllable; that is, like a minimal heavy syllable in many languages. Thus, it looks as though CVC is a heavy syllable initially (as a reduplicant, for example) but not finally (since the roots in Pattern I are not in Pattern VI). We might assume a three-way weight distinction between light, heavy, and extra-heavy syllables, as has been typically assumed for Egyptian Arabic, for example.

---

[3] There are a number of technical ways of achieving this, of course. One would be to assume that the final vowels in these forms are somehow underspecified. If specification of underspecified vowels carries some "cost" (an intuition which is trivial to spell out in optimality-theoretic terms), then we should not be surprised to see these vowels surfacing only when other constraints require them to (in this case, possibly phonotactic constraints, or a ban on unparsed material intervening between parsed segments).

Here I will follow McCarthy and Prince (1986) in operating under a different assumption; that final consonants are extrametrical in Ponapean. McCarthy and Prince (1986) cite the data in 2 as evidence for this claim. The vowel lengthening in monosyllabic nouns that surface as consonant-final is taken to be a response to a minimal word requirement on nouns requiring them to be minimally bimoraic. Thus, lengthening is triggered by the extrametricality of the final consonant. As expected, vowels do not lengthen in monosyllabic nouns ending in consonant clusters (e.g., ***emp*** 'coconut crab', ***mall*** 'grassy area', ***kengk*** 'coconut, containing no nut'), since such nouns are still bimoraic even with final consonant extrametricality. Similarly, there are many polysyllabic nouns with no long vowels (***kasap*** 'frigate bird', ***lapwed*** 'salt water eel'). Rehg and Sohl (1979) list more than thirty nouns of the shape C∇, but only one CV noun root (***se*** 'sandy place inside a barrier reef'), which is what we expect, again, if the minimal word requirement on nouns outweighs the tendency toward final vowel deletion discussed in the previous section; almost all roots of the shape CV have their vowels lengthened to conform to the requirement that nouns be at least bimoraic syllables[4]. Finally, there are many nouns with long vowels in both affixed and unaffixed forms, which suggests that the data in 2 are indeed cases of vowel lengthening in the unaffixed form rather than of vowel shortening in the affixed form:

---

[4] This predicts that some of these C∇ nouns should have their vowels shorten in their possessed forms; I have been unable to find any data bearing on this prediction.

3. rahn 'day' rahnin 'day of'

moahd 'echo' moahdin 'echo of'

kehp 'yam' kehpin 'yam of'

This leads us to the conclusion that CVC syllables should be metrically light word-finally, and metrically heavy in other positions in the word. Thus, the unmarked CVC reduplicant is a minimal metrically heavy syllable, while Pattern I verb stems like ***lal*** 'to make a noise' are metrically light.

2.2 Patterns I-IV and VI

Now we are in a position to account for the reduplication facts having to do with reduplication of monosyllables.

Pattern I: CVC, CVG-->CVC-, CVG-
- lal 'to make a sound'-->lallal
- dou 'to climb' -->doudou

Pattern II: CV-->C∇-
- pa 'to weave' -->pahpa /pāpa/
- du 'to dive' -->duhdu /tūtu/

Pattern III: VC-->VC∇C
- el 'to rub or massage' -->elehl /elēl/
- uk 'fast' -->ukuhk /ukūk/

Pattern IV: ∇C-->Vy-
- ahn 'to be accustomed to' -->aiahn /ayān/
- ohn 'hung over' -->oiohn /oyōn/

Pattern VI: heavy syllable base-->CV-
- duhp 'to dive' -->duduhp /tutūp/
- laud 'big, old' -->lalaud
- pou 'cold' -->popou
- mand 'tame' -->mamand

We have already seen that the default reduplicant in Ponapean is CVC-; that is, somewhere in the constraint ranking there must be a constraint RED=$\sigma_{\mu\mu}$[5]. This reduplicant appears in Patterns I and II. In Patterns III, IV, and VI, on the other hand, the constraint is

---

[5] For definitions of this and other constraints used here, see Appendix A.

violated; we get a light CV syllable. The result, in each case, is an iambic foot (that is, a foot consisting of a light syllable followed by a heavy syllable, where light syllables are monomoraic and heavy syllables contain more than one mora). One way of dealing with this pattern, then, would be to assume that there is a constraint preferring iambic feet (call it Iamb) which outranks RED=$\sigma_{\mu\mu}$, forcing the use of a non-canonical reduplicant when this will create an iamb. In patterns I and II, where the base is light (recall that we are assuming final CVC syllables to be metrically light, following McCarthy and Prince 1986) and no form of a prefixal reduplicant could possibly create an iamb, the default heavy reduplicant is used. Let us now look at each pattern in more detail.

2.2.1 Pattern I: lal-->lallal

Pattern I involves a consonant-initial monosyllabic stem which, because of final consonant extrasyllabicity, is metrically light; the reduplicant is a heavy CVC prefix. We have already hypothesized that Iamb>>RED=$\sigma_{\mu\mu}$, so if any means for creating an iamb exist in this case, we will need to rule them out with constraints that outrank Iamb. In the following tableau the reduplicant is underlined, and the winning candidate is marked with a dollar sign:

RED+lal-->lallal

	$\underline{\text{lal}}$lal	l$\underline{\text{al}}$al	l$\underline{\text{al}}$aAl	lal$\underline{\text{aal}}$
Fill	✓	✓	*!	✓
RED=Prefix	✓	✓	✓	*!
Iamb	*	*	✓	✓
RED=$\sigma_{\mu\mu}$	✓	*!	*	✓

The first candidate is the one which we want to win; it violates Iamb and obeys RED=$\sigma_{\mu\mu}$. The last candidate successfully creates an iamb by suffixing the reduplicant; there are a number of ways of ruling this particular candidate out, but I will use a constraint requiring the reduplicant to be a prefix[6], which will be useful later, as we will see. The third candidate creates an iamb by adding an extra vowel to the base; apparently this is ruled out in Ponapean as well, so Fill will need to be highly ranked. Finally, the second candidate ties with the first on the first three constraints, but fails to create a heavy reduplicant and thus loses in the end.

2.2.2 Pattern II: **pa-->pahpa**

Pattern II involves a stem which, by hypothesis, underlyingly consists of a consonant followed by a long vowel. Let us follow the line suggested in footnote 3 and assume that ***pa*** ends in an underspecified vowel; that is, ***pa*** is really /paV/. I will not investigate the nature of underspecified vowels in this paper in any detail; all that will be necessary for our purposes will be that Fill (or whatever version of Fill applies to the addition of vocalic features to an underspecified vowel; perhaps there is a specific constraint Fill (V-feature)) outrank Parse and Iamb, while Base-Dependence for vocalic features is outranked by RED=$\sigma_{\mu\mu}$. This has the result that adding features to an underspecified vowel in the base is more costly than adding features to a copy of an underspecified vowel in a reduplicant, since Fill outranks Base-Dependence; that is, it is easier to realize /paV/ as [pā] as a

---

[6] This could easily be expressed as an Align constraint; I will refer to it as Prefix simply for clarity's sake, since we will be using another Align constraint shortly.

reduplicant than as a base. Thus, no iamb can possibly be created, either by suffixation of the reduplicant (the fourth candidate below) or by filling in the underspecified vowel in the base (by hypothesis, Fill outranks Iamb), and the form with a bimoraic reduplicant wins out.

RED+pa-->pahpa

	$paapa<V>	papa<V>	papaa	papaa
Fill	✓	✓	*!	✓
RED=Prefix	✓	✓	✓	*!
Iamb	*	*	✓	✓
RED=$\sigma_{\mu\mu}$	✓	*!	*	✓
B-D(V-f.)	*	✓	✓	*

2.2.3 Pattern III: el-->elehl

Pattern III involves a short-vowel-initial monosyllabic stem. We might take this form as involving suffixation, but this would get us in trouble later (in particular, with patterns involving polysyllables, which clearly never involve suffixation, even to create iambs). We will apparently be better off assuming that in a form like ***eleel*** the base is the final ***-el*** and the reduplicant is a prefix ***ele-***, as in 4:

4.
```
     V     C          V     C
     g     g     eh         g
     e     l     e    e     l
      z--------m z----m
           1          1
     Reduplicant      Base
```

Thus, both RED=$\sigma_{\mu\mu}$ and Base-Dependence for Vowels are violated in this form in order to create an iamb. We have no means of ranking B-D (V) and RED=$\sigma_{\mu\mu}$ here; we will see later that the ranking is as shown.

RED+el-->elehl

	$eleel	elel	eelel
Fill	✓	✓	✓
RED=Prefix	✓	✓	✓
Iamb	✓	*!	*!
B-D(V)	*	✓	*
RED=$\sigma_{\mu\mu}$	*	*	✓

2.2.4 Pattern VI: duhp-->duduhp

Pattern VI is the case of heavy monosyllables; here we see, as mentioned before, that Iamb outranks RED=$\sigma_{\mu\mu}$.

RED+duhp-->duduhp

	$duduup	duuduup	duupduup
Fill	✓	✓	✓
RED=Prefix	✓	✓	✓
Iamb	✓	*!	*!
B-D(V)	✓	✓	✓
RED=$\sigma_{\mu\mu}$	*	✓	✓

2.2.5 Pattern IV: ahn-->aiahn

Pattern IV is the most interesting of the monosyllabic patterns, showing as it does the lengths to which Ponapean is willing to go to create iambs. Given a stem like ***ahn*** there are in principle at least three things we could do to create an iamb; we could copy the first vowel and the consonant, skipping the second vowel-slot (giving us a reduplicant ***a<a>n-***), or we could skip the first vowel slot and copy the second vowel and the consonant (***<a>an-***), or we could copy only one of the vowels and insert an epenthetic consonant (***a<an>C-***). The first of these violates McCarthy and Prince's (1994) Contiguity constraint on reduplication, which

requires us not to skip segments. The second violates the Anchoring constraint, again from McCarthy and Prince (1994). The third violates Base-Dependence for Consonants, as well as involving a greater violation of McCarthy and Prince's (1994) Max constraint than the other two options do. Given that the third option is preferred, we can see here that Contiguity and Anchoring outrank both B-D(C) and Max. The fact that we prefer to violate B-D(C) and Max in order to create an iamb (rather than creating a form ***ahnahn***, for example) shows that Iamb outranks these constraints as well.

RED+ahn-->aiahn

	$\underline{ai}$aan	__$\underline{an}$aan	a__$\underline{n}$aan	$\underline{ahn}$ahn
Cont	✓	✓	*!	✓
Anch	✓	*!	✓	✓
Fill	✓	✓	✓	✓
RED=Prefix	✓	✓	✓	✓
Iamb	✓	✓	✓	*!
B-D(V)	✓	✓	✓	✓
RED=$\sigma_{\mu\mu}$	*	*	*	✓
Max	**	*	*	✓
B-D(C)	*	✓	✓	✓

3. Polysyllables

When we turn our attention to the polysyllables, we can see that some other factors must be at work. Vowel-initial polysyllables, for example, do not typically behave like their monosyllabic counterparts[7]:

[7] I use % here to indicate the acceptability of a form in certain dialects.

5.	a.	el 'massage'	-->elehl
	b.	alu 'walk'	-->alialu
		arekarek 'gritty'	-->ariarekarek
	c.	amas 'raw, sober'	-->%amiamas
			-->%amahmas
		ewetik 'abstemious'	-->%ewiewetik
			-->%ewehwetik

In 5.a we see what we have seen before; a vowel-initial monosyllable reduplicates in such a way as to create an iamb, thereby violating RED=$\sigma_{\mu\mu}$ and B-D(V). In 5.b, however, no iambs are created. Interestingly, in 5.c we can see a dialect split reported by Rehg and Sohl (1981); apparently some speakers do create iambs at the beginnings of vowel-initial polysyllables just in case no violations of B-D(V) result.

Similarly, polysyllables beginning with heavy syllables do not always behave like monosyllabic heavy syllables:

6.	a.	duhp 'dive'	-->duduhp
	b.	luhmwuhmw 'be sick'	-->luluhmwuhmw
		wahntuhke 'calculate'	-->wawahntuhke
	c.	duhpek 'starved'	-->duhduhpek

Polysyllables beginning with heavy syllables create iambs at the beginning apparently just in case the rest of the word can stand on its own, metrically speaking. If what would be left over if the first two syllables were parsed as an iambic foot would be a single light syllable, then an iamb is not created. This is shown in 6.c, where creation of an initial iamb in the reduplicated form (**(duduh)pek*) would leave a single light syllable at the end of the word. If the remnant of the word would be a single heavy syllable (as in the first example in 6.b) or a pair of syllables (as in the second example in 6.b), on the other hand, then an iamb can be created.

These examples suggest that the metrical status of the end of the word is especially important. In 5 we saw that iambs are created at the ends of words in violation of B-D(V) and RED=$\sigma_{\mu\mu}$, while iambs at the beginnings of words are not sufficiently valuable for these constraints to be violated. In 6, we see that the metrical status of the end of the word overrides the mechanisms that conspire to create iambs in similar environments in monosyllables.

Thus far we have been assuming that the word is completely parsed into feet, a fairly reasonable assumption when we were looking exclusively at monosyllables. As we begin to look at polysyllables, however, this assumption is no longer so innocent. Standard Optimality Theory gives us two mechanisms for forcing metrical parsing of syllables. One is the constraint Parse (σ, Ft), and the other involves Align constraints requiring alignment of edges of various categories with feet. Apparently what we need here is a comparatively high-ranking constraint Align (Pwd, R, Ft, R)[8] along with a lower-ranking constraint Parse (σ, Ft). This will give us the desired results; proper metrification of the end of the word (or the whole word, in the case of monosyllable reduplication) will be more important than metrification of the rest of the word. This will be the basic intuition we will pursue in what follows.

3.1 Patterns VIII and XI: liahn-->lihliahn and rere-->rerrere

Let us begin with two of the less interesting patterns. These can both be analyzed as CVC reduplication. No initial iambs can be

---

[8] Given that verbal suffixation does not affect reduplication, this is not quite right; we will need to invoke alignment of some category containing the verb stem but not the suffixes with a foot (perhaps an Align Stem constraint).

created in these cases, since the base words do not begin with heavy syllables, and the high-ranking Contiguity, Anchoring, and RED=Prefix constraints forbid us to create final iambs by suffixation. We therefore give up on creating iambs and simply prefix a bimoraic syllable. In the case of Pattern VIII (***liahn*** /liyān/-->***lihliahn*** /lĪliyān/), we can either claim that phonetic effects cause a tautosyllabic high vowel+glide sequence to be realized as a long vowel (that is, we could say that ***liahn*** reduplicates as /liyliyān/, and that phonetic effects realize the first ***iy*** sequence as ***Ī***or that some high-ranking constraint rules out such sequences, forcing reduplication to create a long vowel in order to satisfy RED=$\sigma_{\mu\mu}$. I will arbitrarily assume the first of these here, although nothing depends on this.

RED+rere-->rerrere

	$rer(rere)	ree(rere)	re(rere)	(rere-er)
Cont	✓	✓	✓	*!
Anch	✓	✓	✓	✓
Fill	✓	✓	✓	✓
RED=Prefix	✓	✓	✓	*
Iamb	*	*	*	✓
B-D(V)	✓	*!	✓	✓
RED=$\sigma_{\mu\mu}$	✓	✓	*!	*
Max	*	**	**	**
B-D(C)	✓	✓	✓	✓

RED+liahn-->lihliahn

	liy(liyaan)	li(liyaan)
Cont	✓	✓
Anch	✓	✓
Fill	✓	✓
RED=Prefix	✓	✓
Iamb	✓	✓
B-D(V)	✓	✓
RED=$\sigma_{\mu\mu}$	✓	*!
Max	***	****
B-D(C)	✓	✓

3.2 Pattern IX: luhmwuhmw-->luluhmwuhmw
duhpek-->duhduhpek

Pattern IX is the case of polysyllables beginning with heavy syllables. In this case, an iamb is created just in case the remaining syllables can create a well-formed foot aligned with the right edge of the word.

In the case where an initial iamb is created, we can see that the constraint forcing creation of nonfinal feet (namely, Parse (σ,Ft)) must be higher-ranked than both Max and RED=$\sigma_{\mu\mu}$, so that the reduplicant will be smaller than a bimoraic syllable, thus creating an initial iamb[9]:

[9] I am assuming here that feet consisting of multiple heavy syllables are ruled out, either by a high-ranking constraint or by properties of Gen.

RED+luhmwuhmw-->luluhmwuhmw

	(luluh) (mwuhmw)	luhluh (mwuhmw)	(luh)(luh) (mwuhmw)
Cont	✓	✓	✓
Anch	✓	✓	✓
Fill	✓	✓	✓
RED=Prefix	✓	✓	✓
Iamb	✓*	*	***!
B-D(V)	✓	✓	✓
Parse(σ,Ft)	✓	**!	✓
RED=$\sigma_{\mu\mu}$	*	✓	✓
Max	*****	****	****
B-D(C)	✓	✓	✓

Now we can turn to the cases where an initial iamb is not created, apparently in order to avoid stranding a single light syllable at the end of the word. In order to get this effect we will need, first of all, to have Align (Pwd, R, Ft, R) outrank Iamb, since the winning form sacrifices initial iamb creation in order to contain a final foot. We will also have to say that a constraint marking feet consisting of a single light syllable as ill-formed (FtHd=H) outranks Parse (σ,Ft); this will have the effect of ruling out the second candidate in the table below, which is superior to the winning candidate in terms of foot parsing but contains a foot consisting of a single light syllable[10].

[10] Note that I am crucially assuming here that Iamb counts non-iambic feet and marks them as violations, ignoring iambic feet; otherwise the second candidate, which does in fact contain an iamb, would beat the first candidate, which does not. By hypothesis, what is important for Iamb is that the first and the second candidate each contains one non-iambic foot, and they therefore tie on this constraint.

RED+duhpek-->duhduhpek

	$duh (duhpek)	(duduh)(pek)	(duduh)pek	du(duhpek)
Cont	✓	✓	✓	✓
Anch	✓	✓	✓	✓
Fill	✓	✓	✓	✓
RED=Prefix	✓	✓	✓	✓
Align (Pwd, R, Ft, R)	✓	✓	*!	✓
lamb	*	✓*	✓	*
B-D(V)	✓	✓	✓	✓
FtHd=H	✓	*!	✓	✓
Parse(σ,Ft)	*	✓	*	*
RED=$\sigma_{\mu\mu}$	✓	*	*	*!
Max	***	****	****	****
B-D(C)	✓	✓	✓	✓

3.3 Pattern X: mpek-->mpimpek

mmed-->mmimmed

This is one of the two patterns for which an intermediate level of representation appears to be necessary. Rehg and Sohl (1981, 52-53) state that forms like *mpek* are bisyllabic, consisting of a syllabic nasal followed by a CVC syllable. They also note that this is the only position in which syllabic nasals are possible; apparently syllabic nasals cannot have onsets or codas.

Running a form like this through our constraints as they now stand, we expect to get ***m̩pm̩pek*** (this table crucially assumes that a syllable containing a syllabic nasal is heavy):

RED+mpek-->mpimpek

	m(pmpek)	m(pimpek)	m(pempek)	mpe(kmpek)
Cont	✓	✓	✓	✓
Anch	✓	✓	✓	✓
Fill	✓	✓	✓	✓
RED=Prefix	✓	✓	✓	✓
Align (Pwd, R, Ft, R)	✓	✓	✓	✓
Iamb	*	*	*	*
B-D(V)	✓	*!	✓	✓
FtHd=H	✓	✓	✓	✓
Parse(σ,Ft)	*	*	*	**!
RED=$\sigma_{\mu\mu}$	✓	*	*!	*
Max	**	**	*	✓
B-D(C)	✓	✓	✓	✓

In fact, *mpmpek* is not a possible word in Ponapean; as just noted, syllabic nasals can only appear in word-initial position. Creating a constraint to rule this form out should therefore be fairly straightforward. The problem is then how to get the second candidate in the tableau above to beat the third candidate. As the tableau shows, these forms are identical with respect to all the constraints we currently have but two; Base-Dependence (V) and Max, which mark the attested form as worse than its unattested competitor **mpempek*. It is hard to see what constraint we could possibly use to get the form which actually occurs. Note that there is no obvious phonological reason for *mpempek* to change to *mpimpek*; cf. such lexical items as *tempel* 'to rhythmically pound a kava stone', *lempe* 'unripe fallen fruit', and *kempenial* 'bush sp. (*Psychotria carolinensis*)'.

If we allow ourselves the use of multiple levels, of course, the problem becomes trivial. The output of the above tableau is in fact the correct one, and a later process of vowel epenthesis deals with the illicit syllabic nasal by inserting an *i*.

3.4 Pattern VII: alu-->alialu

This is the pattern involving polysyllabic vowel-initial roots. As we noted before, polysyllabic vowel-initial forms, unlike vowel-initial monosyllables, do not create initial iambs, and this was taken as evidence that two different constraints are involved in metrification[11].

RED+alu-->alialu

	al(yalu)	a(lalu)	a(laalu)	ali(alu)	alu(alu)
Cont	✓	✓	✓	✓	✓
Anch	✓	✓	✓	✓	✓
Fill	✓	✓	✓	✓	✓
RED=Prefix	✓	✓	✓	✓	✓
Align (Pwd, R, Ft, R)	✓	✓	✓	✓	✓
Iamb	*	*	*	*	*
B-D(V)	✓	✓	*!	*!	✓
FtHd=H	*	*	✓	*	*
Parse(σ,Ft)	*	*	*	**	**!
RED=$\sigma_{\mu\mu}$	✓	*!	*	*	*
Max	*	*	*	*	✓
B-D(C)	*	✓	✓	✓	✓

Again, our constraints as they stand predict a form which is tantalizingly close to the attested form but is still not correct; we are getting ***al.ya.lu***, as opposed to the attested, four-syllable form

[11] I am assuming here that the first candidate violates Base-Dependence for Consonants but not Fill, since the epenthetic consonant is part of the reduplicant.

a.li.ya.lu. Furthermore, it seems impossible to correct this prediction in the Optimality system; doing so would involve finding some way to prefer the fourth candidate to the fifth, despite the fact that the two candidates are identical in every way except that the fourth candidate involves less faithfulness to the base on the part of the reduplicant. In this case, in fact, we would have the additional problem of ruling out the second candidate, which is metrically preferable to the winning candidate in that it contains fewer unparsed syllables.

Interestingly enough, words like ***al.ya.lu*** are unattested in Ponapean. When clusters consisting of a consonant plus a following glide are created by affixation, some way of breaking them up is employed (Rehg and Sohl 1981, 236-7):

7. koh+iei -->kohiei [kōyey] 'to go out'
   lus+iei -->lusiei [lusiyey] 'to jump out'
   -->lusehi [lusēy]

The suffix ***-iei*** 'out' begins with a glide when it follows a vowel-final root such as ***koh*** 'go', but this glide is not allowed to immediately follow another consonant; either a vowel is inserted or the glide is deleted.

Here, again, the notion of using multiple levels of representation seems very attractive. We seem to want to say that the form generated by the above tableau is the right one, and that a later process of vowel insertion breaks up the illicit consonant cluster. Note that this is a classic case of the type that motivates derivational analyses; the glide is being inserted to close the reduplicant syllable, so that the constraint RED=$\sigma_{\mu\mu}$ can be obeyed, and a later rule inserts a vowel, destroying the bimoraicity of the

reduplicant which was the motivation for inserting the glide in the first place. This kind of blindness to the motivations of other rules is quite difficult, perhaps impossible, to model insightfully in a representational approach. I will discuss this problem again in section 4, arguing further for the approach proposed here.

3.5 Dialectical variation in Pattern VII

Rehg and Sohl (1981) note that in cases where Base-Dependence (V) need not be violated to create a heavy second syllable in the reduplicated form, such a syllable is created in some dialects, though not in all:

8. a. alu --> alialu
   arekarek --> ariarekarek
   b. amas --> %amiamas
   --> %amahmas
   ewetik --> %ewiewetik
   --> %ewehwetik

At this point, we would expect all dialects to choose to create heavy second syllables:

RED+amas-->amahmas

	a(mamas)	$a(maamas)	am(yamas)
Cont	✓	✓	✓
Anch	✓	✓	✓
Fill	✓	✓	✓
RED=Prefix	✓	✓	✓
Align (Pwd, R, Ft, R)	✓	✓	✓
lamb	*	*	*
B-D(V)	✓	✓	✓
FtHd=H	*!	✓	*!
Parse(σ,Ft)	*	*	*
RED=$\sigma_{\mu\mu}$	*	*	✓
Max	**	*	**
B-D(C)	✓	✓	*

RED+ewetik-->ewehwetik

	ewe(wetik)	$(ewee)(wetik)	ewye(wetik)
Cont	✓	✓	✓
Anch	✓	✓	✓
Fill	✓	✓	✓
RED=Prefix	✓	✓	✓
Align (Pwd, R, Ft, R)	✓	✓	✓
lamb	*	*✓	*
B-D(V)	✓	✓	✓
FtHd=H	*	✓*	*
Parse(σ,Ft)	**!	✓	**!
RED=$\sigma_{\mu\mu}$	*	*	✓
Max	****	***	****
B-D(C)	✓	✓	*

We apparently cannot deal with the dialectical variation here by permuting existing constraints. In order to prefer the third candidate over the second in the tableau for ***amas***, for example, we would have to reorder RED=$\sigma_{\mu\mu}$ and FtHd=H. This would have undesirable effects, however, on Pattern IX; in order to get the correct result for ***duhpek***, we need FtHd=H to outrank Parse(σ,Ft), and in order to get the correct result for ***luhmwuhmw***, we need Parse(σ,Ft) to outrank RED=$\sigma_{\mu\mu}$. So this kind of solution seems untenable; we will need a new constraint.

As was noted in the discussion of Pattern III (***el-->elehl***), forms such as ***amahmas*** are assumed here to have a structure like that in 9:

9.

	V	C		V	C	V	C
	g	g	eh		g	g	g
	a	m	a	a	m	a	s
	z		m	z			m
		1			1		
	Reduplicant			Base			

Such forms, in other words, involve a long vowel half of which is part of the reduplicant and half of which is part of the base. We might create a constraint ruling out such configurations:

1Ø. *V=RB

A vowel of the form

V

4

R B

is ill-formed.

There are various ways of making this look more elegant, of course (for example, we could generalize the constraint so that it bars vowels which cross morpheme boundaries), but this will do for our purposes.

For those dialects in which the behavior of words in pattern VII is completely consistent, then, we will want *V=RB to outrank FtHd=H, thus ruling out the second candidate in each of the following tableaux:

RED+amas-->amiamas

	a(mamas)	a(maamas)	$am(yamas)
Cont	✓	✓	✓
Anch	✓	✓	✓
Fill	✓	✓	✓
RED=Prefix	✓	✓	✓
Align (Pwd, R, Ft, R)	✓	✓	✓
lamb	*	*	*
B-D(V)	✓	✓	✓
*V=RB	✓	*!	✓
FtHd=H	*	✓	*
Parse(σ,Ft)	*	*	*
RED=$\sigma_{\mu\mu}$	*!	*	✓
Max	**	*	**
B-D(C)	✓	✓	*

RED+ewetik-->ewiewetik

	ewe(wetik)	(ewee) (wetik)	$ewye (wetik)
Cont	✓	✓	✓
Anch	✓	✓	✓
Fill	✓	✓	✓
RED=Prefix	✓	✓	✓
Align (Pwd, R, Ft, R)	✓	✓	✓
lamb	*	*✓	*
B-D(V)	✓	✓	✓
*V=RB	✓	*!	✓
FtHd=H	*	✓*	*
Parse(σ,Ft)	**	✓	**
RED=$\sigma_{\mu\mu}$	*!	*	✓
Max	****	***	****
B-D(C)	✓	✓	*

Thus, the dialect in which the reduplicated forms of *amas* and *ewetik* are *amiamas* and *ewiewetik* might be taken to have some

constraint like *V=RB outranking FtHd=H (and thus also Parse(σ,Ft)). Note that this will not affect our analysis of Pattern III (***el-->elehl***); in this form *V=RB is violated in order to better satisfy the higher-ranking constraints Align (Pwd, R, Ft, R) and lamb.

Having postulated the constraint *V=RB, we need to decide where to place it for the dialect in which it is violated in these forms (that is, in which the reduplicated forms of *amas* and *ewetik* are *amahmas* and *ewehwetik*). Interestingly enough, it must be outranked by Parse(σ,Ft); that is, it must move down two constraints, from above FtHd=H to below Parse(σ,Ft):

RED+amas-->amahmas

	a(mamas)	$a(maamas)	am(yamas)
Cont	✓	✓	✓
Anch	✓	✓	✓
Fill	✓	✓	✓
RED=Prefix	✓	✓	✓
Align (Pwd, R, Ft, R)	✓	✓	✓
lamb	*	*	*
B-D(V)	✓	✓	✓
FtHd=H	*!	✓	*!
Parse(σ,Ft)	*	*	*
*V=RB	✓	*	✓
RED=$\sigma_{\mu\mu}$	*	*	✓
Max	**	*	**
B-D(C)	✓	✓	*

RED+ewetik-->ewehwetik

	ewe(wetik)	$(ewee)(wetik)	ewye(wetik)
Cont	✓	✓	✓
Anch	✓	✓	✓
Fill	✓	✓	✓
RED=Prefix	✓	✓	✓
Align (Pwd, R, Ft, R)	✓	✓	✓
lamb	*	*✓	*
B-D(V)	✓	✓	✓
FtHd=H	*	✓*	*
Parse(σ,Ft)	**!	✓	**!
*V=RB	✓	*	✓
RED=$\sigma_{\mu\mu}$	*	*	✓
Max	****	***	****
B-D(C)	✓	✓	*

Since there are two different constraints involved in making the long vowels in these reduplicated forms desirable (namely, FtHd=H for *amas* and Parse(σ,Ft) for *ewetik*), *V=RB must be ranked above both of these constraints for a dialect in which long vowels are not created in either case, and below both for a dialect in which long vowels are created in both cases. If we think of dialect variation as involving strictly local permutation of constraints (an attractive, though certainly not a necessary way of thinking), we may find this mildly unsettling.

There are at least three possible solutions to this problem. One would involve showing that there are in fact three dialects: the two discussed above along with a third for which the reduplicated forms are *ewiewetik* and *amahmas*[12]. This third dialect would have

[12]Since FtHd=H outranks Parse(σ, Ft), the prediction would be that no dialect could exist for which the correct forms are *ewehwetik* and *amiamas*.

*V=RB ranked between FtHd=H and Parse(σ,Ft), and would allow us to describe the dialect variations in terms of strictly local permutation of constraints. Another possibility is that FtHd=H is crucially unlike lamb (cf. footnote 10) in that it counts obedient feet rather than violating feet. I have been assuming that FtHd=H is like lamb in this regard, and on this assumption FtHd=H fails to make any distinction among the candidates in the tableau for ***ewetik***, as they all violate the constraint once. If FtHd=H counts obedient feet, however, then it will choose the second candidate in the ***ewetik*** tableau, as this candidate, unlike the other two, obeys FtHd=H once. Then FtHd=H will be the relevant constraint marking long-vowel creation as desirable for both ***amas*** and ***ewetik***, and local permutation of FtHd=H and *V=RB will be sufficient to describe the dialects. Finally, we might describe the two dialects in terms of presence versus absence of the *V=RB constraint, rather than attempting to rerank it, as it does no work in any paradigms other than the one seen here. This runs counter to most current work in Optimality Theory, as far as I know, which assumes that the set of constraints is universal (modulo possible parametric variation of certain constraint types; cf. e.g. Everett (1994))

4. Patterns VII and X; the argument for multiple levels

In sections 3.3-3.4 we saw an argument for multiple levels of representation based on the forms in 11:

11. a. alu -->alialu /aliyalu/
arekarek -->ariarekarek /ariyarekarek/
b. mpek -->mpimpek /m̩pimpek/

Our constraints as they stand generate the following forms without the inserted /i/; that is, they give ***alyalu*** as the reduplicated form of ***alu*** and *mpmpek* as the reduplicated form of *mpek*. I wanted to claim that this was in fact the correct result; that these are the outputs of the optimality system, and that /i/ is later inserted in response to phonotactic constraints. The presence of the phonotactic constraints in question is not at issue, since in fact words like ***alyalu*** and *mpmpek* are not well-formed in Ponapean. The only question is whether to represent them as applying at a later level of representation, as I have claimed, or as applying in the same level of representation as the rest of the processes involved in reduplication, as a more orthodox optimality-theoretic stance would lead us to assume. I have argued that the former approach is the better one. Let us consider the tableau for ***alialu*** again:

RED+alu-->alialu

	al(yalu)	a(lalu)	a(laalu)	ali(alu)	alu(alu)
Cont	✓	✓	✓	✓	✓
Anch	✓	✓	✓	✓	✓
Fill	✓	✓	✓	✓	✓
RED=Prefix	✓	✓	✓	✓	✓
Align (Pwd, R, Ft, R)	✓	✓	✓	✓	✓
lamb	*	*	*	*	*
B-D(V)	✓	✓	*!	*!	✓
FtHd=H	*	*	✓	*	*
Parse(σ,Ft)	*	*	*	**	**!
RED=$\sigma_{\mu\mu}$	✓	*!	*	*	*
Max	*	*	*	*	✓
B-D(C)	*	✓	✓	✓	✓

There are at least two problems facing an account that tries to get the attested output ***alialu*** from a single level. Even if phonotactic constraints are added to the above tableau to knock out the first candidate (an uncontroversial move, as we have seen, since the first candidate is in fact phonotactically ill-formed in Ponapean), some way of getting ***alialu*** to beat ***alualu*** and ***alalu*** must still be found. ***alualu*** is superior to ***alialu*** by virtue of being more faithful to the base, and ***alalu*** is preferable because it has fewer unparsed syllables. In what follows I will try to show that any account we might construct to deal with the first of these difficulties will be awkward at best, and that the second is probably insuperable. First, I will demonstrate that the general behavior of reduplication in Ponapean leads us to expect that ***alualu*** should at least be an option, as "overcopying" of this type is attested in the language as a means of dealing with phonotactic constraints. I will then discuss the metrical constraints that lead us to prefer ***alalu*** to ***alialu***, and show that any alteration of the constraints that might solve this problem would have undesirable effects on other reduplication patterns.

4.1 Ruling out *alualu*

4.1.1 Phonotactics

Ponapean phonotactics are both fairly strict and fairly complicated, and a detailed analysis of them is beyond the scope of this paper. I will do no more than sketch an outline of the relevant facts here.

Apparently the only consonant clusters allowed in Ponapean are homorganic nasal + C clusters and geminate sonorants[13] (cf. Itô 1988). Clusters which do not fall into these categories are done away either by changing one of the consonants or by inserting a vowel.

Consonant shifts apparently take place in reduplicated forms whenever the first consonant can change [nasal] or [lateral] features to get a well-formed cluster:

12. kak --> kangkak 'able'
    sel --> sensel 'tied'
    lirohro --> lillirohro 'overprotective'

This mechanism will not be important for our concerns, and I list it here only for completeness' sake.

Rehg and Sohl (1981) then identify three distinct types of vowel insertion to break up illicit clusters that cannot be made licit by consonant shifts. They refer to these as copy vowels, epenthetic vowels, and base vowels, and to minimize confusion I will simply adopt their terminology here.

Copy vowels involving copying of a following vowel:

13. a. ak+pwung --> akupwung 'petty'
       ak+tantat --> akatantat 'to abhor'
    b. tang+da --> tangada 'run up'
       tang+do --> tangodo 'run here'
       tang+wei --> tangewei 'run there'

Epenthetic vowels are high vowels whose roundedness is determined by harmony with the following vowel or consonant (that is, the epenthetic vowel is /u/ before the labialized consonants *pw*

[13]Geminate obstruents are also found, in such loanwords as ***nappa*** 'Chinese cabbage' and ***kiassi*** 'catcher'.

and *mw* or before a syllable containing a rounded vowel, and /i/ otherwise)

14. pwihk+men -->pwihkimen 'a pig'
    ak+suwei --->akusuwei 'boastful'
    loang+loange -->loangiloange 'cross over'
    ngkoal -->ungkoal 'make sennit'
    skuhl -->sukuhl 'school'
    silk -->silik 'silk'

Rehg and Sohl (1981) note that these vowels are generally optional and never stressed, and that their quality is hard to determine. This last fact is particularly visible in their example ***loangiloange***, which is given as an example of an epenthetic vowel although it apparently violates their rule determining epenthetic vowel quality (the digraph ***oa*** represents the rounded vowel /ɔ/, and we would therefore expect an epenthetic vowel ***u***). This is not simply a matter of their having given us the wrong rule, as their example ***ungkoal*** 'make sennit' shows; rather, the difficulty of determining the vowel quality of an epenthetic vowel is apparently in play here. This will be worth bearing in mind as we look at vowel epenthesis in monosyllabic reduplication.

Finally, there are some cases of what Rehg and Sohl (1981) refer to as base vowels. They claim that base vowels are part of the underlying representation:

15. kihl+niki -->kiliniki 'to have skin'
    ngihl+niki -->ngileniki 'to have a voice'
    dihp+niki -->dipaniki 'to have sin'

Here there appears to be no way to predict the form of the inserted vowel from the phonological shape of the noun, so Rehg and Sohl claim that the underlying forms of these nouns are actually ***kili***,

ngile, and ***dipa***, with obligatory deletion of the last vowel in word-final position.

4.1.2 Overcopying in polysyllables

In polysyllable reduplication we can definitely see cases of copy vowels and epenthetic vowels used to break up impermissible clusters:

16. a. kadek-->kadakadek 'kind'
masukun -->masamasukun 'blind'
b. katohre -->katikatohre 'subtract'
rese -->resirese 'saw'

In fact, however, we also have the option of overcopying (Rehg 1984, 319):

17. sapeng -->sapesapeng 'answer'
sakone -->sakosakone 'force'
marep -->maremarep 'blink'

It is not clear what drives the choice among these options; there are a few near-minimal pairs such as ***sapesapeng*** 'answer' (with overcopying) vs. ***sipisiped*** 'brush off' (with either a copy vowel or an epenthetic vowel). Still, it is fairly clear that overcopying is among our options for dealing with phonotactic problems created by polysyllable reduplication. On the unmarked assumption that reduplication of ***alu*** and ***mpek*** is just like reduplication of any other polysyllable, then, we ought to see all three of these options surfacing in dealing with phonotactic problems; that is, these patterns ought to contain roots which involve overlong copying, roots which involve copy vowels, and roots with epenthetic vowels, just as the Pattern XI roots above do:

18. alu -->*aluwalu (overlong copying)
-->*alaalu (copy vowel)
-->aliyalu (epenthetic vowel)

But in fact we have only the last option. Let us take another look at the tables for ***alu*** and ***mpek***:

	m(pmpek)	m(pimpek)	m(pempek)	mpe(kmpek)
Cont	✓	✓	✓	✓
Anch	✓	✓	✓	✓
Fill	✓	✓	✓	✓
RED=Prefix	✓	✓	✓	✓
Align (Pwd, R, Ft, R)	✓	✓	✓	✓
lamb	*	*	*	*
B-D(V)	✓	*!	✓	✓
FtHd=H	✓	✓	✓	✓
Parse(σ,Ft)	*	*	*	**!
RED=$\sigma_{\mu\mu}$	✓	*	*!	*
Max	**	**	*	✓
B-D(C)	✓	✓	✓	✓

	al(yalu)	a(lalu)	a(laalu)	ali(alu)	alu(alu)
Cont	✓	✓	✓	✓	✓
Anch	✓	✓	✓	✓	✓
Fill	✓	✓	✓	✓	✓
RED=Prefix	✓	✓	✓	✓	✓
Align (Pwd, R, Ft, R)	✓	✓	✓	✓	✓
lamb	*	*	*	*	*
B-D(V)	✓	✓	*!	*!	✓
FtHd=H	*	*	✓	*	*
Parse(σ,Ft)	*	*	*	**	**!
RED=$\sigma_{\mu\mu}$	✓	*!	*	*	*
Max	*	*	*	*	✓
B-D(C)	*	✓	✓	✓	✓

We have seen that our tableaux as they stand predict phonotactically ill-formed reduplicated forms for these verbs; ***alyalu*** and ***mpmpek***. However, it now seems clear that we cannot solve this problem by simply adding a phonotactic constraint to rule out the winning candidate in these tableaux. Such a move would force us to come up

with some way of ruling out forms which are metrically identical to the attested forms but which involve overcopying as a means of dealing with phonotactic problems. We have just seen that any constraint we might invent to rule out such overcopying would be undesirable, since overcopying is indeed attested in Ponapean as a means of dealing with phonotactic problems. In the next section we will see that overcopying is available not only in polysyllables, but in apparent monosyllables as well; overcopying is thus quite generally an available option for dealing with phonotactic problems, and we should not create a constraint to rule it out.

4.1.3 Overcopying in monosyllables

In this section I will try to show that overcopying is an option in monosyllables as well; that is, that there are lexical differences between monosyllabic roots having to do with some structure at the right edge of the root, and that this structure is sometimes copied in reduplication in order to avoid phonotactic problems. In monosyllable reduplication there are numerous fairly clear cases of the use of copy (19) and epenthetic (2Ø) vowels to break up impermissible clusters:

19. tang -->tangatang 'run'
    rot -->rotorot 'dark'
    tep -->tepetep 'kick'
    kos -->kosokos 'throw'
2Ø. seng -->sengiseng 'cry'
    ped -->pediped 'be squeezed'
    tep -->tepitep 'begin'
    net -->netinet 'sell'

There are also a couple of verb stems in the available data which bear what might be considered base vowels:

21. lop-->lopilop 'cut'
    kos-->kosikos 'bent'

The inserted *i* in these forms is clearly not a copy vowel, and is apparently not an epenthetic vowel, either, since it fails to harmonize in roundedness with the following vowel. On the other hand, we have seen that there is at least one case in Rehg and Sohl (1981) which is identified as an epenthetic vowel but transcribed as failing to harmonize in roundedness with the following vowel (namely, ***loangiloange*** 'cross over', which we would expect to surface as ***loanguloange***). Thus, these are either base vowels or non-harmonizing epenthetic vowels.

The decision as to whether a monosyllabic stem will use copy vowels or epenthetic vowels is apparently lexically determined. We can find minimal pairs such as those in 22:

22. a. tep-->tepitep 'begin'
       tep-->tepetep 'kick'
    b. kos-->kosikos 'bent'
       kos-->kosokos 'throw'

Furthermore, roots are consistent in their use of one or another of these strategies:

23. a. tangewei 'run there'
       tangodo 'run here'
       tangada 'run up'
       tangatang 'running'
    b. rotada 'get dark'
       rotorot 'being dark'
    c. lusida 'jump up'
       lusisang 'jump from'
       lusulus 'jumping'

Thus, roots can apparently be divided reliably into "copy-vowel roots" (such as ***tang*** and ***rot***) and "epenthetic-vowel roots" (such as ***lus***); there are no roots that use epenthetic vowels for reduplication and copy vowels for suffixation, for example. Furthermore, the lexical difference between roots apparently has something to do

with the right edge, since the rules for determining which type of vowel insertion is used to break up consonant clusters created by prefixes is quite different (see Rehg and Sohl 1981, 92 for discussion), so that a root may use copy vowels for suffixes and epenthetic vowels for prefixes, as below:

24. tikitik+la -->tikitikala 'become small'
ak+tikitik -->ak(i)tikitik 'humble'

These facts seem to suggest that monosyllabic roots are lexically distinguished in some way which has to do with the right edge of the root; perhaps some roots end in an underspecified vowel and others do not. Given this, the fact that the vowel appearing in reduplication is identical to that appearing in suffixation for a given root (cf. the data in 23) suggests that reduplication involves overcopying; the quality of the right edge that determines which type of vowel is inserted is being copied in the prefixal reduplicant, giving rise to a reduplicant that is more than one syllable in length.

Thus, overcopying is apparently also an option in monosyllables, though less transparently than in polysyllables. The fact that overcopying appears to quite generally be an option in dealing with phonotactic constraints suggests strongly that we cannot rule out overcopying in principle in dealing with the phonotactic problems raised by Patterns VII and X. It seems that we are better off, then, allowing the phonotactically problematic forms predicted by the tableaux above to be generated, and breaking up the illicit clusters at a later level.

4.1.4 Paring away unwanted options

We have seen that we might expect to find vowel-initial polysyllabic roots to have any of three options for dealing with phonotactic problems, as consonant-initial polysyllabic roots do:

25. alu -->*aluwalu (overlong copying)
-->*alaalu (copy vowel)
-->aliyalu (epenthetic vowel)

Michael Kenstowicz (p.c.) has suggested an alternative way of ruling out the first two options above. He proposes a phonotactic constraint on contact between vowels, requiring them not to differ in place. The reduplicant in this case, then, is ***a1V'***-, with an underspecified final vowel lacking a place feature. The proposed constraint is then met; the final vowel of the reduplicant and the first vowel of the base do not differ in place, since the first vowel is unspecified for place. The underspecified vowel is then realized as [i] by default. Thus, a violation of whatever requirement drives copying of vocalic features (presumably Max, or perhaps a more specific version of Max pertaining exclusively to vocalic features) is incurred in order to avoid a violation of a phonotactic constraint on vowels[14].

This solution seems to me to still involve the use of multiple levels. Depending on our assumptions about the nature of default vowels, it looks as though the phonotactic constraint that forces underparsing of the second vowel of the reduplicant may be violated at the end of the derivation, when the vowel becomes specified for place by default.

---

[14] Rehg's (1984) discussion of epenthetic vowels as "reduced" or incompletely copied versions of full vowels is similar in spirit.

More importantly, even if we adopt this account we will still need to come up with some way of ruling out ***alalu***. Let us turn to this problem.

4.2 Ruling out *alalu*

Let us look again at the tableau for ***alu***:

RED+alu-->alialu

	al(yalu)	a(lalu)	a(laalu)	ali(alu)	alu(alu)
Cont	✓	✓	✓	✓	✓
Anch	✓	✓	✓	✓	✓
Fill	✓	✓	✓	✓	✓
RED=Prefix	✓	✓	✓	✓	✓
Align (Pwd, R, Ft, R)	✓	✓	✓	✓	✓
lamb	*	*	*	*	*
B-D(V)	✓	✓	*!	*!	✓
FtHd=H	*	*	✓	*	*
Parse(σ,Ft)	*	*	*	**	**!
RED=$\sigma_{\mu\mu}$	✓	*!	*	*	*
Max	*	*	*	*	✓
B-D(C)	*	✓	✓	✓	✓

Suppose we add a phonotactic constraint to rule out the first candidate. Even if we can find some way of explaining the discrepancies between the fifth candidate and the actual output (perhaps following Kenstowicz' suggestion), we will still need to rule out the second candidate, which is superior to the fifth with regard to Parse. Thus, we will need to have some constraint that outranks Parse and prefers ***alValu*** to ***alalu***.

It seems to me that accounts along these lines will generally have difficulties dealing simultaneously with ***alialu*** and with *luluhmwuhmw*:

RED+luhmwuhmw-->luluhmwuhmw

	$(luluh)(mwuhmw)	luhluh(mwuhmw)	(luh)(luh)(mwuhmw)
Cont	✓	✓	✓
Anch	✓	✓	✓
Fill	✓	✓	✓
RED=Prefix	✓	✓	✓
Align R	✓	✓	✓
Iamb	✓*	*	***!
B-D(V)	✓	✓	✓
*V=RB	✓	✓	✓
FtHd=H	✓	✓	✓
Parse(σ,Ft)	✓	**!	✓
RED=$\sigma_{\mu\mu}$	*	✓	✓
Max	*****	****	****
B-D(C)	✓	✓	✓

The behavior of ***luhmwuhmw*** shows that Ponapean reduplicants may be made smaller than a bimoraic syllable in order to minimize Parse violations; that is, Parse outranks both RED=$\sigma_{\mu\mu}$ and Max. This leaves us with the question of why we may not minimize Parse violations in the case of ***alu*** by prefixing a monomoraic syllable to get ***alalu***.

We cannot create a new constraint preferring reduplicants consisting of two light syllables to ones consisting of one light syllable (RED=$\sigma_{\mu}\,\sigma_{\mu}$). As we have seen, the constraint in question will have to outrank Parse, and Parse outranks RED=$\sigma_{\mu\mu}$, so such a move would have undesirable effects on patterns like XI (***rere*** 'skin, peel'-->***rerrere***), where the reduplicant cannot create an iamb and thus appears as a single bimoraic syllable. We would expect the form to obey the hypothetical new constraint RED=$\sigma_{\mu}\,\sigma_{\mu}$, giving ****rererere***.

Given that obedience to RED=$\sigma_{\mu\mu}$ is possible for ***rere*** and impossible (for phonotactic reasons) for ***alu***, we might try getting

around this by having a constraint outranking Parse which is obeyed by both *rer-* and *ali-*. For example, we might divide RED=$\sigma_{\mu\mu}$ into RED=μμ (obeyed by bimoraic syllables and pairs of monomoraic syllables) and a lower-ranked RED=σ (obeyed by monomoraic and bimoraic syllables)[15]. RED=μμ could outrank Parse without adversely affecting *rere*, since the attested reduplicant *rer-* obeys RED=μμ. But now we run into the difficulty mentioned above with *luhmwuhmw*. Apparently Ponapean allows the reduplicant to be a monomoraic syllable in order to minimize Parse violations; thus, RED=μμ cannot outrank Parse.

Another possibility would involve altering the metrical constraints to favor parsing of bisyllabic non-iambic feet over failure to parse, while still preferring iambs over non-iambs. This could be accomplished fairly straightforwardly by dividing Iamb into two constraints, one requiring that feet be binary and the other that they be right-headed. Thus, (*ali*)(*alu*) would be preferred over (*alalu*), *a*(*lalu*) or (*a*)(*lalu*). However, we can see that the behavior of Pattern VII is not driven simply by a need to create words with even numbers of syllables, since words like *ewetik* 'abstemious', which ought to surface as **ewewetik* (metrically identical to *alialu*), in fact yield the five-syllable form *ewiewetik*. We would need to create a new constraint, then, to rule out the metrically preferable **ewewetik*, and this constraint will have to outrank Parse, since in a system that allows bisyllabic non-iambic feet (*ewe*)(*wetik*) will

---

[15] Something like this may be necessary in any event to ensure that phonotactic problems are dealt with by overcopying rather than undercopying; *marep* 'blink'-->*maremarep*, not **mamarep*. Thus, it is apparently more important that the reduplicant be bimoraic than that it be a syllable.

be preferable to (*ewi*)*e*(*wetik*) with regard to Parse. It is difficult to see what form such a constraint might take. The forms in question differ, for example, in that the first foot of (*ewe*)(*wetik*) consists partly of the reduplicant and partly of the base, while the first foot of (*ewi*)*e*(*wetik*) consists entirely of the reduplicant; we might consider creating a constraint *Ft=RB, parallel to *V=RB, to take advantage of this fact. But we have just seen that *Ft=RB will have to outrank Parse, and it will therefore rule out (*luluh*)(*mwuhmw*) in favor of ***luhluh**(*mwuhmw*).

What we have seen is that some constraint that outranks Parse and prefers ***alialu*** to ***alalu*** needs to be in play, or else Parse needs to be interpreted in such a way that it does not prefer ***alalu*** to ***alialu***, as it currently does. It is difficult to see how this can be accomplished without adversely affecting the accounts of other patterns. The behavior of *luhmwuhmw*, in particular, shows us that a monomoraic reduplicant may be employed to minimize Parse violations, and this is similar enough to the procedure involved in constructing the unattested ***alalu*** that specifying a constraint to rule out the latter while allowing the former is not a trivial matter.

The problem is exacerbated somewhat by the standard optimality-theoretic methodology for dealing with metrification, which involves separating the constraints driving syllable parsing (e.g., Parse, Align (Pwd, R, Ft, R), etc.) from those constraining the possible forms of feet (e.g., Iamb, FtBin) (cf. Buckley (1994), Cohn and McCarthy (1994), Kenstowicz (1994), and McCarthy and Prince (1993), among many others, for accounts pursuing this line). This approach makes it impossible to make the creation of an initial iamb

in *luhmwuhmw*-->*luluhmwuhmw* especially "valuable"; if the constraint we use to rule out ***alalu*** outranks Parse, it will rule out creation of an initial iamb in the case of *luhmwuhmw*, even though iamb creation is not at issue in the case of ***alu***. If we were to separate Parse into constraints $\text{Parse}_{\text{iamb}}$, $\text{Parse}_{\text{trochee}}$, etc., each of which would drive parsing into specific kinds of feet, the problem would be comparatively simple to solve, since we could order a new constraint with respect to the Parse constraint relevant to ***alu*** without necessarily affecting *luhmwuhmw*. I will not explore this option further here, since it involves considerable complications of the OT approach to metrification which I am not in a position to independently motivate.

5. The additional level

The theory developed above generates ***alyalu*** as the reduplicated form of ***alu*** 'walk'. I have argued above that this is the correct result, and that vowel epenthesis to yield ***alialu*** should be handled at a subsequent level of representation. In this section I will briefly explore the nature of this level.

It seems clear that the new level cannot simply be a result of passing the output of the old level through the same constraints in the same order. Consider the tableau for ***alu*** again:

RED+alu-->alialu

	al(yalu)	a(lalu)	a(laalu)	ali(alu)	alu(alu)
Cont	✓	✓	✓	✓	✓
Anch	✓	✓	✓	✓	✓
Fill	✓	✓	✓	✓	✓
RED=Prefix	✓	✓	✓	✓	✓
Align (Pwd, R, Ft, R)	✓	✓	✓	✓	✓
lamb	*	*	*	*	*
B-D(V)	✓	✓	*!	*!	✓
FtHd=H	*	*	✓	*	*
Parse(σ,Ft)	*	*	*	**	**!
RED=$\sigma_{\mu\mu}$	✓	*!	*	*	*
Max	*	*	*	*	✓
B-D(C)	*	✓	✓	✓	✓

The conclusion of the arguments above was that the first candidate should win, as shown. Any phonotactic constraint ruling out forms like ***alyalu***, then, will have to be ordered below RED=$\sigma_{\mu\mu}$. If ***alyalu*** is simply passed through the constraints again, no epenthesis will occur, since RED=$\sigma_{\mu\mu}$ will still favor ***alyalu*** over ***alialu***, and RED=$\sigma_{\mu\mu}$, by hypothesis, outranks the phonotactic constraint that prefers ***alialu*** over ***alyalu***.

There are at least two possible ways of avoiding this result. One is to simply reorder the constraints on the new level so that the phonotactic constraint driving epenthesis outranks RED=$\sigma_{\mu\mu}$. This seems like an undesirable step, since it would increase the power of the theory considerably to allow distinct levels to have different constraint orderings.

Another possible move would be to introduce some equivalent of bracket erasure; to assume, that is, that ***alyalu*** is not "recognizable" as a reduplicated form on the new level. As a result,

the constraint RED=$\sigma_{\mu\mu}$ is irrelevant on the new level, and is satisfied vacuously. The phonotactic constraint can then trigger epenthesis, and the correct output form is derived. This move is consistent with Kenstowicz' (1994) conclusion, based on investigation of metrical phenomena in Carib, that a given level generally fails to make distinctions between underlyingly present structure and structure inserted at the previous level. This move seems to be the most conservative way of dealing with the data discussed here, since we can now maintain the claim that constraint orderings are constant for a given language, rather than allowing them to be differently ranked on different levels.

One difficulty facing this solution concerns the status of Fill. Vowel epenthesis will violate Fill, after all, and we have seen that Fill is quite highly ranked in this language; certainly it outranks RED=$\sigma_{\mu\mu}$, which appears to outrank the phonotactic constraint that drives epenthesis.

One way out of this would involve postulating two distinct Fill constraints. The types of Fill violations that have been relevant thus far have involved adding a mora to an existing syllable in order to improve the representation with respect to lamb. Thus, Fill was highly ranked in order to avoid forms like ***lalaAl*** as the reduplicated form of ***lal***; adding an additional mora to the base in order to make it into a heavy syllable had to be impermissible.

In this case of vowel epenthesis in ***alialu***, on the other hand, we are constructing a new syllable. We might exploit this by splitting Fill into a high-ranking Fill$_{\mu}$ which outranks lamb and rules out ***lalaAl*** and a lower-ranking Fill$_{\sigma}$ which militates against

constructing a new syllable in ***alialu*** but is overridden by a phonotactic constraint.

It is interesting to note that insertion of ***i*** and ***u*** to break up illicit clusters may conform to the general characteristics of the Word Level of Lexical Phonology (cf. Booij and Rubach 1987, Borowsky 1992). The phenomenon appears not to be cyclic, since, as we have seen, it applies to illicit clusters in foreign borrowings:

26. skuhl -->sukuhl 'school'
    skru -->sukuru 'screw'
    sdamp -->sidamp 'stamp'
    silk -->silik 'silk'

Whether vowels are inserted between consonants brought together by contact between words is less clear. Rehg and Sohl (1981, 62-63) do mention that such clusters can be made acceptable by nasal substitution in some cases:

27. a. E kalap pahn soupisek
       [e kalam pahn soupisek]
    'He will always be busy'
    b. E saik kengwini
       [e saing kengwini]
    'He hasn't yet taken medicine'

They then point out that nasal substitution of this kind does not occur with coronals:

28. Ke meid dangahnga
    [ke meid dangahnga]
    *[ke mein dangahnga]
    'Aren't you lazy!'

Finally, they note that coronal clusters are broken up by vowel insertion within a word:

29. a. wied+da-->wiedida 'proceed upward'

b. lus+sang-->lusisang 'jump from'

This seems to suggest that vowel insertion is not employed between words; if it were, Rehg and Sohl would presumably mention this at this point, rather than specifically noting that vowel insertion is employed to break up coronal clusters within a word.

The framework of Lexical Phonology, then, would be able to predict that insertion of ***i*** or ***u*** takes place after reduplication, since Word Level rules are claimed in that framework to apply after cyclic lexical rules. Replicating this kind of result will be difficult, it seems to me, in Optimality Theory. We might try to capture the basic insight of Lexical Phonology, that non-cyclic rules will always follow cyclic ones, by postulating a set of constraints that make reference to morphological boundaries of various kinds[16]. Such constraints will have the effect of rules that apply only in derived contexts, since they require the presence of a morphological boundary in order to have any effect. Coupled with the notion of "bracket erasure" described above, such constraints will always have their effects on the level of representation at which morphemes are concatenated, since at later levels of representation the morphological boundaries that appear in the structural descriptions of the constraints will have been erased.

On the other hand, phenomena triggered by the constraints that make no reference to morphological boundaries (the "noncyclic"

---

[16] Thanks to Martha McGinnis for suggesting this to me.

constraints) can take place at any level. Consider a case in which a "noncyclic" constraint N and a "cyclic" constraint C make conflicting demands on GEN; only one of the constraints can be satisfied. Suppose further that N outranks C. Then the operation in GEN triggered by C will not appear if this would violate N; rather, GEN will alter the structure in whatever way is necessary to satisfy N. Translating this into a rule-based framework, this means that the rule simulated by N would bleed the one simulated by C; to put it another way, the noncyclic rule N would precede the cyclic rule C. Optimality Theory has no principled way of avoiding this result, it seems to me, apart from extrinsically ordering all "cyclic" constraints above "noncyclic" ones.

6. Conclusion

I have tried to show that the Ponapean reduplication system can be fairly straightforwardly analyzed in Optimality-theoretic terms, simply by assuming a particular ordering of mostly familiar constraints. What makes Ponapean reduplication unusual, in this framework, is the relatively high ranking of various metrical constraints which override constraints on the shape of the reduplicant and on the intrusion of new material into the reduplicant which fails to correspond to anything in the base. Given the particular constraints assumed in this framework, in fact, we effectively predict the existence of a language like Ponapean.

I have also tried to show that the best optimality-theoretic analysis of Ponapean reduplication will involve multiple levels of representation. If my analysis is correct, Optimality Theory will

apparently have to retreat from a stance of forbidding intermediate levels of representation altogether to one of minimizing their role.

Bibliography

Booij, Geert, and Jerzy Rubach. 1987. Postcyclic versus Postlexical Rules in Lexical Phonology. Linguistic Inquiry 18.1-44.

Borowsky, Toni. 1992. On the Word-Level. Studies in Lexical Phonology, ed. by S. Hargus and E. Kaisse. San Diego: Academic Press.

Buckley, Eugene. 1994. Alignment in Manam Stress. ms., University of Pennsylvania.

Cohn, Abigail, and John McCarthy. 1994. Alignment and Parallelism in Indonesian Phonology. ms., Cornell University and University of Massachusetts at Amherst.

Everett, Daniel. 1994. Quantity, Sonority, and Alignment Constraints in Arawan Prosody. handout of talk given at MIT.

Itô, Junko. 1988. Syllable Theory in Prosodic Phonology. New York: Garland Publishing, Inc.

Kenstowicz, Michael. 1994. Cyclic vs. Noncyclic Constraint Evaluation. Papers on Phonology and Morphology. MIT Working Papers in Linguistics, v. 21.

Levin, Juliette. 1985. A Metrical Theory of Syllabicity. Ph.D. Dissertation, MIT.

McCarthy, John, and Alan Prince. 1986. Prosodic Morphology. ms., U Mass Amherst and Brandeis University.

McCarthy, John, and Alan Prince. 1993. Generalized Alignment. ms., U Mass Amherst and Rutgers University.

McCarthy, John, and Alan Prince. 1994. The Emergence of the Unmarked: Optimality in Prosodic Morphology. ms., U Mass Amherst and Rutgers University.

Rehg, Kenneth. 1984. Nasal Substitution Rules in Ponapean. in Byron W. Bender, ed., Studies in Micronesian Linguistics. Pacific Linguistics Series C, no. 80. Canberra: The Australian National University.

Rehg, Kenneth, and Damian Sohl. 1979. Ponapean-English Dictionary. Honolulu: The University Press of Hawaii.

Rehg, Kenneth, and Damian Sohl. 1981. Ponapean Reference Grammar. Honolulu: The University Press of Hawaii.

Appendix A: Constraint Definitions

Align (Pwd, R, Ft, R)--(see McCarthy and Prince 1993 for discussion) All phonological words must end in a foot. Violated by phonological words ending in unparsed syllables.

Anchoring--(McCarthy and Prince 1994, 8) Correspondence preserves alignment in the following sense: the left (right) peripheral element of R[eduplicant] corresponds to the left (right) peripheral element of B[ase], if R is to the left (right) of B. For prefixal reduplicants, this is violated by reduplicants whose first segment does not correspond to the first segment of the base, e.g., *bratu-->ratu-bratu*.

Base-Dependence (V, C)--(based on McCarthy and Prince 1994, 9) Every (vocalic, consonantal) element of R has a correspondent in B. Violated by reduplicants containing vowels or consonants with no correspondent in the base, e.g. Ponapean *ahn-->aiahn*, where /i/ has no correspondent in the base.

Contiguity--(from McCarthy and Prince 1994, 8) The portion of the base standing in correspondence forms a contiguous string, as does the correspondent portion of the reduplicant. Violated by reduplication that skips segments of the base, e.g. Tagalog *blowawt-->boblowawt*.

Fill--(McCarthy and Prince 1994, 11) Epenthetic structure is prohibited.

FtHd=H--The head of a foot must be a metrically heavy syllable.

Iamb--(adapted from McCarthy and Prince 1993, 10)

Ft-->σ_W σ_R

Violated by feet that do not consist of a light syllable followed by a heavy syllable.

Max--(McCarthy and Prince 1994, 9) Every element of B has a correspondent in R. Violated by reduplicants which do not contain correspondents for every element in the base, e.g. Ponapean *duhp-->duduhp*.

Parse (σ,Ft)--(McCarthy and Prince 1993, 10) All syllables must be parsed by feet.

RED=$\sigma_{\mu\mu}$--(based on McCarthy and Prince 1994, 18) The reduplicant is a bimoraic syllable. Violated by reduplicants which are larger or smaller than a bimoraic syllable, e.g., Ponapean *duhp-->duduhp*.

RED=Prefix (equivalent to the Align (R, Left, Stem, Left) constraint of McCarthy and Prince 1994, 14) R is a prefix.

***V=RB**--(defined in this paper, p. 22)

A vowel of the form

is ill-formed.

Appendix B: Full tableaux and constraint orderings

In the preceding paper I have tried to sketch a general outline of my proposal for dealing with Ponapean aspectual reduplication within an Optimality framework. Here I show in more detail the crucial constraint orderings involved.

Pattern I: lal-->lallal

	$(lallal)	(lalal)	(lalaAl)	(lalaal)
Cont	✓	✓	✓	✓
Anch	✓	✓	✓	*!
Fill	✓	✓	*!	✓
RED=Prefix	✓	✓	✓	*
Align R	✓	✓	✓	✓
lamb	*	*	✓	✓
B-D(V)	✓	✓	✓	*
*V=RB	✓	✓	✓	✓
FtHd=H	✓	*!	✓	✓
Parse(σ,Ft)	✓	✓	✓	✓
RED=$\sigma_{\mu\mu}$	✓	*	*	✓
Max	✓	*	*	*
B-D(C)	✓	✓	✓	✓

Fill, Max, or RED=$\sigma_{\mu\mu}$>>lamb
lamb>>Max (Pattern IV)
lamb>>RED=$\sigma_{\mu\mu}$ (Pattern III)
∴ Fill>>lamb
Anch, RED=Prefix, B-D(V), or Max>>lamb
lamb>>Max (Pattern IV)
lamb>>B-D(V) (Pattern III)
∴ Anch or RED=Prefix>>lamb

Pattern II: pa-->pahpa

	$(paapa<V>)	(papa<V>)	(papaa)	(papaa)
Cont	✓	✓	✓	✓
Anch	✓	✓	✓	✓
Fill	✓	✓	*!	✓
RED=Prefix	✓	✓	✓	*!
Align R	✓	✓	✓	✓
lamb	*	*	✓	✓
B-D(V)	✓	✓	✓	✓
*V=RB	✓	✓	✓	✓
FtHd=H	✓	*!	✓	✓
Parse(σ,Ft)	✓	✓	✓	✓
RED=$\sigma_{\mu\mu}$	✓	*	*	✓
Max	✓	*	*	✓
B-D(C)	✓	✓	✓	✓

RED=Pref>>lamb

Pattern III: el-->elehl

	$(eleel)	(elel)	(eelel)
Cont	✓	✓	*!
Anch	✓	✓	✓
Fill	✓	✓	✓
RED=Prefix	✓	✓	✓
Align R	✓	✓	✓
lamb	✓	*!	*
B-D(V)	*	✓	*
*V=RB	*	✓	✓
FtHd=H	✓	*	✓
Parse(σ,Ft)	✓	✓	✓
RED=$\sigma_{\mu\mu}$	*	*	✓
Max	✓	✓	✓
B-D(C)	✓	✓	✓

lamb or FtHd=H>>B-D(V) and *V=RB

*V=RB>>FtHd=H (Pattern VII, ***amiamas*** dialect)

∴lamb>>*V=RB

B-D(V)>>FtHd=H (Pattern VII, ***alialu***)

∴lamb>>B-D(V)

Cont or lamb>>*V=RB and RED=$\sigma_{\mu\mu}$

Pattern IV: ahn-->aiahn

	$(aiaan)	(_anaan)	(a_naan)	(aan)(aan)
Cont	✓	✓	*!	✓
Anch	✓	*!	✓	✓
Fill	✓	✓	✓	✓
RED=Prefix	✓	✓	✓	✓
Align R	✓	✓	✓	✓
lamb	✓	✓	✓	**!
B-D(V)	✓	✓	✓	✓
*V=RB	✓	✓	✓	✓
FtHd=H	✓	✓	✓	✓
Parse(σ,Ft)	✓	✓	✓	✓
RED=$\sigma_{\mu\mu}$	*	*	*	✓
Max	**	*	*	✓
B-D(C)	*	✓	✓	✓

Anch>>Max and B-D(C)
Cont>>Max and B-D(C)
lamb>>RED=$\sigma_{\mu\mu}$, Max, and B-D(C)

Pattern VI: duhp-->duduhp

	$(duduup)	(duu)(duup)	(duup)(duup)
Cont	✓	✓	✓
Anch	✓	✓	✓
Fill	✓	✓	✓
RED=Prefix	✓	✓	✓
Align R	✓	✓	✓
lamb	✓	*!	**!
B-D(V)	✓	✓	✓
*V=RB	✓	✓	✓
FtHd=H	✓	✓	✓
Parse(σ,Ft)	✓	✓	✓
RED=$\sigma_{\mu\mu}$	*	✓	✓
Max	**	*	✓
B-D(C)	✓	✓	✓

lamb>>RED=$\sigma_{\mu\mu}$, Max

Pattern VII: alu-->alialu

	$\underline{a}$l($\underline{u}$alu)	$\underline{a}$(l$\underline{a}$lu)	$\underline{a}$(l$\underline{aa}$lu)	$\underline{ali}$(alu)	$\underline{alu}$(alu)
Cont	✓	✓	✓	✓	✓
Anch	✓	✓	✓	✓	✓
Fill	✓	✓	✓	✓	✓
RED=Prefix	✓	✓	✓	✓	✓
Align (Pwd, R, Ft, R)	✓	✓	✓	✓	✓
lamb	*	*	*	*	*
B-D(V)	✓	✓	*!	*!	✓
*V=RB	✓	✓	*	✓	✓
FtHd=H	*	*	✓	*	*
Parse(σ,Ft)	*	*	*	**	**!
RED=$\sigma_{\mu\mu}$	✓	*!	*	*	*
Max	*	*	*	*	✓
B-D(C)	*	✓	✓	✓	✓

RED=$\sigma_{\mu\mu}$>>B-D(C)

B-D(V), *V=RB, or RED=$\sigma_{\mu\mu}$>>FtHd=H and B-D(C)

FtHd=H>>Parse(σ,Ft) (Pattern IX, *duhpek*)

Parse(σ,Ft)>>RED=$\sigma_{\mu\mu}$ (Pattern IX, *luhmwuhmw*)

∴FtHd=H>>RED=$\sigma_{\mu\mu}$

∴B-D(V) or *V=RB>>FtHd=H

B-D(V)>>Parse(σ,Ft) (Pattern VII, *arekarek*)

Parse(σ,Ft)>>*V=RB (Pattern VII, *ewehwetik*)

∴B-D(V)>>*V=RB

∴B-D(V)>>FtHd=H

B-D(V) or RED=$\sigma_{\mu\mu}$>>B-D(C)

B-D(V)>>Parse(σ,Ft) (Pattern VII, *arekarek*)

Parse(σ,Ft)>>RED=$\sigma_{\mu\mu}$ (Pattern IX, *luhmwuhmw*)

∴B-D(V)>>RED=$\sigma_{\mu\mu}$

∴B-D(V)>>B-D(C)

B-D(V), Parse(σ,Ft), or RED=$\sigma_{\mu\mu}$>>B-D(C)

Parse(σ,Ft) or RED=$\sigma_{\mu\mu}$>>Max and B-D(C)

Parse(σ,Ft)>>RED=$\sigma_{\mu\mu}$ (Pattern IX, *luhmwuhmw*)

∴Parse(σ,Ft)>>Max and B-D(C)

Pattern VII: arekarek-->ariarekarek

	$aryare (karek)	arare (karek)	(araa)re (karek)	ariare (karek)	areare (karek)
Cont	✓	✓	✓	✓	✓
Anch	✓	✓	✓	✓	✓
Fill	✓	✓	✓	✓	✓
RED=Prefix	✓	✓	✓	✓	✓
Align (Pwd, R, Ft, R)	✓	✓	✓	✓	✓
lamb	*	*	*✓	*	*
B-D(V)	✓	✓	*!	*!	✓
FtHd=H	*	*	✓*	*	*
Parse(σ,Ft)	***	***	*	****	****!
RED=$\sigma_{\mu\mu}$	✓	*!	*	*	*
Max	******	******	******	******	*****
B-D(C)	*	✓	✓	✓	✓

RED=$\sigma_{\mu\mu}$>>B-D(C)

B-D(V) or RED=$\sigma_{\mu\mu}$>>Parse(σ,Ft) and B-D(C)

Parse(σ,Ft)>>RED=$\sigma_{\mu\mu}$ (Pattern IX, *luhmwuhmw*)

B-D(V)>>Parse(σ,Ft) and B-D(C)

Parse(σ,Ft) or RED=$\sigma_{\mu\mu}$>>Max and B-D(C)

Parse(σ,Ft)>>RED=$\sigma_{\mu\mu}$ (Pattern IX, *luhmwuhmw*)

∴Parse(σ,Ft)>>Max and B-D(C)

Pattern VII, *amiamas* dialect: amas-->amiamas

	a(mamas)	a(maamas)	$am(yamas)
Cont	✓	✓	✓
Anch	✓	✓	✓
Fill	✓	✓	✓
RED=Prefix	✓	✓	✓
Align (Pwd, R, Ft, R)	✓	✓	✓
Iamb	*	*	*
B-D(V)	✓	✓	✓
*V=RB	✓	*!	✓
FtHd=H	*	✓	*
Parse(σ,Ft)	*	*	*
RED=$\sigma_{\mu\mu}$	*!	*	✓
Max	**	*	**
B-D(C)	✓	✓	*

RED=$\sigma_{\mu\mu}$>>B-D(C)

*V=RB or RED=$\sigma_{\mu\mu}$>>FtHd=H, Max, and B-D(C)

FtHd=H>>Parse(σ,Ft) (Pattern IX, ***duhpek***)

Parse(σ,Ft)>>RED=$\sigma_{\mu\mu}$ (Pattern IX, *luhmwuhmw*)

∴*V=RB>>FtHd=H, Max, and B-D(C)

Pattern VII, *ewiewetik* dialect: ewetik-->ewiewetik

	ewe(wetik)	(ewee) (wetik)	$ewye (wetik)
Cont	✓	✓	✓
Anch	✓	✓	✓
Fill	✓	✓	✓
RED=Prefix	✓	✓	✓
Align (Pwd, R, Ft, R)	✓	✓	✓
lamb	*	*✓	*
B-D(V)	✓	✓	✓
*V=RB	✓	*!	✓
FtHd=H	*	✓*	*
Parse(σ,Ft)	**	✓	**
RED=$\sigma_{\mu\mu}$	*!	*	✓
Max	****	***	****
B-D(C)	✓	✓	*

RED=$\sigma_{\mu\mu}$>>B-D(C)

*V=RB or RED=$\sigma_{\mu\mu}$>>Parse(σ,Ft), Max, and B-D(C)

Parse(σ,Ft)>>RED=$\sigma_{\mu\mu}$ (Pattern IX: *luhmwuhmw*)

∴ *V=RB>>Parse(σ,Ft), Max, and B-D(C)

Pattern VII, *amahmas* dialect: amas-->amahmas

	a(mamas)	$a(maamas)	am(yamas)
Cont	✓	✓	✓
Anch	✓	✓	✓
Fill	✓	✓	✓
RED=Prefix	✓	✓	✓
Align (Pwd, R, Ft, R)	✓	✓	✓
lamb	*	*	*
B-D(V)	✓	✓	✓
FtHd=H	*!	✓	*!
Parse(σ,Ft)	*	*	*
*V=RB	✓	*	✓
RED=$\sigma_{\mu\mu}$	*	*	✓
Max	**	*	**
B-D(C)	✓	✓	*

FtHd=H or Max>>*V=RB

FtHd=H, Max, or B-D(C)>>*V=RB and RED=$\sigma_{\mu\mu}$

RED=$\sigma_{\mu\mu}$>>Max (Pattern X)

RED=$\sigma_{\mu\mu}$>>B-D(C) (Pattern VII: ***alu***)

∴FtHd=H>>*V=RB and RED=$\sigma_{\mu\mu}$

Pattern VII, *ewehwetik* dialect: ewetik-->ewehwetik

	ewe(wetik)	$(ewee)(wetik)	ewue(wetik)
Cont	✓	✓	✓
Anch	✓	✓	✓
Fill	✓	✓	✓
RED=Prefix	✓	✓	✓
Align (Pwd, R, Ft, R)	✓	✓	✓
lamb	*	*✓	*
B-D(V)	✓	✓	✓
FtHd=H	*	✓*	*
Parse(σ,Ft)	**!	✓	**!
*V=RB	✓	*	✓
RED=$\sigma_{\mu\mu}$	*	*	✓
Max	****	***	****
B-D(C)	✓	✓	*

Parse(σ,Ft) or Max>>*V=RB

Parse(σ,Ft)>>Max (Pattern IX, *luhmwuhmw*)

∴Parse(σ,Ft)>>*V=RB

Parse(σ,Ft), Max, or B-D(C)>>*V=RB and RED=$\sigma_{\mu\mu}$

RED=$\sigma_{\mu\mu}$>>B-D(C) (Pattern VII, *alu*)

∴Parse(σ,Ft) or Max>>*V=RB and RED=$\sigma_{\mu\mu}$

Parse(σ,Ft)>>Max (Pattern IX, *luhmwuhmw*)

∴Parse(σ,Ft)>>*V=RB and RED=$\sigma_{\mu\mu}$

Pattern VIII: liahn-->lihliahn

	$liy(liyaan)	li(liyaan)
Cont	✓	✓
Anch	✓	✓
Fill	✓	✓
RED=Prefix	✓	✓
Align R	✓	✓
lamb	✓	✓
B-D(V)	✓	✓
*V=RB	✓	✓
FtHd=H	✓	✓
Parse(σ,Ft)	*	*
RED=$\sigma_{\mu\mu}$	✓	*!
Max	***	****
B-D(C)	✓	✓

Pattern IX: luhmwuhmw-->luluhmwuhmw

	$(luluh) (mwuhmw)	luhluh (mwuhmw)	(luh)(luh) (mwuhmw)
Cont	✓	✓	✓
Anch	✓	✓	✓
Fill	✓	✓	✓
RED=Prefix	✓	✓	✓
Align R	✓	✓	✓
lamb	✓*	*	***!
B-D(V)	✓	✓	✓
*V=RB	✓	✓	✓
FtHd=H	✓	✓	✓
Parse(σ,Ft)	✓	**!	✓
RED=$\sigma_{\mu\mu}$	*	✓	✓
Max	*****	****	****
B-D(C)	✓	✓	✓

Parse(σ,Ft)>>RED=$\sigma_{\mu\mu}$ and Max

lamb>>RED=$\sigma_{\mu\mu}$ and Max

Pattern IX: duhpek-->duhduhpek

	$duh (duhpek)	(duduh)(pek)	(duduh)pek	du(duhpek)
Cont	✓	✓	✓	✓
Anch	✓	✓	✓	✓
Fill	✓	✓	✓	✓
RED=Prefix	✓	✓	✓	✓
Align (Pwd, R, Ft, R)	✓	✓	*!	✓
lamb	*	✓*	✓	*
B-D(V)	✓	✓	✓	✓
*V=RB	✓	✓	✓	✓
FtHd=H	✓	*!	✓	✓
Parse(σ,Ft)	*	✓	*	*
RED=$\sigma_{\mu\mu}$	✓	*	*	*!
Max	***	****	****	****
B-D(C)	✓	✓	✓	✓

FtHd=H, RED=$\sigma_{\mu\mu}$, or Max>>Parse(σ,Ft)

Parse(σ,Ft)>>RED=$\sigma_{\mu\mu}$ and Max (Pattern IX: *luhmwuhmw*)

∴FtHd=H>>Parse(σ,Ft)

Align R, RED=$\sigma_{\mu\mu}$, or Max>>lamb

lamb>>RED=$\sigma_{\mu\mu}$ and Max (Pattern VI)

∴Align R>>lamb

Pattern X: mpek-->mpimpek

	$m(pmpek)	m(pimpek)	m(pempek)	mpe(kmpek)
Cont	✓	✓	✓	✓
Anch	✓	✓	✓	✓
Fill	✓	✓	✓	✓
RED=Prefix	✓	✓	✓	✓
Align (Pwd, R, Ft, R)	✓	✓	✓	✓
lamb	*	*	*	*
B-D(V)	✓	*!	✓	✓
FtHd=H	✓	✓	✓	✓
Parse(σ,Ft)	*	*	*	**!
RED=$\sigma_{\mu\mu}$	✓	*	*!	*
Max	**	**	*	✓
B-D(C)	✓	✓	✓	✓

RED=$\sigma_{\mu\mu}$>>Max

Parse(σ,Ft) or RED=$\sigma_{\mu\mu}$>>Max

Pattern XI: rere-->rerrere

	$rer(rere)	ree(rere)	re(rere)	(rere-er)
Cont	✓	✓	✓	*!
Anch	✓	✓	✓	✓
Fill	✓	✓	✓	✓
RED=Prefix	✓	✓	✓	*
Align R	✓	✓	✓	✓
lamb	*	*	*	✓
B-D(V)	✓	*!	✓	✓
*V=RB	✓	✓	✓	*
FtHd=H	*	*	*	✓
Parse(σ,Ft)	*	*	*	✓
RED=$\sigma_{\mu\mu}$	✓	✓	*!	*
Max	*	**	**	**
B-D(C)	✓	✓	✓	✓

Cont, RED=Prefix, *V=RB, RED=$\sigma_{\mu\mu}$, or Max>>FtHd=H and Parse(σ,Ft)

amiamas, ewiewetik dialect:

{Cont, Anch, {{Fill, RED=Pref, Align (Ft,R, Pwd, R)}>>lamb>>
{*V=RB, B-D(V)}>>FtHd=H>>Parse(σ,Ft)>>RED=$\sigma_{\mu\mu}$}}>>{Max, B-D(C)}

amahmas, ewehwetik dialect[17]:

{Cont, Anch, {{Fill, RED=Pref, Align (Ft,R, Pwd, R)}>>lamb>>
B-D(V)>>FtHd=H>>Parse(σ,Ft)>>RED=$\sigma_{\mu\mu}$}}>>{Max, B-D(C)}

[17] Here I assume that this dialect is distinguished by the absence of *V=RB. Nothing crucial hinges on this.